Approche arithmétique RNS de la cryptographie asymétrique

Julien Eynard

THÈSE DE DOCTORAT DE
l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Julien Eynard

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Approche arithmétique RNS de la cryptographie asymétrique

Soutenue le 28 Mai 2015

devant le jury composé de :

Rapporteur	S. Duquesne	Professeur, Université Rennes I
Rapporteur	L. Goubin	Professeur, Université de Versailles S^{t} Quentin-en-Yvelines
Examinateur	P. Elbaz-Vincent	Professeur, Université Joseph Fourier
Examinateur	C. Fontaine	Chargée de recherche CNRS, Télécom Bretagne
Examinateur	N. Guillermin	Ingénieur de l'Armement, Ministère de la Défense
Examinateur	A. Joux	Chaire de cryptologie, Université Pierre et Marie Curie
Directeur de thèse	J.-C. Bajard	Professeur, Université Pierre et Marie Curie
Co-directeur de thèse	L.-S. Didier	Professeur, Université de Toulon

À mes parents et mon frère.

Remerciements

Je tiens tout d'abord à remercier la Direction Générale de l'Armement, pour avoir financé ces travaux, et en particulier Madame Serfaty pour sa disponibilité.

Mes vifs remerciements vont à Sylvain Duquesne et Louis Goubin pour avoir accepté la tâche de rapporter ma thèse. Je sais mon style pas toujours des plus agréables à lire, mais j'essaie de me soigner. Ma gratitude va à Caroline Fontaine, Philippe Elbaz-Vincent, Nicolas Guillermin et Antoine Joux pour avoir bien voulu faire partie de ce jury de qualité. Plus particulièrement, je suis reconnaissant envers Philippe, sans qui cette thèse ne se serait faite.

Je remercie profondément mes directeurs de thèse, Jean-Claude Bajard et Laurent-Stéphane Didier, d'avoir placé en moi leur confiance pour accomplir ce travail. Si Laurent-Stéphane a rapidement été amené à partir sous des cieux plus ensoleillés, chaque rencontre a été source de conseils avisés et précieux.

Ma thèse restera indissociable de ma rencontre avec Jean-Claude. De par ses qualités humaines inestimables, il a su m'apporter beaucoup de son savoir, et pas uniquement sur le plan scientifique. Pour tout ce qu'il m'a donné, je lui voue une reconnaissance sans borne.

J'ai eu la chance d'avoir mon espace de travail en bureau 301. Enfin de travail, pas toujours... Et heureusement, car outre pouvoir décompresser, cela m'aura permis d'engranger des souvenirs bien agréables ! Ce bureau restera pour moi celui de Benoit et ses Converse, Christophe, ses bonnes bières et son fork de curve fever avec des bruitages totalement immersifs, Olga et son sérieux, qui a permis de recanaliser des esprits parfois trop dissipés, et Issam et sa compagnie toujours agréable entre deux escapades au pays de l'oncle Sam. Merci à vous d'avoir participé à faire de ce bureau un lieu où il a fait bon vivre.

Mais il y a aussi eu une vie en dehors du bureau 301. Outre par l'équipe PEQuAN dans son entièreté, mes pensées sont en particulier marquées par Nastya et une soirée mémorable s'il en est, Thibault et ses spams toujours d'une grande finesse, et Mourad et les parties de tennis de table à Autrans. Merci également à Thibault pour m'avoir fait confiance dès mon arrivée en me proposant d'encadrer ses TP à Polytech'.

Mieux, il y a aussi eu une vie en dehors du laboratoire ! Si j'ai pu conserver ma santé mentale par ces week-ends salutaires en terre natale, c'est pour beaucoup grâce à mes glunois préférés : Benjamin l'expert Reddit attitré de Glun, Cédric le fils unique, feu Bouli, Chattine, Madame et Monsieur, mais aussi PierrO, compagnon de galère grenoblois.

Enfin, mon éternelle reconnaissance et mon amour vont à ma famille, sans qui la vie n'aurait guère de sens. Merci à Gisèle et Philippe qui, par leur soutien et leur présence indéfectibles, m'ont permis de ne pas trébucher et de repartir du bon pied, et grâce à qui la vie est plus simple et plus belle, et à Rémi, mon frère adoré, avec qui j'espère partager encore de très nombreux moments dans un futur proche (en grande voie notamment j'espère ;) et plus lointain. La vie nous attend.

« If you have built castles in the air, your work need not be lost; that is where they should be. Now put the foundations under them. »
Henry David Thoreau

« Don't these talking monkeys know that Eden has enough to go around?
Plenty in this holy garden, silly monkeys,
Where there's one you're bound to divide it.
Right in two. »
Tool

Table des matières

Liste des Algorithmes

Liste des Figures

Liste des Tableaux

Notations

$\mathbb{R}$ ($\mathbb{R}^*$)	ensemble des nombres réels (privé de 0)
$\mathbb{Z}$ ($\mathbb{Z}^*$)	ensemble des entiers relatifs (privé de 0)
$\mathbb{N}$ ($\mathbb{N}^*$)	ensemble des entiers naturels (privé de 0)
$\mathbb{F}_p$	corps fini à p éléments
$\alpha, \beta, a, b, \ldots$	nombres réels, entiers relatifs ou naturels représentés par des caractères minuscules des alphabets latin et grec
$\mathbf{a}, \mathbf{v}$	vecteurs représentés par des caractères minuscules gras de l'alphabet latin
$\mathbf{Q}, \mathbf{R}$	matrices représentées par des caractères majuscules gras de l'alphabet latin
$[\![x, y[\![$	ensemble des entiers de l'intervalle $[x, y[$, *i. e.* $[x, y[\cap \mathbb{Z}$
$\mathbb{Z}/m\mathbb{Z}$	pour $m \in \mathbb{Z}$, anneau quotient des classes de congruence modulo m, classes aussi dénommées résidus
$(\mathbb{Z}/m\mathbb{Z})^\times$	groupe des unités de $\mathbb{Z}/m\mathbb{Z}$
$\lvert x \rvert_y \in \mathbb{Z}/y\mathbb{Z}$	pour $(x, y) \in \mathbb{Z} \times \mathbb{Z}^*$, résidu de x modulo y
$\lvert x^{-1} \rvert_y$	pour $(x, y) \in \mathbb{Z} \times \mathbb{Z}^*$ avec $\lvert x \rvert_y \in (\mathbb{Z}/y\mathbb{Z})^\times$, résidu de l'inverse de x modulo y
$x \wedge y$	pour $(x, y) \in \mathbb{Z}^2$, $\operatorname{pgcd}(x, y)$ qui par convention est positif
$\texttt{Mul}_\beta(u, v)$	multiplication d'un entier composé de u digits par un second de v digits en base β
$\texttt{Add}_\beta(u, v)$	addition d'un entier composé de u digits par un second de v digits en base β

$\texttt{EMul}_\beta$	multiplication « élémentaire » de deux entiers d'un digit en base β, soit une $\texttt{Mul}_\beta(1,1)$
$\texttt{EAdd}_\beta$	addition « élémentaire » de deux entiers d'un digit en base β, soit une $\texttt{Add}_\beta(1,1)$
$\texttt{MMEl}_\beta$	« multiplication modulaire élémentaire » de deux entiers d'un digit en base β modulo un entier d'un digit
$\texttt{AMEl}_\beta$	« addition modulaire élémentaire » de deux entiers d'un digit en base β modulo un entier d'un digit
$\texttt{ETEl}_\beta$	étape de calcul élémentaire suivant un schéma $a \leftarrow (a + b \times c) \bmod m$ ou $a \leftarrow (a + b) \times c \bmod m$, où les données a, b, c et m s'écrivent sur un digit en base β

Introduction

À l'ère du tout numérique, la cryptographie participe de la stabilité d'un paradigme complexe dans lequel nos sociétés modernes s'avancent chaque jour un peu plus. Si cet art de l'« écriture cachée » remonte à l'antiquité, son développement s'est accéléré avec l'évolution rapide des techniques de communication au cours du 20$^{\text{ième}}$ siècle et les nouvelles problématiques soulevées par un monde complètement connecté. Dans la période moderne, l'essor de la cryptographie a d'abord et avant tout été le fait des milieux militaires, avant de connaître une démocratisation certaine avec l'explosion d'Internet.

Une formalisation des principes de la cryptographie intervient en fait dès le 19$^{\text{ième}}$ siècle. Kerckhoffs (1883) établit dans « La cryptographie militaire » 6 principes fondamentaux visant à amoindrir la faillibilité d'un système de chiffrement. Parmi ceux-ci, les second et troisième incarnent notablement l'esprit de la cryptographie moderne.

> « *2^0 Il faut [que le système de chiffrement] n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;*
>
> *3^0 La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;* »

La sécurité du système ne doit dépendre que d'une « clef » facile à communiquer et à retenir, et ne doit en aucun cas être liée à la méthode de chiffrement. À partir de là, la cryptographie moderne se scinde en deux branches principales : la cryptographie symétrique et la cryptographie asymétrique.

La cryptographie symétrique est fondée sur la connaissance mutuelle par les parties concernées d'une même clef secrète. Les algorithmes de chiffrement symétrique les plus connus sont notamment ceux standardisés par le National Institute of Standards and Technology, anciens ou encore en vigueur, comme le Data Encryption Standard DES (FIPS 1999) remplacé en 2001 par l'Advanced Encryption Standard AES (FIPS 2001). Les principes généraux de fonctionnement des cryptosystèmes symétriques reposent sur des permutations, substitutions, combinaisons linéaires de blocs de données. L'information contenue dans la clef secrète est injectée au cours du schéma de chiffrement.

Contrairement à la cryptographie symétrique, la cryptographie asymétrique ne suppose aucun échange secret préalable. Pour mettre en œuvre ce principe, il s'agit d'utiliser des fonctions à trappe à sens unique. Par définition, pour une telle fonction le calcul des images est facile, mais le calcul des antécédents est particulièrement ardu lorsque la trappe est inconnue. Diffie et Hellman (1976) proposent une solution pratique à la problématique d'échange de clef, basée sur la fonction à trappe d'exponentiation modulaire dans un groupe multiplicatif. Pour qu'Alice et Bob puissent s'échanger une clef secrète, ils s'accordent dans un premier temps sur des paramètres $\mathbb{G}$, un groupe multiplicatif, et un élément $g \in \mathbb{G}$ de ce groupe. Le principe du protocole est alors résumé dans la

Table 1, où la clef secrète partagée par Alice et Bob est g^{ab}. La sécurité de cette technique repose sur la difficulté du problème du logarithme discret : à g^a, g^b et g donnés, il est difficile de retrouver g^{ab} en temps raisonnable.

Alice	canal	Bob
choisit $a \in \mathbb{N}$		choisit $b \in \mathbb{N}$
calcule g^a		calcule g^b
envoie g^a	$\rightarrow$	récupère g^a
		calcule $(g^a)^b = g^{ab}$
récupère g^b	$\leftarrow$	envoie g^b
calcule $(g^b)^a = g^{ab}$		

TABLE 1 – *Procotole d'échange de clefs de Diffie-Hellman.*

Rivest, Shamir, et Adleman proposent en 1978 un type de cryptosystème révolutionnaire, dénommé RSA, et donnent véritablement naissance à la cryptographie asymétrique. Dans ce nouveau paradigme, les parties impliquées ne partagent pas la même clef secrète. Chacune dispose d'une paire de clefs, l'une secrète et l'autre publique. Si Alice souhaite par exemple envoyer un message privé à Bob, il lui suffit de le chiffrer via la clef publique de Bob, celui-ci pouvant retrouver le message initial en utilisant sa propre clef secrète. L'intérêt est évident. Il n'y a plus besoin de la connaissance mutuelle préalable d'une information secrète pour pouvoir échanger de manière chiffrée. En pratique, les cryptosystèmes asymétriques ont l'inconvénient d'être beaucoup plus lents que les systèmes de chiffrement symétriques. Ainsi, la cryptographie asymétrique sert généralement à l'échange sécurisé d'une clef secrète symétrique, ou à établir des procédures d'authentification.

L'Algorithme 1 détaille le principe historique du RSA. Une paire de clefs asymétrique est la donnée d'un doublet public d'entiers (n, e), où n est le produit de deux grands nombres premiers p et q, et e un entier premier avec $(p-1)(q-1)$, et d'un entier secret d vérifiant $d \times e \equiv 1 \bmod (p-1)(q-1)$. Cette propriété implique que, pour tout entier m, $m^{de} \bmod n \equiv m \bmod n$. Pour que Bob récupère le message en clair m depuis le chiffré c à l'étape 6, il lui suffit alors de calculer $c^d \bmod n \equiv m^{de} \bmod n \equiv m \bmod n$. Or, si m est correctement choisi dans $[0, n-1]$, alors $m \bmod n = m$.

La sécurité de ces systèmes asymétriques repose sur des problèmes calculatoires difficiles à résoudre en temps raisonnable. Par exemple, la sécurité de RSA est conjecturée reposer sur la difficulté à factoriser le produit de deux grands nombres premiers. Le problème du logarithme discret (Odlyzko 1985) déjà cité précédemment nourrit une autre classe importante de problèmes utilisés pour construire aussi bien des schémas de chiffrement (Elgamal 1985), que des schémas de signature comme DSA (Digital Signature Algorithm) (FIPS 2013). Les structures mathématiques sous-jacentes sont également variées. Outre l'arithmétique modulaire sur les entiers, les courbes elliptiques sont par exemple aussi largement utilisées (Koblitz 1987, Miller 1986), et sujets à standardisation (ECDSA Elliptic Curve Digital Signature Algorithm (FIPS 2013), ECDH Elliptic Curve Diffie-Hellman (SP 2007)).

Une nouvelle étape dans le monde de la cryptographie asymétrique a été marquée par un résultat de Shor (1994), lequel montrant que l'avènement de l'informatique quantique verra l'obsolescence des fonctions à trappe à sens unique construites sur les problèmes de factorisation et de recherche de lo-

Algorithme 1 : Procédures de chiffrement/déchiffrement RSA

Données :

- Données publiques : la clef publique de Bob constituée de deux entiers $s_{Bob} = (n, e)$.
- Donnée privée : d la clef secrète de Bob.

Entrées : m le message qu'Alice souhaite envoyer à Bob, m est un entier dans $[\![0, n-1]\!]$.

1 **début**
2 Alice récupère s_{Bob} ;
3 Alice calcule $c \leftarrow m^e \bmod n$;
4 Alice envoie c à Bob ;
5 Bob récupère c ;
6 Bob calcule $m \leftarrow c^d \bmod n$.

garithme discret. Ce constat met en lumière d'autres approches pouvant répondre à cette nouvelle problématique, comme l'est la cryptographie basée sur les réseaux euclidiens (Ajtai 1996, Goldreich et al. 1997, Hoffstein et al. 1998). Outre le fait de constituer un candidat de choix pour une cryptographie post-quantique, l'attrait de ces systèmes est due à des problèmes calculatoires possédant des propriétés intéressantes de réduction aux pires cas, ainsi qu'à des propriétés d'homomorphie (Gentry 2009) rendant possible l'exécution d'opérations arithmétiques directement sur des données chiffrées. Cette dernière particularité se révèle d'autant plus précieuse à l'heure où le « cloud computing » est en plein essor.

L'évolution grandissante et ininterrompue des applications, outils et interactions numériques fait évoluer constamment les contraintes de sécurisation. La cryptographie asymétrique doit ainsi se développer et tendre vers des objectifs d'optimisation des primitives cryptographiques, afin d'accroître les performances en termes de temps d'exécution, de surface matérielle, ou encore de consommation d'énergie. La carte à puce avec son format largement standardisé est un exemple typique de matériel autonome utilisé pour diverses opérations sensibles (paiement, identification, authentification, etc) et devant répondre à un cahier des charges spécifique (rapidité des protocoles, sécurisation des données et échanges, paiement sans contact, etc). La nomadisation des supports intégrant des fonctions cryptographiques donne une dimension particulière à certaines contraintes de sécurité liées aux interactions physiques avec le monde extérieur. Un matériel embarquant un cryptosystème se trouvant entre les mains d'un tiers peu fiable peut être le sujet d'attaques diverses et variées, sans que la sécurité théorique dudit système ne soit en cause. Ces attaques par canaux auxiliaires reposent sur de possibles failles dans l'implantation matérielle et/ou logicielle. Les « canaux » susceptibles de laisser fuir des informations observables et exploitables par l'adversaire sont multiples : analyse du temps d'exécution (Kocher 1996, Brumley et Boneh 2003), des relevés de consommation d'énergie (Kocher et al. 1999; 2011, Coron 1999, Goubin 2002), d'émissions électromagnétiques (Gandolfi et al. 2001, Agrawal et al. 2003). En outre, des attaques plus invasives peuvent être élaborées lorsqu'il y a une capacité à provoquer l'apparition de fautes dans les données traitées

(Barenghi et al. 2012). Selon les moyens techniques dont dispose l'adversaire, des impulsions laser, de rayonnements électromagnétiques, des fluctuations de l'alimentation en énergie, ou autres, peuvent être très efficaces pour casser certaines implantations spécifiques de cryptosystèmes comme RSA (Boneh et al. 1997) ou ECDSA (Barenghi et al. 2011). La cryptographie ne peut donc pas se cantonner aux aspects mathématiques et algorithmiques, mais doit aussi embrasser des considérations beaucoup plus vastes en intégrant au mieux le modèle du monde réel pour lequel elle est faite.

Objectif général

Cette thèse s'inscrit dans un axe de recherche général concernant l'amélioration pratique des primitives cryptographiques asymétriques. Plus précisément, nous nous intéressons aux moyens permettant d'accélérer une primitive, ou encore de la prémunir contre certains types d'attaques tels que décrits précédemment. Pour ce faire, la méthodologie choisie repose sur une approche particulière mais source de nombreux résultats récents et significatifs. Il est question d'aborder les problématiques d'optimisation vues plus avant par le biais des systèmes de représentation des nombres et de l'arithmétique engendrée. Plus précisément, les systèmes de représentation des nombres par les restes, appelés Residue Number Systems (RNS) en anglais, ont naturellement attiré l'attention des personnes animées par les motivations sus-citées. Cela tient au fait que les RNS offrent un paradigme singulier, dans lequel l'arithmétique sur de grands entiers, dont se nourrit notablement la cryptographie asymétrique, prend un autre visage. Ces entiers deviennent représentables par un ensemble fini de petites valeurs indépendantes, sur lesquelles les opérations arithmétiques de base peuvent être réalisées directement, d'une manière parallèle et sans nécessité aucune d'un quelconque mécanisme de propagation de retenue. Ainsi, les spécificités même des RNS recoupent les considérations ayant trait aux besoins d'accélération des calculs, ou encore de diminution des besoins en consommation. Les résultats obtenus jusqu'à présent, dans ce domaine à la croisée de la cryptographie et de l'arithmétique des ordinateurs, sont nombreux et variés (Kawamura et al. 2000, Bajard et al. 2001, Bajard et Imbert 2004, Bajard et al. 2004, Guillermin 2010, Cheung et al. 2011, Bajard et al. 2013a) et les recherches sur le sujet demeurent aujourd'hui encore très actives (Bigou et Tisserand 2014, Bajard et Merkiche 2014).

L'objectif poursuivi dans cette thèse est d'identifier des points de convergence possible entre des besoins relatifs aux problématiques intéressant la cryptographie asymétrique et certaines propriétés apportées par les RNS, puis d'étudier dans quelle mesure cela peut mener à des solutions efficaces.

Plan du mémoire

Le premier chapitre présente les systèmes de représentation par les restes ainsi que les outils essentiels pour les propos du mémoire. Après l'introduction des principales notations utilisées par la suite, une définition formelle des RNS est donnée, comme conséquence du théorème des restes chinois. Les avantages immédiats apportés par ces systèmes de représentation des nombres en terme de complexité sur les opérations arithmétiques de base sont expliqués, puis le mixed-radix system (MRS), système positionnel non standard naturellement

associé aux RNS, est décrit. L'utilisation des RNS an arithmétique modulaire s'appuie sur des opérations de conversion de base. Les techniques de conversion les plus utilisées en pratique sont détaillées, ce qui permet alors d'introduire le principe de la réduction modulaire en RNS, que ce soit pour une arithmétique dans un corps fini premier ou non.

Le deuxième chapitre est dédié à une problématique essentielle en cryptographie asymétrique, la détection de faute. Les RNS redondants fournissent un moyen de détection d'erreur qui a été abondamment étudié au cours des dernières décennies. Cependant, ces RNS redondants semblent difficilement compatibles avec une arithmétique modulaire RNS, à cause de la structure particulière de la réduction modulaire. Après les avoir introduits, l'objectif de ce chapitre est d'étudier dans quelle mesure RNS redondants et arithmétique dans les corps finis peuvent s'accommoder. Afin de mener une étude détaillée, le cas des fautes simples (*i.e.* sur une seule unité RNS) et des corps finis premiers sera privilégié dans un premier temps. Dans le but de raccrocher cette étude théorique à un contexte d'implantation matérielle, le modèle de faute est ensuite affiné pour s'adapter à des contraintes plus spécifiques, et des considérations générales pour une intégration des RNS redondants à des architectures classiques dédiées aux RNS sont données.

Le troisième chapitre est consacré à la généralisation des principes développés dans le chapitre deux. La première étape consiste à élargir le modèle de faute, en l'ouvrant désormais aux fautes multiples. Une seconde étape se concentre sur l'adaptation des techniques précédentes au cas des extensions de corps finis.

Enfin, dans un quatrième chapitre, les RNS sont exploités dans le domaine de la cryptographie basée sur les réseaux euclidiens. La faisabilité d'une adaptation RNS de la procédure classique appelée round-off est étudiée en détail. Dans un premier temps, deux approches différentes sont développées et comparées, afin de fournir un algorithme en représentation hybride RNS-MRS. Dans un second temps, une technique d'accélération construite sur les caractéristiques géométriques des bases de réseaux utilisées est présentée. Elle permet de construire un algorithme réalisant le round-off entièrement en RNS.

Contributions ayant donné lieu à publication

RNS et détection de fautes en arithmétique modulaire (Bajard, Eynard, et Gandino 2013b) Les résultats contenus dans le deuxième chapitre et concernant la protection de la multiplication modulaire contre les fautes sur un résidu ont donné lieu à la publication d'un article intitulé

« Fault Detection in RNS Montgomery Modular Multiplication »

dans les actes de la conférence 21^{st} *IEEE Symposium on Arithmetic Computer* tenue du 7 au 10 avril 2013 à Austin, Texas, en collaboration avec Jean-Claude Bajard, du Laboratoire d'Informatique de Paris 6, et Filippo Gandino, du département « Department of Control and Computer Engineering » à Politecnico di Torino (Italie).

RNS et cryptographie basée sur les réseaux (Bajard, Eynard, Merkiche, et Plantard 2014; 2015) Les travaux ayant trait à l'application du RNS dans le

domaine de la cryptographie basée sur les réseaux et présentés dans le quatrième chapitre ont été diffusés via deux articles, en collaboration avec Jean-Claude Bajard, Nabil Merkiche, ingénieur de l'Armement à la Direction générale de l'Armement, département Maîtrise de l'Information (Rennes), et Thomas Plantard, docteur au « Center for Computer and Information Security Research » de l'Université de Wollongong (Australie).

Le premier (Bajard et al. 2014) concerne la création d'un algorithme résolvant le problème du plus proche vecteur en utilisant l'approche du round-off de Babai et basé sur une représentation hybride RNS-MRS. L'article intitulé

« Babai round-off CVP method in RNS : Application to lattice based cryptographic protocols »

a été publié dans les actes de la conférence *14th International Symposium on Integrated Circuits*, tenue du 10 au 12 décembre 2014 à Singapour.

Le second (Bajard et al. 2015) présente une technique d'accélération de l'approche précédente permettant d'aboutir à un algorithme entièrement RNS. L'article intitulé

« RNS Arithmetic Approach in Lattice-based Cryptography - Accelerating the "Rounding-off" Core Procedure »

a été accepté pour publication à la conférence *22nd IEEE Symposium on Arithmetic Computer* qui aura lieu à Lyon du 22 au 24 juin 2015.

Les systèmes de représentation par les restes

1

Sommaire

Ce chapitre définit le cadre théorique mathématique et algorithmique des systèmes de représentation par les restes. Dans une première section, les RNS sont formellement définis. Afin de faire le lit des propos de ce mémoire quant aux contributions qu'il apporte, nous exposons les aspects algorithmiques liés aux opérations arithmétiques RNS, et notamment à la réduction modulaire, opération essentielle pour créer une arithmétique sur des corps finis. Enfin, nous dressons une revue rapide de l'état-de-l'art concernant l'utilisation des RNS dans le contexte de corps finis non premiers.

1.1 Les Systèmes de Représentation par les Restes

Cette section brosse un portrait des systèmes de représentation par les restes. Nous détaillons les principaux aspects algorithmiques liés aux opérations arithmétiques tenant une place centrale dans le contexte des contributions originales apportées dans ce mémoire.

1.1.1 Un système non positionnel de représentation des nombres

L'existence des systèmes de représentation des nombres par les restes est une conséquence du théorème des restes chinois (TRC). Des prémisses de ce résultat remontant aux premiers siècles de notre ère ont trait à des considérations portant sur une résolution heuristique de systèmes d'équations congruentielles. De nombreuses études sur l'Histoire des Mathématiques montrent comment ces considérations arithmétiques ont été particulièrement persistantes à travers les siècles et les civilisations (Ing 2003, Kangsheng 1988, Lam et Ang 2004). L'un des témoignages emblématiques se retrouve dans l'énoncé d'un problème issu d'un traité mathématique chinois du troisième siècle de notre ère et intitulé « Sun Zi Suanjing ». Une traduction anglaise est disponible dans (Ing 2003).

> « *Now there are an unknown number of things. If we count by threes, there is a remainder 2 ; if we count by fives, there is a remainder 3 ; if we count by sevens, there is a remainder 2. Find the number of things.* »

En l'état actuel, ce théorème se décline en plusieurs versions à caractère plus ou moins général. Il décrit des relations riches entre certains anneaux quotients, qui vont notamment permettre de construire des systèmes de représentation des nombres particuliers, les Residue Number Systems (RNS). Ceux-ci, apparus dans les années 50 (Garner 1959), n'ont eu de cesse de croître en popularité, notamment au sein de la communauté centrée sur le traitement du signal (Jenkins et Leon 1977, Soderstrand et al. 1986, Conway et Nelson 2004). Cet intérêt est fondé sur la possibilité donnée par le RNS de passer du paradigme arithmétique traditionnel de l'anneau des nombres entiers $\mathbb{Z}$ à un paradigme différent, au sein duquel les entiers deviennent représentables par de petites quantités indépendantes les unes des autres. Il devient même possible de transposer simplement les opérations arithmétiques standards d'addition et de multiplication directement sur ces petites unités. De ce fait, des opérations comme les « sommes de produits », qui sont un schéma calculatoire récurrent dans le traitement du signal notamment (transformée de Fourier rapide (FFT), etc), peuvent être réalisées très efficacement (Tseng et al. 1979, Huang et Taylor 1980, Taylor 1990). De plus, le RNS est un candidat compétitif concernant les problématiques d'implantation matérielle à basse consommation d'énergie (Freking et Parhi 1997, Stouraitis et Paliouras 2001), point essentiel pour réaliser des implantations sur matériel embarqué par exemple.

Nous allons introduire une formulation de ce théorème, qui apporte la première pierre dans la construction des RNS. Comme il est question de relations entre des anneaux quotients, il est important de fixer dès à présent les notations suivantes.

Définition 1.1 *Soit $M \in \mathbb{Z}$ un entier positif non nul. Un élément de l'anneau quotient $\mathbb{Z}/M\mathbb{Z}$ est noté $|x|_M$, où x est un entier dont la classe dans $\mathbb{Z}/M\mathbb{Z}$ est par définition $|x|_M$. Autrement dit, à un élément de $\mathbb{Z}/M\mathbb{Z}$ noté $|x|_M$ est naturellement associé le représentant x. De*

plus, si $\mathcal{C}_M \subset \mathbb{Z}$ est un système complet de représentants des classes de $\mathbb{Z}/M\mathbb{Z}$, alors $\sigma_{\mathcal{C}_M}$ désigne la bijection canonique suivante :

$$\begin{array}{rccc} \sigma_{\mathcal{C}_M} : & \mathbb{Z}/M\mathbb{Z} & \to & \mathcal{C}_M \\ & |x|_M & \mapsto & y \in \mathcal{C}_M,\ y \equiv x \bmod M \end{array} \tag{1.1}$$

Le caractère très général de l'énoncé qui suit tient au fait de notre volonté de faire la distinction entre une classe et un représentant. Cette démarche a pour objectif de justifier le plus rapidement possible le fait de ne plus avoir à distinguer une classe d'un représentant et vice-versa, sans que cela ne soulève de problème d'interprétation lorsque nous définirons les RNS.

Théorème 1.1 ***(des restes chinois, TRC)*** *Soit n entiers $(m_1, \ldots, m_n) \in \mathbb{Z}^n$, $M = \text{ppcm}(m_1, \ldots, m_n)$, et $\mathcal{C}_M \subset \mathbb{Z}$ un système complet de représentants des classes de $\mathbb{Z}/M\mathbb{Z}$. Alors l'application $\varphi_{\mathcal{C}_M}$ suivante est un morphisme d'anneaux injectif :*

$$\begin{array}{rccccc} \varphi_{\mathcal{C}_M} : & \mathbb{Z}/M\mathbb{Z} & \overset{\sigma_{\mathcal{C}_M}}{\to} & \mathcal{C}_M & \overset{\chi_{\mathcal{C}_M}}{\to} & \mathbb{Z}/m_1\mathbb{Z} \times \ldots \times \mathbb{Z}/m_n\mathbb{Z} \\ & |x|_M & \mapsto & \sigma_{\mathcal{C}_M}(x) & \mapsto & \left(|\sigma_{\mathcal{C}_M}(x)|_{m_1}, \ldots, |\sigma_{\mathcal{C}_M}(x)|_{m_n}\right) \end{array} \tag{1.2}$$

De plus, $\varphi_{\mathcal{C}_M}$ ne dépend pas du choix de $\mathcal{C}_M$. Enfin $\varphi_{\mathcal{C}_M}$ est bijective si, seulement si, $\prod_{i=1}^{n} m_i = M$.

Démonstration. Si $\mathcal{C}'_M$ est un autre système complet de représentants de $\mathbb{Z}/M\mathbb{Z}$, alors $\chi_{\mathcal{C}_M} \circ \sigma_{\mathcal{C}_M}$ et $\chi_{\mathcal{C}'_M} \circ \sigma_{\mathcal{C}'_M}$ définissent bien la même application φ. En effet, si $(x, a, b) \in \mathbb{Z}/M\mathbb{Z} \times \mathcal{C}_M \times \mathcal{C}'_M$ avec $a = \sigma_{\mathcal{C}_M}(x)$ et $b = \sigma_{\mathcal{C}'_M}(x)$, alors par définition nous avons $a - b \in M\mathbb{Z} \subseteq \bigcap_{i=1}^{n} m_i\mathbb{Z}$. Donc $\chi_{\mathcal{C}_M}(a) = \chi_{\mathcal{C}'_M}(b)$, et par suite $\chi_{\mathcal{C}_M} \circ \sigma_{\mathcal{C}_M}(x) = \chi_{\mathcal{C}'_M} \circ \sigma_{\mathcal{C}'_M}(x)$. $\varphi_{\mathcal{C}_M}$ est ainsi bien définie indépendamment du choix de $\mathcal{C}_M$ ou $\mathcal{C}'_M$. Nous pouvons ainsi dès maintenant la noter plus simplement φ. Le choix des représentants restera donc, sauf indication contraire, implicite. Par soucis de simplification toute notation du type $|x|_M$ pourra ainsi également faire référence à un quelconque de ses représentants dans $\mathbb{Z}$.

La preuve du fait que φ est un morphisme est, somme toute, aisée. L'élément clef est que $M\mathbb{Z} \subseteq \bigcap_{i=1}^{n} m_i\mathbb{Z}$. Soit par exemple $(a, b) \in \mathbb{Z}^2$. Alors nous avons :

$$\begin{aligned} \varphi(|a|_M + |b|_M) &= \varphi(|a+b|_M) \to \textit{loi additive de } \mathbb{Z}/M\mathbb{Z} \\ &= \left(||a+b|_M|_{m_1}, \ldots, ||a+b|_M|_{m_n}\right) \\ &= \left(|a+b|_{m_1}, \ldots, |a+b|_{m_n}\right) \to \textit{car } M\mathbb{Z} \subseteq \bigcap_{i=1}^{n} m_i\mathbb{Z} \\ &= \left(|a|_{m_1} + |b|_{m_1}, \ldots, |a|_{m_n} + |b|_{m_n}\right) \to \textit{lois additives des } \mathbb{Z}/m_i\mathbb{Z} \\ &= \left(|a|_{m_1}, \ldots, |a|_{m_n}\right) + \left(|b|_{m_1}, \ldots, |b|_{m_n}\right) \to \textit{loi additive de } \prod_{i=1}^{n} \mathbb{Z}/m_i\mathbb{Z} \\ &= \varphi(|a|_M) + \varphi(|b|_M). \end{aligned}$$

En procédant de manière similaire, nous obtenons $\varphi(|a \times b|_M) = \varphi(|a|_M) \times \varphi(|b|_M)$ ainsi que $\varphi(|1|_M) = \left(|1|_{m_1}, \ldots, |1|_{m_n}\right)$. Ceci conclut la preuve du fait que φ est un morphisme d'anneaux.

La preuve de l'injectivité de φ est directe. En effet, pour tout $a \in \mathbb{Z}$, alors $\varphi\left(|a|_M\right) = 0 \Leftrightarrow \left(\forall i \in [\![1,n]\!],\ |a|_{m_i} = 0\right) \Leftrightarrow a \in \bigcap_{i=1}^{n} m_i = M\mathbb{Z}$. Par conséquent, $\ker(\varphi) = \{0\}$.

Si φ est surjective, et donc bijective, alors en particulier $M = \prod_{i=1}^{n} m_i$. Supposons maintenant à l'inverse que $\prod_{i=1}^{n} m_i = M$. Alors le Lemme 1.1 qui suit permet de conclure à la surjectivité. □

Lemme 1.1 *Soit $n \in \mathbb{Z}$, $n \geqslant 2$, et $\mathcal{B} = \{m_1, \ldots, m_n\} \subset \mathbb{N}$ un ensemble de n entiers.*

1. $\mathrm{ppcm}(m_1, \ldots, m_n) = \prod_{i=1}^{n} m_i$ *si, et seulement si, les entiers de $\mathcal{B}$ sont premiers entre eux deux à deux.*

2. *Les entiers de $\mathcal{B}$ sont premiers entre eux deux à deux si, et seulement si, pour tout $(x_1, \ldots, x_n) \in \prod_{i=1}^{n} \mathbb{Z}/m_i\mathbb{Z}$, il existe $x \in \mathbb{Z}$ tel que $\left(|x|_{m_1}, \ldots, |x|_{m_n}\right) = (x_1, \ldots, x_n)$.*

Démonstration. 1. Le principe de la preuve, que nous ne détaillons pas, est une simple récurrence sur n en utilisant les résultats élémentaires d'arithmétique suivants : pour tout triplet d'entiers $(a,b,c) \in \mathbb{Z}^3$, $|ab| = \mathrm{pgcd}(a,b) \times \mathrm{ppcm}(a,b)$ et $\mathrm{ppcm}(a,b,c) = \mathrm{ppcm}(\mathrm{ppcm}(a,b), c)$.

2. Prouvons la suffisance. Montrons que pour tout $(i,j) \in [\![1,n]\!]^2$ tel que $i \neq j$, $\mathrm{pgcd}\left(m_i, m_j\right) = 1$. Par hypothèse, il existe $x \in \mathbb{Z}$ vérifiant en particulier $|x|_{m_i} = |1|_{m_i}$ et $|x|_{m_j} = |0|_{m_j}$. Autrement dit, il existe $\left(k_i, k_j\right) \in \mathbb{Z}^2$ tel que $x = 1 + k_i m_i$ et $x = k_j m_j$. Par conséquent, $k_i m_i - k_j m_j = 1$, ce qui, par le théorème de Bachet-Bézout, implique que $\mathrm{pgcd}\left(m_i, m_j\right) = 1$.

   Prouvons la nécessité. Les entiers $m_1, \ldots, m_n$ sont supposés premiers entre eux deux à deux. En particulier, par l'assertion 1 du lemme, si M désigne le $\mathrm{ppcm}(m_1, \ldots, m_n)$, alors $M = \prod_{i=1}^{n} m_i$. Ainsi, $\left|\frac{M}{m_i}\right|_{m_i} \in \mathbb{Z}/m_i\mathbb{Z}^{\times}$. Son inverse est alors noté ξ_i, et ce pour tout $i \in [\![1,n]\!]$. Soit $(x_1, \ldots, x_n) \in \prod_{i=1}^{n} \mathbb{Z}/m_i\mathbb{Z}$, et, pour tout $i \in [\![1,n]\!]$, $\hat{x}_i \in \mathbb{Z}$ un représentant de x_i. Alors tout représentant x de la classe de congruence modulo M définie par $\left|\left(\sum_{i=1}^{n} \hat{x}_i \times \xi_i \times \frac{M}{m_i}\right)\right|_M$ vérifie $|x|_{m_i} = x_i$ pour tout $i \in [\![1,n]\!]$, ce qui termine la preuve.

□

Remarque 1.1 *En considérant que les entiers $\{m_1, \ldots, m_n\}$ sont premiers entre eux deux à deux, que $M = \prod_{i=1}^{n} m_i$, et que $\mathcal{C}_M$ est de la forme $a + [\![0, M[\![$, pour a un entier quelconque, la preuve de l'assertion 2 du Lemme 1.1 montre comment construire un antécédent du n-uplet $(x_1, \ldots, x_n) \in \prod_{i=1}^{n} \mathbb{Z}/m_i\mathbb{Z}$ par l'application $\chi_{\mathcal{C}_M}$ (1.2). En notant $M_i = \frac{M}{m_i}$ et*

$\left|M_i^{-1}\right|_{m_i}$ *l'inverse de* $\left|\frac{M}{m_i}\right|_{m_i}$ *dans* $\mathbb{Z}/m_i\mathbb{Z}$*, alors nous avons :*

$$\chi^{-1}_{[\![k,k+M[\![}((x_1,\ldots,x_n)) = a + \left[\left(\sum_{i=1}^{n} \sigma_{[\![0,m_i[\![}\left(x_i \times \left|M_i^{-1}\right|_{m_i}\right) \times M_i\right) \bmod M\right]. \tag{1.3}$$

Comme φ est injective, le Théorème 1.1 fournit donc un moyen de créer un système de représentation univoque des éléments de tout ensemble de représentants des classes de congruence modulo M par des uplets de représentants de ces mêmes éléments dans des anneaux congruentiels construits à partir des facteurs de M. Comme souligné dans la preuve du théorème, le choix des représentants $\mathcal{C}_M$ ne joue pas sur la définition de φ. En pratique, il est courant de travailler avec un intervalle $[\![a, a+M[\![$. L'intérêt est de couvrir un ensemble de données sur lesquelles des opérations arithmétiques peuvent être effectuées, et qui présente donc une certaine stabilité pour l'addition et la multiplication.

L'entier M n'est généralement pas une donnée initiale dans la construction d'un RNS, mais seulement un paramètre : pour tout jeu de données $\mathcal{E} \subset \mathbb{Z}$, un système de représentation RNS des éléments de $\mathcal{E}$ se construit avant tout par le choix de l'ensemble des moduli m_i.

Remarque 1.2 *La discussion concernant le fait que le choix de l'ensemble $\mathcal{C}_M$ ne modifie pas la définition de φ vaut également pour l'application inverse φ^{-1} définie sur* $\mathrm{Im}(\varphi)$.

Le Théorème fondamental 1.1 désormais posé, il est possible de définir formellement les systèmes de représentation RNS.

Définition 1.2 *Soit $\mathcal{E} \subseteq \mathbb{Z}$ un ensemble fini d'entiers, et $M_\mathcal{E} = \max\{|x-y| \mid (x,y) \in \mathcal{E}^2\}$ son diamètre. Un système de représentation RNS des éléments de $\mathcal{E}$ est la donnée d'un ensemble de $n \in \mathbb{N}^*$ entiers $\mathcal{B} = \{m_1,\ldots,m_n\}$, dénommés moduli, vérifiant $M = \mathrm{ppcm}(m_1,\ldots,m_n) \geqslant M_\mathcal{E}$, et de l'application de conversion $\psi_{\mathcal{E},\mathcal{B}}$ consistant en la composition d'applications suivante :*

$$\begin{array}{rccccc} \psi_{\mathcal{E},\mathcal{B}} : & \mathcal{E} & \overset{\rho_M}{\to} & \mathbb{Z}/M\mathbb{Z} & \overset{\varphi_\mathcal{B}}{\to} & \mathbb{Z}/m_1\mathbb{Z} \times \ldots \times \mathbb{Z}/m_n\mathbb{Z} \\ & x & \mapsto & |x|_m & \mapsto & \left(|x|_{m_1},\ldots,|x|_{m_n}\right). \end{array} \tag{1.4}$$

La taille de l'espace d'états du système est défini par $M^\mathcal{B} = \prod_{i=1}^{n} m_i$, et l'intervalle $[\![0, M[\![$ est dénommé intervalle dynamique.

L'ensemble $\mathcal{B}$ est appelé base RNS. Deux bases RNS $\mathcal{B}$ et $\mathcal{B}'$ sont dites premières entre elles, ou copremières, si, et seulement si, les entiers $M^\mathcal{B}$ et $M^{\mathcal{B}'}$ sont premiers entre eux. Tout anneau quotient $\mathbb{Z}/m_i\mathbb{Z}$ pourra aussi être dénommé « canal » de $\mathcal{B}$.

Exemple 1.1 *Le problème énoncé dans « Sun Zi Suanjing » suggère l'utilisation du RNS formé par la base $\mathcal{B} = \{3,5,7\}$. L'intervalle dynamique du système est $[\![0,105[\![$. Un tel système permet de représenter de manière univoque tout ensemble d'entiers $\mathcal{E}$ de diamètre maximal* 105.

Par exemple, en choisissant basiquement $\mathcal{E} = [\![0,105[\![$, la représentation du nombre 23 *est donnée par :*

$$\psi_{\mathcal{E},\mathcal{B}}(23) = \varphi_\mathcal{B}(\rho_{105}(23)) = \varphi_\mathcal{B}(|23|_{105}) = (|23|_3, |23|_5, |23|_7) = (|2|_3, |3|_5, |2|_7).$$

L'unique solution du problème de « Sun Zi Suanjing » dans l'ensemble $\mathcal{E}$ est donc 23.

La construction de l'application $\varphi_{\mathcal{B}}$ de la Définition 1.2 se fait en pratique par la bijection naturelle entre $\mathbb{Z}/M\mathbb{Z}$ et $\mathcal{C}_M = [\![0, M[\![$. Il est cependant toujours possible de choisir $\mathcal{C}_M$ tel que $\mathcal{E} \subset \mathcal{C}_M$. L'application ρ est évidemment injective puisque $\text{Card}(\mathcal{E}) < M$. Ainsi, par le Théorème 1.1 des restes chinois et la condition $M \geqslant M_{\mathcal{E}}$, l'application $\psi_{\mathcal{E},\mathcal{B}} = \varphi_{\mathcal{B}} \circ \rho_M$ est injective, et fournit la représentation univoque de tout élément de $\mathcal{E}$ par ses résidus dans $\mathbb{Z}/m_1\mathbb{Z} \times \ldots \times \mathbb{Z}/m_n\mathbb{Z}$. La conversion inverse est donc bien définie sur $\text{Im}_{\psi_{\mathcal{E},\mathcal{B}}}(\mathcal{E})$, et $\mathcal{E} = \psi^{-1}_{\mathcal{E},\mathcal{B}}\left(\text{Im}_{\psi_{\mathcal{E},\mathcal{B}}}(\mathcal{E})\right)$. De plus, si les moduli sont choisis de manière à ce qu'ils soient premiers entre eux deux à deux, alors d'après le Théorème 1.1 l'application $\varphi_{\mathcal{B}}$ est bijective, et la Formule 1.3 donne une construction effective de $\varphi_{\mathcal{B}}$. Dans la mesure où la non surjectivité de $\varphi_{\mathcal{B}}$ ne nous intéressera pas pour définir un RNS, la convention suivante est établie pour la suite :

Convention 1.1 *Les moduli d'une base RNS sont, sauf mention contraire, premiers entre eux deux à deux.*

Remarque 1.3 *De la convention précédente il découle alors que tout sous-ensemble d'une base RNS est une base RNS. Il en va de même pour la conjonction de deux bases copremières.*

La condition de coprimalité des moduli n'est pas réellement limitante. L'intérêt des RNS repose en grande partie sur la flexibilité dans le choix des tailles et formes des moduli sélectionnés. Il est peu utile d'augmenter l'espace d'états du système en rajoutant à la base $\mathcal{B}$ des produits des facteurs de ces moduli. Cela ne changerait pas l'intervalle dynamique $[\![0, M[\![$. De plus, introduire une redondance en procédant de cette manière revient simplement à utiliser une répétition de certains résidus. Or il se trouve, comme il sera vu par la suite, qu'il est beaucoup plus utile, dans le cadre de la création de systèmes RNS redondants, de rajouter des moduli supplémentaires premiers à ceux de la base initiale. Dans de tels systèmes, les résidus redondants ont l'avantage de contenir une information partagée avec l'ensemble des résidus du système RNS principal, ce qui se révèle pratique lorsqu'il s'agit par exemple de construire des procédures de détection d'erreurs (cf. Chapitre 2).

Les moduli d'un RNS jouant des rôles identiques, un RNS est intrinsèquement un système de représentation non positionnel. Les résidus n'ont pas de relation d'interdépendance, et n'ont donc pas de poids particulier l'un envers l'autre. De ce fait, l'opération de comparaison sur l'unique donnée de résidus se révèle compliquée. Ceci explique que des opérations classiques comme la division sur les entiers et la réduction modulaire sont difficile à mettre en œuvre dans ces systèmes, et peuvent nécessiter de passer par une représentation positionnelle alternative comme le Mixed Radix System (MRS, cf. 1.2.4), ou bien d'utiliser des procédures spécifiques qui permettent de changer de base RNS.

1.1.2 Notations et termes spécifiques aux RNS

Afin de faciliter les discussions à venir, nous établissons une liste de notations et de termes fréquemment utilisés par la suite. Dans un premier temps, la convention suivante est établie :

Convention 1.2 *Pour tout doublet d'entiers $(a, b) \in \mathbb{Z}$, la notation $|a|_b$ désigne indifféremment le représentant de a dans $\mathbb{Z}/b\mathbb{Z}$ et l'entier $\sigma_{[\![0,b[\![}(|a|_b) = a \bmod b \in [\![0, b[\![$.*

Pour la majeure partie des propos du mémoire, nous utiliserons fréquem-

ment deux bases RNS copremières ayant respectivement n et ℓ moduli : $\mathcal{B} = \{m_1, \ldots, m_n\}$ et $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$. Les notations qui suivent concernent directement la base $\mathcal{B}$ mais valent également pour $\mathcal{B}'$ et toute autre base RNS qui pourra être introduite si besoin.

base RNS :
$\rightsquigarrow$ tout ensemble d'entiers premiers entre eux deux à deux

modulus (pluriel : moduli) :
$\rightsquigarrow$ tout élément d'une base RNS (*e.g.* m_1 est un modulus de $\mathcal{B}$)

bases copremières :
$\rightsquigarrow$ tout ensemble de bases RNS dont la conjonction est une base RNS

canal de $\mathcal{B}$:
$\rightsquigarrow$ tout quotient $\mathbb{Z}/m_i\mathbb{Z}$ où m_i est un modulus de $\mathcal{B}$

M, M'
$\rightsquigarrow M = \prod_{i=1}^{n} m_i,\ M' = \prod_{i=1}^{\ell} m'_i$

intervalle dynamique de $\mathcal{B}$:
$\rightsquigarrow [\![0, M[\![$

M_i**, pour tout** $i \in [\![1, n]\!]$ **:**
$\rightsquigarrow \dfrac{M}{m_i}$

ρ_M **:**
$\rightsquigarrow$ fonction donnant la classe d'un entier modulo M

$$\begin{array}{rccc} \rho_M : & \mathbb{Z} & \to & \mathbb{Z}/M\mathbb{Z} \\ & x & \mapsto & |x|_M \end{array}$$

résidu x_i**, pour tout** $i \in [\![1, n]\!]$ **et tout** $x \in [\![0, M[\![$ **:**
$\rightsquigarrow \rho_{m_i}(x) = |x|_{m_i}$ comme élément de $\mathbb{Z}/m_i\mathbb{Z}$ ou entier de $[\![0, m_i[\![$ (cf. Convention 1.2)

$\left|a^{-1}\right|_b$**, pour tout** $(a, b) \in \mathbb{Z} \times \mathbb{Z}^*$ **avec** $\mathrm{pgcd}(a, b) = 1$ **:**
$\rightsquigarrow$ l'inverse de $\rho_b(a)$ dans $\mathbb{Z}/b\mathbb{Z}^\times$, ou plus simplement son représentant dans $[\![0, b[\![$ (cf. Convention 1.2)

$\boldsymbol{x}_\mathcal{B}$**, pour tout** $x \in [\![0, M[\![$ **:**
$\rightsquigarrow$ le vecteur $(x_1, \ldots, x_n)$ des résidus de x dans $\mathcal{B}$

$\xi_{x,i,\mathcal{B}}$**, pour tout** $i \in [\![1, n]\!]$ **et tout** $x \in [\![0, M[\![$ **:**
$\rightsquigarrow \left|x_i M_i^{-1}\right|_{m_i}$

$\mathrm{sum}_\mathcal{B}(\boldsymbol{x}_\mathcal{B})$**, pour tout** $x \in [\![0, M[\![$ **:**
$\rightsquigarrow \sum_{i=1}^{n} \xi_{x,i,\mathcal{B}} M_i$

$\kappa_\mathcal{B}(\boldsymbol{x}_\mathcal{B})$**, pour tout** $x \in [\![0, M[\![$ **:**
$\rightsquigarrow \left\lfloor \dfrac{\mathrm{sum}_\mathcal{B}(\boldsymbol{x}_\mathcal{B})}{M} \right\rfloor$

$\varphi_{\mathcal{B}}$:

$\rightsquigarrow$ l'isomorphisme

$$\begin{array}{rccl} \varphi_{\mathcal{B}}: & \mathbb{Z}/M\mathbb{Z} & \to & \mathbb{Z}/m_1\mathbb{Z} \times \ldots \times \mathbb{Z}/m_n\mathbb{Z} \\ & |x|_M & \mapsto & \boldsymbol{x}_{\mathcal{B}} = \left(|x|_{m_1}, \ldots, |x|_{m_n}\right) \end{array}$$

où $\mathbb{Z}/M\mathbb{Z}$ est, sauf mention contraire, identifié à $[\![0, M[\![$

$\psi_{\mathcal{E},\mathcal{B}}$:

$\rightsquigarrow$ fonction de conversion dans le RNS, défini par la base $\mathcal{B}$, des éléments d'un ensemble $\mathcal{E}$

$$\begin{array}{rccl} \psi_{\mathcal{E},\mathcal{B}}: & \mathcal{E} & \to & \mathbb{Z}/m_1\mathbb{Z} \times \ldots \times \mathbb{Z}/m_n\mathbb{Z} \\ & x & \mapsto & \boldsymbol{x}_{\mathcal{B}} = \left(|x|_{m_1}, \ldots, |x|_{m_n}\right) \end{array}$$

Il suit immédiatement des notations précédentes la réécriture de l'Équation (1.3) qui, pour une base RNS $\mathcal{B}$, permet de reconstruire tout entier $x \in [\![0, M[\![$ depuis ses résidus $\boldsymbol{x}_{\mathcal{B}}$:

$$\forall x \in [\![0, M[\![,\ x = \sum_{i=1}^{n} \left|x_i M_i^{-1}\right|_{m_i} M_i \bmod M = \mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) \times M. \quad (1.5)$$

Précisément, l'Équation (1.5) donne la construction de l'application $\psi^{-1}_{[\![0,M[\![,\mathcal{B}}$ de la Définition 1.2 d'un RNS avec $\mathcal{E} = [\![0, M[\![$. De plus, vu la Convention 1.2, nous pouvons identifier $\psi_{[\![0,M[\![,\mathcal{B}}$ et $\varphi_{\mathcal{B}}$. Ainsi, nous obtenons la formule suivante que nous utiliserons souvent par la suite, où $\boldsymbol{x}_{\mathcal{B}}$ sont les résidus d'un élément x quelconque de l'intervalle dynamique de $\mathcal{B}$:

$$\varphi_{\mathcal{B}}^{-1}(\boldsymbol{x}_{\mathcal{B}}) = \mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) \times M \in [\![0, M[\![. \quad (1.6)$$

1.2 RNS et opérations arithmétiques élémentaires, éléments de complexité

L'injectivité de l'application $\psi_{\mathcal{E},\mathcal{B}}$ de la Définition 1.2 a été précédemment exploitée pour construire un système de représentation non positionnel, dans lequel un entier est représenté par ses restes modulaires. Un intérêt majeur de ces systèmes réside dans le fait qu'il est possible de transférer l'exécution des opérations arithmétiques standards d'addition, de soustraction, de multiplication, et sous certaines conditions de division, sur les éléments de l'ensemble fini initial d'entiers $\mathcal{E}$ directement sur leurs résidus, et ceci sans aucun mécanisme de propagation de retenue entre les résidus. L'arithmétique de base devient ainsi entièrement parallélisable sur les résidus.

1.2.1 Opérations élémentaires

La possibilité de transférer les opérations d'addition et de multiplication depuis l'ensemble $\mathcal{E}$ de la Définition 1.2 vers les résidus est une conséquence du fait que l'application $\varphi_{\mathcal{B}}$ du Théorème 1.1 des restes chinois est un morphisme d'anneaux. Cette propriété est particulièrement intéressante lorsque $\mathcal{E}$

présente une certaine stabilité sous l'application de ces opérations. C'est le cas par exemple dans un contexte cryptographique où les RNS sont utilisés pour représenter les éléments d'un corps $\mathbb{Z}/p\mathbb{Z}$ identifiés par leurs représentants de l'intervalle $\mathcal{E} = [\![0, p[\![$, où p est un grand nombre premier.

Un RNS $\mathcal{B}$ va permettre de représenter de manière univoque tout élément de $[\![0, p[\![$ dès que $M \geqslant p$. Par suite, pour un tel RNS, si par exemple le doublet $(e_1, e_2) \in [\![0, p[\![^2$ vérifie $e_3 = e_1 \star e_2 \in [\![0, p[\![$ pour une opération $\star \in \{+, -, \times\}$, alors

$$\psi^{-1}_{[\![0,p[\![,\mathcal{B}}\left(\{\psi_{[\![0,p[\![,\mathcal{B}}(e_1) \star \psi_{[\![0,p[\![,\mathcal{B}}(e_2)\}\right) = \{e_3\}.$$

Autrement dit, les résidus du résultat de l'opération $\star$ menée sur les résidus de e_1 et e_2 sont effectivement ceux de e_3. En effet, grâce au Théorème 1.1 des restes chinois nous pouvons écrire :

$$\begin{aligned}
\varphi^{-1}_{\mathcal{B}}\left(\{\psi_{[\![0,p[\![,\mathcal{B}}(e_1) \star \psi_{[\![0,p[\![,\mathcal{B}}(e_2)\}\right) &= \{\varphi^{-1}_{\mathcal{B}}\left(\psi_{[\![0,p[\![,\mathcal{B}}(e_1) \star \psi_{[\![0,p[\![,\mathcal{B}}(e_2)\right)\} \\
&\rightarrow \text{par bijectivité de } \varphi_{\mathcal{B}} \\
&= \{\varphi^{-1}_{\mathcal{B}} \circ \psi_{[\![0,p[\![,\mathcal{B}}(e_1) \star \varphi^{-1}_{\mathcal{B}} \circ \psi_{[\![0,p[\![,\mathcal{B}}(e_2)\} \\
&\rightarrow \text{car } \varphi^{-1}_{\mathcal{B}} \text{ est un morsphime} \\
&= \{\rho_M(e_1) \star \rho_M(e_2)\}.
\end{aligned}$$

Comme ρ_M est une simple réduction modulaire par M appliquée aux éléments de $[\![0, p[\![$, il vient $\rho_M(e_1) \star \rho_M(e_2) = \rho(e_1 \star e_2) = \rho(e_3)$. Et l'injectivité de ρ_M sur $\mathcal{E}$ permet d'obtenir le résultat attendu. Ainsi, si $\star \in \{+, -, \times\}$ et en notant $\mathcal{E}_\star = \{(e_1, e_2) \in [\![0, p[\![^2 |\ e_1 \star e_2 \in [\![0, p[\![\}$ le sous-ensemble de $[\![0, p[\![^2$ stable par $\star$, alors $\psi^{-1}_{[\![0,p[\![,\mathcal{B}}\left(\mathrm{Im}_{\psi_{[\![0,p[\![,\mathcal{B}}}(\mathcal{E}_\star)\right) \subseteq \mathcal{E}_\star$, ce qui signifie que la stabilité sous $\star$ est bien conservée dans le RNS.

Lorsqu'une opération $\star$ est exécutée sur un doublet de résidus d'éléments (e_1, e_2) de $[\![0, p[\![^2 \backslash \mathcal{E}_\star$ dans le RNS, cela peut avoir deux conséquences sur les résidus donnés par $\psi_{[\![0,p[\![,\mathcal{B}}(e_1) \star \psi_{[\![0,p[\![,\mathcal{B}}(e_2) = \boldsymbol{e}_{\mathcal{B}}$. La première concerne le fait où $\psi^{-1}_{[\![0,p[\![,\mathcal{B}}(\{\boldsymbol{e}_{\mathcal{B}}\}) = \varnothing$. Ce cas n'est pas surprenant puisque par définition du doublet (e_1, e_2), l'entier e_3 n'appartient pas à $[\![0, p[\![$. C'est le cas où e_3 est dans $[\![p, M[\![$. La seconde conséquence possible est celle où $\psi^{-1}_{[\![0,p[\![,\mathcal{B}}(\{\boldsymbol{e}_{\mathcal{B}}\}) \neq \varnothing$. Ce cas de figure peut se présenter lorsqu'un phénomène dit de « dépassement de capacité » apparaît.

Définition 1.3 *Soit $\boldsymbol{x}_{\mathcal{B}}$ et $\boldsymbol{y}_{\mathcal{B}}$ les résidus de $(x, y) \in [\![0, M[\![^2$, et $\star$ une opération arithmétique. Alors l'opération $\boldsymbol{x}_{\mathcal{B}} \star \boldsymbol{y}_{\mathcal{B}}$ provoque un dépassement de capacité si $x \star y \notin [\![0, M[\![$.*

Reprenant les notations précédentes avec de plus $p < M < 2p$ et $e_1 + e_2 \in [\![M, 2p[\![$ par exemple, alors le dépassement de capacité est illustré par le fait que :

$$\varphi^{-1}_{\mathcal{B}}(\boldsymbol{e}_{\mathcal{B}}) = |e_1 + e_2|_M = e_1 + e_2 - M.$$

Dans cet exemple, nous avons même $e_1 + e_2 - M < p$. Ainsi, $\psi^{-1}_{[\![0,p[\![,\mathcal{B}}(\{\boldsymbol{e}_{\mathcal{B}}\}) = \{e_1 + e_2 - M\} \neq \varnothing$. Mais en aucun cas les résidus $\boldsymbol{e}_{\mathcal{B}}$ du résultat n'ont de raison d'être ceux de l'entier $(e_1 + e_2) \bmod p$.

Il est important de souligner que l'apparition d'un dépassement de capacité sur un résultat intermédiaire d'une suite d'opérations arithmétiques élémentaires ne pose pas de problème de l'instant où le résultat final de cette

suite d'opérations est censé appartenir à $\mathcal{E}$ (cf. l'Exemple 1.2 suivant). Ainsi, l'associativité des opérations n'est pas remise en question par cette notion de dépassement de capacité.

Exemple 1.2 *Reprenant la base RNS $\mathcal{B} = \{3,5,7\}$ de l'Exemple 1.1, le calcul en RNS de la somme de produits $12 \times 6 + 13 \times 5 + 9 \times 8 + 11 \times 4 = 253$ donne :*

$$\begin{array}{lll}
\psi_{[\![0,105[\![,\mathcal{B}}(12) \times \psi_{[\![0,105[\![,\mathcal{B}}(6) & = (|0|_3, |2|_5, |5|_7) \times (|0|_3, |1|_5, |6|_7) & = (|0|_3, |2|_5, |2|_7) \\
+\psi_{[\![0,105[\![,\mathcal{B}}(13) \times \psi_{[\![0,105[\![,\mathcal{B}}(5) & = (|1|_3, |3|_5, |6|_7) \times (|2|_3, |0|_5, |5|_7) & = (|2|_3, |0|_5, |2|_7) \\
+\psi_{[\![0,105[\![,\mathcal{B}}(9) \times \psi_{[\![0,105[\![,\mathcal{B}}(8) & = (|0|_3, |4|_5, |2|_7) \times (|2|_3, |3|_5, |1|_7) & = (|0|_3, |2|_5, |2|_7) \\
+\psi_{[\![0,105[\![,\mathcal{B}}(11) \times \psi_{[\![0,105[\![,\mathcal{B}}(4) & = (|2|_3, |1|_5, |4|_7) \times (|1|_3, |4|_5, |4|_7) & = (|2|_3, |4|_5, |2|_7) \\
= & & (|1|_3, |3|_5, |1|_7)\,.
\end{array}$$

Or, à cause du dépassement de capacité, $\psi^{-1}_{[\![0,105[\![,\mathcal{B}}(|1|_3, |3|_5, |1|_7) = 43 = 253 - 2 \times 105$. Mais en divisant maintenant 253 par 11, il vient :

$$\begin{aligned}
(|1|_3, |3|_5, |1|_7) \times \left(\left|11^{-1}\right|_3, \left|11^{-1}\right|_5, \left|11^{-1}\right|_7\right) &= (|1|_3, |3|_5, |1|_7) \times (|2|_3, |1|_5, |2|_7) \\
&= (|2|_3, |3|_5, |2|_7) \\
&= \psi_{[\![0,105[\![,\mathcal{B}}(23)\,.
\end{aligned}$$

Ainsi, le calcul de la quantité $\frac{12\times6+13\times5+9\times8+11\times4}{11}$ dans la base RNS $\mathcal{B}$ donne le résultat attendu, malgré le dépassement de capacité du numérateur.

Le cas de la division euclidienne classique sur les entiers reste délicat (Hitz et Kaltofen 1995, Bajard et al. 1998; 2003). Cela tient à la difficulté d'effectuer une comparaison. En ce qui concerne les divisions exactes, la situation est plus simple. Il suffit de constater que si $m \in \mathbb{N}^*$ est un modulus, et si le triplet $(a,b,c) \in [\![0,m[\![^3$ vérifie $a = b \times c$, alors lorsque b premier avec m nous avons :

$$\left|a \times \left|b^{-1}\right|_m\right|_m = |c|_m = \left|\frac{a}{b}\right|_m.$$

Dans un RNS doté d'une base $\mathcal{B} = \{m_1, \ldots, m_n\}$, cette approche nécessite un diviseur premier avec M. Si l'entier b n'est pas premier avec certains moduli de $\mathcal{B}$, la division reste possible sur les autres résidus. Si par exemple $c < M_i = \dfrac{M}{m_i}$ et $b = m_i$, l'ensemble des résidus $\left|a \times \left|m_i^{-1}\right|_{m_j}\right|_{m_j}$ pour $j \neq i$ définit complètement c dans la sous-base $\mathcal{B}\backslash\{m_i\}$. Retrouver le reste de c modulo m_i à partir de ces $n-1$ résidus est possible en utilisant une procédure dite d'extension de base. Ces procédures sont coûteuses. Elles peuvent également être exploitées pour définir une opération de réduction modulaire. Il devient alors possible d'exécuter des opérations de type division euclidienne via la formule $a = \lfloor\frac{a}{b}\rfloor \times b + (a \bmod b)$, où $\lfloor\frac{a}{b}\rfloor = \frac{a-(a \bmod b)}{b}$ est cette fois une division exacte réalisable directement sur les résidus.

1.2.2 Influence du RNS sur les bornes de complexité

Les RNS définissent un cadre arithmétique efficace. Les opérations arithmétiques peuvent être effectuées sur les résidus de manière totalement parallèle, c'est-à-dire sans propagation de retenue. La complexité des opérations devient

alors seulement contingente de la taille des moduli choisis. Celle-ci est asymptotiquement équivalente à $\log(M_{\mathcal{E}})$, où $M_{\mathcal{E}}$ n'est autre que la taille de l'ensemble des données que l'on souhaite représenter en RNS (cf. Définition 1.2). En effet, si $\mathcal{B}$ est la base RNS telle que M est la primorielle $p_n\#$, c'est-à-dire le produit des n premiers nombres premiers $\prod_{i=1}^{n} p_i$, alors Erdös (1934) a montré en particulier que $M < 4^{p_n}$. Plus précisément, un résultat concernant la première fonction de Chebyshev définie par $\theta(x) = \sum_{i=1}^{\pi(x)} \log(p_i)$, où $\pi(x)$ est le nombre de nombres premiers inférieurs ou égaux à x, permet d'obtenir l'estimation suivante (Rosser et Schoenfeld 1962) :

$$\left| \frac{\theta(p_n)}{p_n} - 1 \right| \in \Theta\left(\frac{1}{2p_n \log(p_n)} \right).$$

Comme $\log(M) = \theta(p_n)$, il en découle que $\log(M) \underset{n\to+\infty}{\sim} p_n$. Le choix des moduli étant seulement limité par la contrainte de conserver une entropie suffisante, il en résulte que si la complexité des opérations d'addition et de multiplication dans $\mathcal{E}$ est de $\mathcal{O}(f(\log M_{\mathcal{E}}))$ opérations binaires, alors en utilisant l'équivalence asymptotique classique $\pi(p_n) \underset{n\to+\infty}{\sim} \frac{p_n}{\ln(p_n)}$ le RNS permet d'abaisser cette borne de complexité à $\mathcal{O}\left(\frac{\log(M_{\mathcal{E}})}{\log\log(M_{\mathcal{E}})} \times f(\log\log(M_{\mathcal{E}})) \right)$. De plus, le facteur $\frac{\log(M_{\mathcal{E}})}{\log\log(M_{\mathcal{E}})}$ peut être « court-circuité » par la parallélisation des calculs sur les résidus.

Lors de l'analyse de complexité d'algorithmes basés sur le RNS, il est pratique et usuel d'exprimer celles-ci en terme de nombre d'additions et/ou de multiplications de mots ou digits RNS dont la taille $\beta = 2^r$ majore celle des moduli de la base RNS utilisée. r s'avère être en pratique un paramètre déterminant dans le choix d'une base, dépendant de facteurs extérieurs comme par exemple la taille du mot machine, ou du registre d'un multiplieur de FPGA. De telles opérations de base sont notées $\texttt{EAdd}_\beta$ et $\texttt{EMul}_\beta$ (ou simplement `EAdd` et `EMul` si aucune ambiguïté n'est possible) pour addition élémentaire et multiplication élémentaire. Les notations $\text{Mul}_\beta(a,b)$ et $\text{Add}_\beta(a,b)$, qui représentent respectivement une multiplication et une addition entre deux entiers possédant respectivement a β-mots et b β-mots, pourront aussi être utilisées. Par exemple, si β est la taille d'un mot/digit RNS, une `EMul` n'est autre que $\text{Mul}_\beta(1,1)$. La notation simplifiée Add_β (resp. Mul_β) désignera $\text{Add}_\beta(1,1)$ (resp. $\text{Mul}_\beta(1,1)$).

Comme les opérations élémentaires `EAdd` et `EMul` correspondent aux additions et multiplications sur des entiers représentants des résidus dans les anneaux quotients $\mathbb{Z}/m_i\mathbb{Z}$, les opérations modulaires modulo les m_i sont également considérées comme « atomiques » dans les analyses de complexité. Les notations `EMul` et `MMEl`, pour addition/multiplication modulaire élémentaire, seront donc aussi utilisées. Le coût réel de ces opérations de base peut être maîtrisé par le biais d'un choix de moduli particuliers pour lesquels la réduction modulaire est aisée.

1.2.3 Sur le choix des moduli

Utiliser des ensembles de moduli pour lesquels la réduction modulaire est particulièrement peu coûteuse est un point central car cela participe de l'efficacité du RNS utilisé. Les nombres de Mersenne par exemple, de la forme

$2^r - 1$, sont un véritable cas d'école. Dans ce cas, une réduction modulaire se réduit à de simples décalages et additions. Néanmoins, étant donné que deux tels nombres $2^{r_1} - 1$ et $2^{r_2} - 1$ sont premiers entre eux si, et seulement si, r_1 et r_2 le sont aussi, le recours à cette classe de nombre pour créer des bases RNS devient rapidement limitant. C'est le cas par exemple lorsqu'il s'agit d'avoir des moduli de taille égale pour une implantation matérielle optimisée et flexible. Des classes plus spécifiques de moduli de la forme $\{2^r - 1, 2^r, 2^r + 1\}$ (Gallaher et al. 1997), $\{2^{r+1} - 1, 2^r, 2^r - 1\}$ (Mohan 2007), $\{2^r \pm 3, 2^r \pm 1, 2^r\}$ (Mohan 2008, Sheu et al. 2004), $\{2^r \pm t\}$ (Matutino et al. 2012) ou bien $\{2^r - 2^{t_i} \pm 1\}$ (Bajard et al. 2009), ont été intensivement étudiées dans une optique d'optimisation pour l'implantation matérielle.

Algorithme 2 : Réduction modulaire efficace pour un modulus pseudo Mersenne

Données : Un modulus $m = 2^r - c$, $r \in \mathbb{N}^*$, $c \in [\![0, \sqrt{m}[\![$, et un entier $a \in [\![0, 2^{2r}[\![$.
Résultat : $|a|_m \in [\![0, m[\![$.

1 **début**
2 $(a_1, a_2) \leftarrow (a >> r,\ a \texttt{ and } (2^r - 1))$
$/\star\ a = a_1 \times 2^r + a_2,\ a_1, a_2 < 2^r\ \star/$
3 $b \leftarrow a_1 \times c$
4 $(b_1, b_2) \leftarrow (b >> r,\ b \texttt{ and } (2^r - 1))$
$/\star\ b = b_1 \times 2^r + b_2\ ;\ b_1 < c, b_2 < 2^r\ \star/$
5 $d \leftarrow b_1 \times c$
$/\star\ d < c^2 < m\ \star/$
6 $r \leftarrow (a_2 + b_2 + d) \bmod m$
$/\star\ r < 2^{r+1} + m = 3m + c\ \star/$

Plus généralement, l'utilisation de nombres dits pseudo Mersenne, *i.e.* de la forme $m = 2^r - c$ où $c < \sqrt{m}$, offre une plus grande latitude de choix tout en garantissant une bonne efficacité. Ceux-ci possèdent en effet l'avantage de rendre possible une réduction modulaire (Algorithme 2) en seulement une multiplication $\mathrm{Mul}_{2^r}\left(1, \frac{1}{2}\right)$ entre deux opérandes de taille r et $\frac{r}{2}$ (étape 3), une seconde $\mathrm{Mul}_{2^r}\left(\frac{1}{2}, \frac{1}{2}\right)$ (étape 5), deux additions Add_{2^r} (étape 6), et la réduction d'un entier $< 3m + c$ (étape 6), ainsi que de simples décalages et `and` bit-à-bit (étapes 1 et 4). Dans un tel cas, nous avons donc :

$$1\ \texttt{MMEl}_\beta \sim \frac{7}{4}\ \texttt{EMul}_\beta + 2\ \texttt{EAdd}_\beta. \tag{1.7}$$

Il est aussi possible d'élargir encore la gamme de moduli utilisables en introduisant d'autres techniques de réduction modulaire efficaces.

Dans le cas de moduli qui ne sont pas pseudo Mersenne, des réductions modulaires efficaces, comme celles de Barrett (1986) ou de Montgomery (1985), peuvent être utilisées. Ceci permet d'augmenter la classe de moduli de taille donnée sans restriction. Par exemple, Bajard et Merkiche (2014) proposent une implantation d'un algorithme de réduction modulaire RNS où les réductions élémentaires se basent sur la réduction de Montgomery (rappelée en 1.3.2).

1.2.4 Un système positionnel associé, le Mixed Radix System (MRS)

À toute base RNS ordonnée $\mathcal{B} = \{m_1, \ldots, m_n\}$ est associé un système de représentation positionnel non standard de l'ensemble des nombres de l'intervalle dynamique dénommé Mixed Radix System (Szabo et Tanaka 1967). Un nombre $x \in [\![0, M[\![$, de résidus $x_\mathcal{B}$ dans $\mathcal{B}$, est représenté par n coefficients $(\tilde{x}_1, \ldots, \tilde{x}_n) \in \mathbb{Z}/m_1\mathbb{Z} \times \mathbb{Z}/m_n\mathbb{Z}$ dans une base dite MRS, constituée des éléments $\{1, m_1, \ldots, \prod_{i=1}^{n-1} m_i\}$. Ces coefficients MRS permettent alors de retrouver x de la manière suivante :

$$x = \tilde{x}_1 + \tilde{x}_2 \times m_1 + \ldots + \tilde{x}_n \times m_1 \ldots m_{n-1} = \sum_{i=1}^{n} \tilde{x}_i \prod_{j=1}^{i-1} m_j. \tag{1.8}$$

La construction des coefficients MRS se déduit des résidus par une approche de type interpolation de Newton, ou différences divisées. Cette construction est séquentielle et se base sur l'Équation (1.8). Si nous notons $\tilde{m}_{i,j} = \left|m_i^{-1}\right|_{m_j}$, alors les coefficients sont les suivants :

$$\begin{cases} \tilde{x}_1 = x_1 \\ \tilde{x}_2 = \left|(x_2 - \tilde{x}_1)\, \tilde{m}_{1,2}\right|_{m_2} \\ \quad \vdots \\ \tilde{x}_n = \left|(\ldots (x_n - \tilde{x}_1)\, \tilde{m}_{1,n} - \ldots - \tilde{x}_{n-1})\, \tilde{m}_{n-1,n}\right|_{m_n}. \end{cases} \tag{1.9}$$

Ainsi l'égalité (1.8) est bien vérifiée. En effet, comme $0 \leqslant \tilde{x}_i \leqslant m_i - 1$, il vient d'une part que cette somme de produits est bien un élément de l'intervalle dynamique :

$$0 \leqslant \sum_{i=1}^{n} \tilde{x}_i \prod_{j=1}^{i-1} m_j \leqslant \sum_{i=1}^{n} \left(\prod_{j=1}^{i} m_j - \prod_{j=1}^{i-1} m_j \right) = \prod_{j=1}^{n} m_j - 1 = M - 1.$$

D'autre part, l'Équation (1.8) étant vérifiée modulo m_i pour tout $i \in [\![1, n]\!]$, la bijectivité du RNS permet alors de conclure à l'égalité.

Sous réserve de pouvoir précalculer et mémoriser les $\frac{n(n-1)}{2}$ inverses $\tilde{m}_{i,j}$, le calcul des coefficients MRS nécessite donc $\frac{n(n-1)}{2}$ `MME1` et `AME1`. La construction (1.9) autorise un certain degré de parallélisation (Gbolagade et Cotofana 2009). Lorsqu'un coefficient $\tilde{x}_i$ est calculé, il peut être propagé sur les $n - i$ résidus suivants. Ainsi, le calcul total peut s'effectuer en $n - 1$ étapes parallèles (cf. Algorithme 3). Une autre approche permet de diminuer le nombre de précalculs à mémoriser à $n - 1$ valeurs. En notant $\tilde{m}_i = \left| \prod_{j=1}^{i-1} \left|m_j^{-1}\right|_{m_i} \right|_{m_i}$ et en factorisant le membre de droite de la i-ième ligne du système d'Équations (1.9) par $\tilde{m}_i$, alors nous obtenons :

$$\tilde{x}_i = \left|(x_i - \tilde{x}_1 - m_1\, (\tilde{x}_2 - m_2\, (\tilde{x}_3 - \ldots)\ldots))\, \tilde{m}_i\right|_{m_i} \tag{1.10}$$

La complexité demeure identique puisque $\frac{n(n-1)}{2}$ `MME1` et autant de `AME1` sont nécessaires. Néanmoins, la parallélisation de ces opérations est perdue puisque le calcul du coefficient $\tilde{x}_i$ requiert d'accéder successivement à $\tilde{x}_{i-1}, \ldots, \tilde{x}_1$.

La reconstruction de x dans une représentation positionnelle standard peut se faire en utilisant un schéma classique de type Hörner :

$$x = \tilde{x}_1 + m_1 \left(\tilde{x}_2 + m_2 \left(\tilde{x}_3 + \ldots\right)\ldots\right). \tag{1.11}$$

Un tel calcul nécessite $n-1$ `MMEl` et `AMEl`.

Le principal avantage de ce système par rapport au RNS est son caractère positionnel. Si x et y sont des éléments de l'intervalle dynamique $[\![0, M[\![$ et sont représentés par leurs coefficients MRS $(\tilde{x}_1, \ldots, \tilde{x}_n)$ et $(\tilde{y}_1, \ldots, \tilde{y}_n)$, alors $x < y$ si, et seulement si, il existe $i \in [\![1, n]\!]$ pour lequel $\tilde{x}_i < \tilde{y}_i$ et $\tilde{x}_j \leqslant \tilde{y}_j$ pour tout $j \in [\![i+1, n]\!]$. Pour montrer cela il suffit de constater que pour tout $i \in [\![1, n]\!]$,

$$(m_1 - 1) + m_1 (m_2 - 1) + \ldots + m_1 \ldots m_{i-1} (m_i - 1) = m_1 \ldots m_i - 1.$$

Comme $y - x = \sum_{t=1}^{n} (\tilde{y}_t - \tilde{x}_t) \prod_{u=1}^{t-1} m_u$ et $\left|\sum_{t=1}^{i-1} (\tilde{y}_t - \tilde{x}_t) \prod_{u=1}^{t-1} m_u\right| < \prod_{t=1}^{i-1} m_t$, la comparaison s'effectue successivement sur les coefficients, du n-ième au premier.

Remarque 1.4 *Pour effectuer une comparaison de deux nombres RNS, le calcul de leurs coefficients MRS suffit. De plus, la procédure de conversion entre les deux représentations ne fait intervenir que des opérations sur des nombres de la taille des résidus.*

Algorithme 3 : MRScoeff $(\mathcal{B}, (x_1, \ldots, x_n)_{\mathcal{B}})$

Données : $\mathcal{B} = \{m_1, \ldots, m_n\}$ une base RNS, $(x_1, \ldots, x_n)_{\mathcal{B}}$ les résidus dans $\mathcal{B}$ d'un nombre $x \in [\![0, M[\![$. Précalculs : $\tilde{m}_{i,j} = \left|m_i^{-1}\right|_{m_j}$ pour tout $i \in [\![1, n-1]\!]$ et tout $j \in [\![i+1, n]\!]$.

Résultat : $(\tilde{x}_1, \ldots, \tilde{x}_n) \in \mathbb{Z}/m_1\mathbb{Z} \times \ldots \times \mathbb{Z}/m_n\mathbb{Z}$ les coefficients MRS de x dans la base MRS $\{1, m_1, \ldots, m_1 \ldots m_{n-1}\}$.

```
1 début
2     (x̃_1, ..., x̃_n) ← (x_1, ..., x_n)
3     pour i ← 2 à n faire
4         pour j ← i à n faire
              /* en parallèle sur les n − i + 1 derniers
              résidus */
5             x̃_j ← |(x̃_j − x̃_{i−1}) m̃_{i−1,j}|_{m_j}
```

Exemple 1.3 *Le contexte est toujours celui de l'Exemple 1.1. Le calcul des coefficients MRS de* $\psi(23) = (|2|_3, |3|_5, |2|_7)$ *est le suivant :*

$$\begin{cases} \tilde{x}_1 = 2 \\ \tilde{x}_2 = \left|(3-2) \times 3^{-1}\right|_5 = 2 \\ \tilde{x}_3 = \left|\left((2-2) \times 3^{-1} - 2\right) \times 5^{-1}\right|_7 = 1. \end{cases}$$

Ainsi,

$$\tilde{x}_1 + m_1 \tilde{x}_2 + m_1 m_2 \tilde{x}_3 = 2 + 3 \times 2 + 3 \times 5 \times 1 = 23.$$

Pour illustrer la comparaison en MRS, considérons par exemple le nombre dont les coefficients MRS sont $(1, 3, 1)$*. Alors il est strictement supérieur à* 23*. En effet, ce nombre n'est autre que* $1 + 3 \times 3 + 1 \times 3 \times 5 = 25$*.*

1.3 Conversions de bases et réduction modulaire RNS

1.3.1 Extensions/Conversions de base

Les discussions précédentes ont en particulier permis de souligner le fait que certaines opérations comme la division ou la réduction modulaire sont difficiles à mettre en œuvre dans le cadre du RNS. Pour résoudre ce problème, il est nécessaire de disposer de procédures efficaces de conversion entre RNS et représentation alternative, ou entre deux RNS.

Dans cette section, deux bases RNS $\mathcal{B} = \{m_1, \ldots, m_n\}$ et $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$ sont considérées. Étant donné un entier x et ses résidus $\boldsymbol{x}_\mathcal{B}$ dans $\mathcal{B}$, une conversion de base est une procédure qui permet de calculer les résidus de $|x|_M$ dans la base $\mathcal{B}'$. Le terme extension s'explique par le fait qu'au final seront obtenus les résidus de $|x|_M$ dans la base étendue $\mathcal{B} \cup \mathcal{B}'$. L'efficacité d'une procédure de conversion de base RNS se mesure notamment par le fait qu'elle ne recourt pas à un retour transitoire à la représentation binaire de grands entiers de la taille de M. Ceci briserait l'indépendance des résidus qui caractérise le RNS à cause de la réapparition de retenues.

Définition 1.4 *Une procédure de conversion/extension de base entre deux bases RNS $\mathcal{B}$ et $\mathcal{B}'$ est une procédure prenant en entrée un ensemble de résidus $\boldsymbol{x}_\mathcal{B}$ dans la base $\mathcal{B}$, représentant un entier x de l'intervalle dynamique de $\mathcal{B}$, et rendant en sortie les résidus $\boldsymbol{x}_{\mathcal{B}'}$ de x dans la base $\mathcal{B}'$. Une telle procédure est notée* $\mathrm{Bex}(\mathcal{B}, \mathcal{B}', \boldsymbol{x}_\mathcal{B})$.

De toute procédure de conversion de base peut se déduire une conversion de la représentation RNS vers la représentation positionnelle binaire standard. Dans ce cas, la procédure qui en découle sera notée $\mathrm{Bex}(\mathcal{B}, \boldsymbol{x}_\mathcal{B})$.

Ces deux procédures vérifient par définition :

$$\mathrm{Bex}\left(\mathcal{B}, \mathcal{B}', \boldsymbol{x}_\mathcal{B}\right) = \varphi_{\mathcal{B}'}\left(\mathrm{Bex}\left(\mathcal{B}, \boldsymbol{x}_\mathcal{B}\right)\right)$$

Remarque 1.5 *Certaines conversions de base que nous allons voir par la suite peuvent ne pas donner une réduction complète, et plus précisément être telles que* $\mathrm{Bex}(\mathcal{B}, \boldsymbol{x}_\mathcal{B}) = x + kM$, *avec un entier $k \in \mathbb{Z}^*$. Par conséquent, suivant la procédure de conversion utilisée, si $x \in [\![0, M[\![$, il sera possible d'avoir* $\mathrm{Bex}(\mathcal{B}, \boldsymbol{x}_\mathcal{B}) = x + kM \neq x = \varphi_\mathcal{B}^{-1}(\boldsymbol{x}_\mathcal{B})$ *et donc* $\mathrm{Bex}(\mathcal{B}, \mathcal{B}', \boldsymbol{x}_\mathcal{B}) = \varphi_{\mathcal{B}'}(x + kM) \neq \varphi_{\mathcal{B}'}(x)$.

A contrario, si la conversion $\mathrm{Bex}(\mathcal{B}, \mathcal{B}', .)$ *est toujours complètement réduite, alors elle s'identifie à $\varphi_{\mathcal{B}'} \circ \varphi_\mathcal{B}^{-1}$.*

Basées sur le MRS

Une première technique consiste à utiliser un système MRS associé à la base $\mathcal{B}$ (Szabo et Tanaka 1967). Comme détaillé précédemment, le calcul des coefficients MRS de x s'effectue directement dans la base $\mathcal{B}$. Une fois ceux-ci obtenus, le calcul des résidus de x dans la base $\mathcal{B}'$ se réalise en parallèle en utilisant la Formule (1.11) de reconstruction de x à partir de ses coefficients MRS, ou bien, s'il est possible de précalculer les produits partiels $\left|\prod_{j=1}^{i} m_j\right|_{m'_t}$ pour tout $(i, t) \in [\![1, n]\!] \times [\![1, \ell]\!]$, simplement par :

$$|x|_{m'_t} = \left|\sum_{i=1}^{n} x_i \times \left|\prod_{j=1}^{i} m_j\right|_{m'_t}\right|_{m'_t}.$$

Si le coût reste le même, cette formule permet de diminuer le nombre d'étapes d'une conversion dans un contexte de parallélisation totale entre $\mathcal{B}$ et $\mathcal{B}'$. En effet, dès qu'un coefficient $\tilde{x}_i$ est obtenu, il peut être propagé directement sur chaque résidu dans la nouvelle base $\mathcal{B}'$, ce qui permet une conversion complète en n étapes de calculs sur la base étendue $\mathcal{B} \cup \mathcal{B}'$.

Une autre manière de voir cette conversion entre $\mathcal{B} = \{m_1, \ldots, m_n\}$ et $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$, une fois les coefficients MRS de x calculés, consiste en le produit matrice-vecteur suivant :

$$\begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_\ell \end{pmatrix} = \begin{pmatrix} 1 & |m_1|_{m'_1} & \ldots & |m_1 \ldots m_{n-1}|_{m'_1} \\ 1 & |m_1|_{m'_2} & \ldots & |m_1 \ldots m_{n-1}|_{m'_2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & |m_1|_{m'_\ell} & \ldots & |m_1 \ldots m_{n-1}|_{m'_\ell} \end{pmatrix} \times \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix} \bmod \begin{pmatrix} m'_1 \\ m'_2 \\ \vdots \\ m'_\ell \end{pmatrix}$$

Algorithme 4 : $\text{Bex}_{mrs}\left(\mathcal{B}, \mathcal{B}', (x_1, \ldots, x_n)_{\mathcal{B}}\right)$

Données : $\mathcal{B} = \{m_1, \ldots, m_n\}$ et $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$ deux bases RNS, $\boldsymbol{x}_{\mathcal{B}}$ les résidus dans $\mathcal{B}$ d'un nombre $x \in [\![0, M[\![$.

Résultat : $\boldsymbol{x}_{\mathcal{B}'}$ les résidus dans $\mathcal{B}'$ de x.

1 **début**
2 $\quad (\tilde{x}_1, \ldots, \tilde{x}_n) \leftarrow \text{MRScoeff}\left(\mathcal{B}, (x_1, \ldots, x_n)_{\mathcal{B}}\right)$ `/* cf. Algo. 3 */`
3 $\quad \left(x'_1, \ldots, x'_\ell\right) \leftarrow (\tilde{x}_n, \ldots, \tilde{x}_n)$
4 $\quad$ **pour** $i \leftarrow 1$ **à** ℓ **faire** `/* en parallèle dans la base` $\mathcal{B}'$ `*/`
5 $\quad\quad$ **pour** $j \leftarrow n-1$ **à** 1 **faire**
6 $\quad\quad\quad x'_i \leftarrow \left|x'_i m_j + \tilde{x}_j\right|_{m'_i}$

La complexité de la procédure de conversion de base nommée Bex_{mrs} et donnée par l'Algorithme 4 est donc :

$$\mathcal{C}\left(\text{Bex}_{mrs}\left(\mathcal{B}, \mathcal{B}', .\right)\right) = \frac{n\,(n-1)}{2}\left(\texttt{MMEl}_{\mathcal{B}} + \texttt{AMEl}_{\mathcal{B}}\right) + \ell\,(n-1)\left(\texttt{MMEl}_{\mathcal{B}'} + \texttt{AMEl}_{\mathcal{B}'}\right). \tag{1.12}$$

En termes de multiplications pour des bases ayant des moduli de même taille β, la complexité se résume donc par :

$$\frac{(n+2\ell)\,(n-1)}{2}\ \texttt{MMEl}_{\beta}. \tag{1.13}$$

Il est entendu que cette borne de complexité vaut pour tout choix de bases $\mathcal{B}$ et $\mathcal{B}'$. Des améliorations peuvent êtres apportées pour améliorer cette complexité en choisissant précautionneusement $\mathcal{B}$ et $\mathcal{B}'$ de manière à ce que les moduli des deux bases et les inverses $|m_i|^{-1}_{m'_j}$ soient creux, par exemple de la forme $2^r - 2^t \pm 1$, limitant ainsi le coût de certaines multiplications à quelques additions (Bajard et al. 2009).

En considérant la Remarque 1.5, il est utile de préciser pour la suite que cette conversion a l'avantage de toujours faire intervenir une réduction complète modulo M. Ainsi,

$$\text{Bex}_{mrs}\left(\mathcal{B}, \mathcal{B}', .\right) \equiv \varphi_{\mathcal{B}'} \circ \varphi_{\mathcal{B}}^{-1}. \tag{1.14}$$

Basées sur le TRC

Ces procédures de conversion découlent de la preuve constructive concernant la surjectivité de l'application $\varphi_{\mathcal{B}}$ du Théorème 1.1 des restes chinois, et qui est à l'origine de l'Équation (1.6), rappelée ici, pour un entier x dans $[\![0, M[\![$:

$$x = \varphi_{\mathcal{B}}^{-1}(\boldsymbol{x}_{\mathcal{B}}) = \text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) - \kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) \times M \in [\![0, M[\![, \tag{1.15}$$

où $\text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) = \sum_{i=1}^{n} \xi_{x,i,\mathcal{B}} M_i$, $\xi_{x,i,\mathcal{B}} = \left| x_i M_i^{-1} \right|_{m_i}$ et $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) = \lfloor \frac{\text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})}{M} \rfloor = \lfloor \sum_{i=1}^{n} \frac{\xi_{x,i,\mathcal{B}}}{m_i} \rfloor$.

Remarque 1.6 *Comme $0 \leqslant \xi_{x,i,\mathcal{B}} < m_i$ pour tout i et tout $x \in [\![0, M[\![$, alors $0 \leqslant \text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) < nM$. Par conséquent, $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) \in [\![0, n-1]\!]$.*

Tout l'enjeu pour définir une procédure de conversion de base efficace basée sur l'Équation (1.15) réside dans le fait de pouvoir calculer le coefficient $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ sans devoir passer par le calcul du grand entier $\text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ suivi de sa réduction modulo M. Pour ce faire, deux approches se démarquent particulièrement dans l'état de l'art. La première propose de retrouver $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ en rajoutant de la redondance à la base $\mathcal{B}$. La seconde utilise une approximation de la fraction $\frac{\text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})}{M}$.

Méthode de Shenoy et Kumaresan Shenoy et Kumaresan (1989) proposent une procédure pour retrouver la valeur $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ en utilisant l'Équation (1.15). L'idée est la suivante. Si un modulus supplémentaire m_{sk} est adjoint à la base $\mathcal{B}$ et est tel que M est inversible modulo m_{sk}, et si le résidu de x modulo m_{sk} peut être obtenu avant la conversion, alors il est possible d'inverser l'Équation (1.15) dans $\mathbb{Z}/m_{sk}\mathbb{Z}$, ce qui permet de calculer $|\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})|_{m_{sk}}$:

$$|\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})|_{m_{sk}} = \left| (\text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) - x_{sk}) M^{-1} \right|_{m_{sk}}. \tag{1.16}$$

Mais d'après la Remarque 1.6, $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) \in [\![0, n-1]\!]$. Par conséquent, si m_{sk} est choisi premier à M et tel que $m_{sk} \geqslant n$, alors $|\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})|_{m_{sk}} = \kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$.

Fondamentalement, il s'agit d'augmenter l'entropie de la base de départ par l'ajout d'une redondance $m_{sk} \geqslant n$. La quantité $\text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ appartient alors à l'intervalle dynamique de la base étendue $\mathcal{B} \cup \{m_{sk}\}$. Puis il reste simplement à calculer la division exacte suivante dans $\mathbb{Z}/m_{sk}\mathbb{Z}$ (ce qui est possible puisque $x_{sk} = x \bmod m_{sk}$ est alors connu) :

$$\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) = \frac{\text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) - x}{M}.$$

L'Algorithme 5 met en œuvre la conversion de base résultant de cette approche. La complexité de cette procédure est :

$$\begin{aligned} \mathcal{C}\left(\text{Bex}_{sk}\left(\mathcal{B} \cup \{m_{sk}\}, \mathcal{B}', ., .\right)\right) = & n \ \texttt{MMEl}_{\mathcal{B}} + \ell(n+1) \ \texttt{MMEl}_{\mathcal{B}'} + n\ell \ \texttt{AMEl}_{\mathcal{B}'} \\ & + (n+1) \ \texttt{MMEl}_{\{m_{sk}\}} + n \ \texttt{AMEl}_{\{m_{sk}\}} \end{aligned} \tag{1.17}$$

Ainsi, en considérant que les moduli de $\mathcal{B}$ et $\mathcal{B}'$ ont une taille identique β, et comme il suffit que m_{sk} soit de l'ordre de n, alors la complexité en termes de

Algorithme 5 : $\mathrm{Bex}_{sk}\left(\mathcal{B} \cup \{m_{sk}\}, \mathcal{B}', (\boldsymbol{x}_{\mathcal{B}}, x_{sk})\right)$

Données : $\mathcal{B} \cup \{m_{sk}\} = \{m_1, \ldots, m_n\} \cup \{m_{sk}\}$ et $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$ deux bases RNS, avec $m_{sk} \geqslant n$, $(\boldsymbol{x}_{\mathcal{B}}, x_{sk})$ les résidus dans $\mathcal{B} \cup \{m_{sk}\}$ d'un nombre $x \in [\![0, M[\![$. Précalculs possibles : $a_{i,j} = |M_i|_{m'_j}$, $a_{i,sk} = |M_i|_{m_{sk}}$ pour tout $(i,j) \in [\![1,n]\!] \times [\![1,\ell]\!]$, $|M|_m$ pour tout $m \in \mathcal{B}'$, $\left|M_i^{-1}\right|_{m_i}$ pour tout $i \in [\![1,n]\!]$, et $\left|M^{-1}\right|_{m_{sk}}$.

Résultat : $\boldsymbol{x}_{\mathcal{B}'}$, les résidus dans $\mathcal{B}'$ de x.

1 **début**

2 $$\begin{pmatrix} \xi_{x,1,\mathcal{B}} \\ \xi_{x,2,\mathcal{B}} \\ \vdots \\ \xi_{x,n,\mathcal{B}} \end{pmatrix} \leftarrow \begin{pmatrix} x_1 \left|M_1\right|_{m_1}^{-1} \\ \vdots \\ x_n \left|M_n\right|_{m_n}^{-1} \end{pmatrix} \bmod \begin{pmatrix} m_1 \\ \vdots \\ m_n \end{pmatrix}$$

`/* en // dans` $\mathcal{B}$ `*/`

3 $$\begin{pmatrix} \kappa \\ x'_1 \\ \vdots \\ x'_\ell \end{pmatrix} \leftarrow \begin{pmatrix} a_{1,sk} & \ldots & a_{n,sk} \\ a_{1,1} & \ldots & a_{n,1} \\ \vdots & & \\ a_{1,\ell} & \ldots & a_{n,\ell} \end{pmatrix} \begin{pmatrix} \xi_{x,1,\mathcal{B}} \\ \xi_{x,2,\mathcal{B}} \\ \vdots \\ \xi_{x,n,\mathcal{B}} \end{pmatrix} \bmod \begin{pmatrix} m_{sk} \\ m'_1 \\ \vdots \\ m'_\ell \end{pmatrix}$$

`/* en // dans` $\{m_{sk}\} \cup \mathcal{B}'$ `*/`

4 $$\kappa \leftarrow \left|(\kappa - x_{sk}) \left|M^{-1}\right|_{m_{sk}}\right|_{m_{sk}}$$

5 $$\begin{pmatrix} x'_1 \\ \vdots \\ x'_\ell \end{pmatrix} \leftarrow \begin{pmatrix} x'_1 \\ \vdots \\ x'_\ell \end{pmatrix} - \kappa \begin{pmatrix} |M|_{m'_1} \\ \vdots \\ |M|_{m'_\ell} \end{pmatrix} \bmod \begin{pmatrix} m'_1 \\ \vdots \\ m'_\ell \end{pmatrix}$$

`/* en // dans` $\mathcal{B}'$ `*/`

nombre de multiplications est :

$$(n\ell + n + \ell)\,\texttt{MME}\,\texttt{l}_{\beta} + (n+1)\,\texttt{MME}\,\texttt{l}_{\log_2(n)}. \tag{1.18}$$

Lorsque le résidu x_{sk} est bien une information purement redondante des résidus $\boldsymbol{x}_{\mathcal{B}}$, c'est-à-dire si les résidus $(\boldsymbol{x}_{\mathcal{B}}, x_{sk})$ sont ceux d'un entier x dans $[\![0, M[\![$, alors cette conversion est complètement réduite. Dans le cas contraire, si l'entier x défini par $(\boldsymbol{x}_{\mathcal{B}}, x_{sk})$ appartient à $[\![M, m_{sk}M[\![$, le principe de cette conversion ne tient plus. Le résultat obtenu est alors de la forme

$$\text{Bex}_{sk}\left(\mathcal{B} \cup \{m_{sk}\}, (\boldsymbol{x}_{\mathcal{B}}, x_{sk})\right) = \text{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - kM$$

où $k = \left|\left(\text{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - x_{sk}\right) M^{-1}\right|_{m_{sk}}$ est un entier dans $[\![0, m_{sk}[\![$. Et $\text{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - kM$ ne peut alors être égal à $|x|_M$. Autrement dit, dans ce cas il existe un entier non nul $a \in [\![-\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right), m_{sk} - 1 - \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)]\!]\backslash\{0\}$ tel que

$$\text{Bex}_{sk}\left(\mathcal{B} \cup \{m_{sk}\}, (\boldsymbol{x}_{\mathcal{B}}, x_{sk})\right) = \varphi_{\mathcal{B}}^{-1}\left(\boldsymbol{x}_{\mathcal{B}}\right) + aM \neq \varphi_{\mathcal{B}}^{-1}\left(\boldsymbol{x}_{\mathcal{B}}\right).$$

Cette affirmation vient du fait que, sous l'hypothèse de coprimalité de M et m_{sk}, pour tout entier $x \in [\![0, m_{sk}M[\![$, alors $x < M$ si, et seulement si, $|x|_M \bmod m_{sk} = x_{sk}$. La preuve de la nécessité est évidente. Pour montrer la suffisance, supposons que $|x|_M \bmod m_{sk} = x_{sk}$ et $x \geqslant M$. Alors les résidus $(\boldsymbol{x}_{\mathcal{B}}, x_{sk})$ ont deux antécédents distincts par l'application $\varphi^{-1}_{\mathcal{B}\cup\{m_{sk}\}}$, à savoir x et $|x|_M$, ce qui est impossible puisque $\mathcal{B} \cup \{m_{sk}\}$ est une base RNS.

Méthode de Kawamura et al. Posch et Posch (2000) et Kawamura et al. (2000) prennent le parti de calculer une approximation du terme $\frac{\text{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})}{M}$ pour retrouver $\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)$. En explicitant le numérateur sous sa forme de somme de produits, la formulation suivante est obtenue :

$$\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) = \lfloor \sum_{i=1}^{n} \frac{\xi_{x,i,\mathcal{B}}}{m_i} \rfloor. \tag{1.19}$$

Si chaque modulus de la base $\mathcal{B}$ est de la forme $m_i = 2^r - c_i$, et si seuls les h bits de poids fort des $\xi_{x,i,\mathcal{B}}$ sont conservés (pour un certain $h \in [\![0, r[\![$) via l'utilisation de la fonction suivante :

$$\text{trunc}_h\left(\xi_{x,i,\mathcal{B}}\right) = \left(\xi_{x,i,\mathcal{B}} >> (r-h)\right) = \frac{\xi_{x,i,\mathcal{B}} - \left|\xi_{x,i,\mathcal{B}}\right|_{2^{r-h}}}{2^{r-h}},$$

alors l'approximation suivante est utilisée :

$$\tilde{\kappa}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) = \lfloor \sum_{i=1}^{n} \frac{\text{trunc}_h\left(\xi_{x,i,\mathcal{B}}\right)}{2^h} \rfloor = \lfloor \sum_{i=1}^{n} \text{eval}_h\left(\xi_{x,i,\mathcal{B}}\right) \rfloor. \tag{1.20}$$

Ce calcul ne consiste qu'en une somme d'entiers de h bits, et la valeur estimée de $\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)$ est la somme des retenues sortantes consécutives.

La fonction $\text{eval}_h\left(\xi_{x,i,\mathcal{B}}\right) = \frac{\text{trunc}_h\left(\xi_{x,i,\mathcal{B}}\right)}{2^h}$ sous-estime la valeur exacte de $\frac{\xi_{x,i,\mathcal{B}}}{m_i}$.

En effet, étant donné que $\xi_{x,i,\mathcal{B}} = 2^{r-h}\,\mathrm{trunc}_h\left(\xi_{x,i,\mathcal{B}}\right) + \left|\xi_{x,i,\mathcal{B}}\right|_{2^{r-h}}$ alors,

$$\begin{aligned} 0 \leqslant \frac{\xi_{x,i,\mathcal{B}}}{m_i} - \mathrm{eval}_h\left(\xi_{x,i,\mathcal{B}}\right) &= \frac{\xi_{x,i,\mathcal{B}}}{m_i} - \frac{\xi_{x,i,\mathcal{B}} - \left|\xi_{x,i,\mathcal{B}}\right|_{2^{r-h}}}{2^r} \\ &= \frac{c_i\xi_{x,i,\mathcal{B}} + m_i\left|\xi_{x,i,\mathcal{B}}\right|_{2^{r-h}}}{2^r m_i} \\ &\leqslant \frac{c_i\xi_{x,i,\mathcal{B}}}{2^r m_i} + \frac{2^{r-h}-1}{2^r} \\ &\leqslant \frac{c_i}{2^r} + \frac{2^{r-h}-1}{2^r} - \frac{c_i}{2^r m_i}. \end{aligned} \tag{1.21}$$

Δ_{kw} désigne un réel majorant l'erreur totale maximale pouvant être commise sur le calcul de la somme, c'est-à-dire par exemple n fois le majorant obtenu à l'Équation (1.21). Ceci permet alors d'écrire :

$$\lfloor \sum_{i=1}^{n} \frac{\xi_{x,i,\mathcal{B}}}{m_i} - \Delta_{kw} \rfloor \leqslant \tilde{\kappa}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) = \lfloor \sum_{i=1}^{n} \mathrm{eval}_h\left(\xi_{x,i,\mathcal{B}}\right) \rfloor \leqslant \lfloor \sum_{i=1}^{n} \frac{\xi_{x,i,\mathcal{B}}}{m_i} \rfloor. \tag{1.22}$$

Si l'approximation est assez précise pour que $\Delta_{kw} \in [0,1[$, il découle des Inégalités (1.22) que $\tilde{\kappa}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) \in \{\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - 1, \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)\}$.

En supposant que les moduli m_i vérifient $2^r - m_i \leqslant 2^c$, alors un majorant de l'erreur totale est donné par :

$$\Delta_{kw} \leqslant n\left(\frac{1}{2^{r-c}} + \frac{1}{2^h}\right) \tag{1.23}$$

ce qui permet d'extraire des conditions sur la taille r des moduli, le nombre d'éléments n de la base RNS $\mathcal{B}$ et sur le coefficient h de la fonction d'approximation eval_h pour pouvoir disposer d'une conversion avec réduction presque complète modulo M. Autrement dit, la conversion qui découle de cette approche, notée $\mathrm{Bex}_{kw,h}$ et explicitée par l'Algorithme 6, permet d'obtenir

$$\mathrm{Bex}_{kw,h}\left(\mathcal{B}, \boldsymbol{x}_{\mathcal{B}}\right) \in \{x, x + M\}.$$

Exemple 1.4 *Par exemple, Kawamura et al. exhibent un choix de paramètres pour utiliser une base RNS sur plus de* 1024 *bits. n et r doivent vérifier $nr > 1024$. Avec $n = 33$ moduli de $r = 32$ bits, alors en choisissant des moduli pseudo Mersenne vérifiant donc $2^{32} - m < 2^{16} = 2^c$, le majorant de l'Équation (1.23) donne $\Delta_{kw} \leqslant 33 \times \left(\frac{1}{2^{16}} + \frac{1}{2^h}\right)$. De ce fait, pour garantir $\Delta_{kw} < 1$, il suffit que h vérifie $h \geqslant 7 > \log_2\left(\frac{66\times 2^{16}}{2^{16}-66}\right)$.*

Kawamura et al. expliquent également qu'il est possible d'obtenir une conversion avec réduction complète lorsque l'entier x est assez petit (dans un sens précisé juste après). Dans ce cas, il est possible de garantir l'obtention de $\tilde{\kappa}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) = \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)$, moyennant une possible correction de l'erreur Δ_{kw}. En écrivant $\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) = \sum_{i=1}^{n} \frac{\xi_{x,i,\mathcal{B}}}{m_i} - \frac{x}{M}$ et si α_{kw} est un réel choisi dans l'intervalle $[\Delta_{kw}, 1[$ et destiné à corriger l'erreur des approximations, alors nous pouvons écrire :

$$\begin{cases} \displaystyle\sum_{i=1}^{n} \mathrm{eval}_h\left(\xi_{x,i,\mathcal{B}}\right) + \alpha_{kw} \geqslant \sum_{i=1}^{n} \frac{\xi_{x,i,\mathcal{B}}}{m_i} - \Delta_{kw} + \alpha_{kw} = \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) + \frac{x}{M} - \Delta_{kw} + \alpha_{kw}, \\ \displaystyle\sum_{i=1}^{n} \mathrm{eval}_h\left(\xi_{x,i,\mathcal{B}}\right) + \alpha_{kw} \leqslant \alpha_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) + \frac{x}{M} + \alpha_{kw} = \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) + \frac{\alpha_{kw} M + x}{M}. \end{cases} \tag{1.24}$$

Du système d'Inéquations (1.24), il vient que pour tout $x \in [\![0, (1-\alpha_{kw})\, M[\![$, $\tilde{\kappa}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) = \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)$. De cette constatation, il est possible de construire une conversion de base avec réduction complète pour au moins une partie de l'intervalle dynamique de $\mathcal{B}$, à savoir $[\![0, (1-\alpha_{kw})\, M[\![$.

Algorithme 6 : $\mathrm{Bex}_{kw,h}\left(\mathcal{B}, \mathcal{B}', \boldsymbol{x}_{\mathcal{B}}, \text{option}{=}\alpha_{kw}\right)$

Données : $\mathcal{B} = \{m_1, \ldots, m_n\}$ et $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$ deux bases RNS de moduli $< 2^r$, $h \leqslant r$, $\boldsymbol{x}_{\mathcal{B}}$ les résidus dans $\mathcal{B}$ d'un nombre $x \in [\![0, M[\![$. Précalculs possibles : $a_{i,j} = |M_i|_{m'_j}$ pour tout $(i,j) \in [\![1,n]\!] \times [\![1,\ell]\!]$, $|M|_m$ pour tout $m \in \mathcal{B}'$.

Résultat : $\boldsymbol{x}_{\mathcal{B}'}$, les résidus dans $\mathcal{B}'$ de x si $\alpha_{kw} \in [\Delta_{kw}, 1[$ et $x \in [\![0, (1-\alpha_{kw})\, M[\![$, ou de $x + \delta M$ avec $\delta \in \{0,1\}$ si $\alpha_{kw} = 0$.

1 **début**
2 **pour** $i \leftarrow 1$ **à** n **faire**
                /* en // dans $\mathcal{B}$ */
3 $\xi_{x,i,\mathcal{B}} \leftarrow \left| x_i \left|M_i\right|_{m_i}^{-1} \right|_{m_i}$
4 $\left(x'_1, \ldots, x'_\ell\right) \leftarrow (0, \ldots, 0)$
5 $\mathrm{reg}_{acc} \leftarrow (\alpha_{kw} << h)$
6 **pour** $i \leftarrow 1$ **à** n **faire**
7 $\kappa \leftarrow 0$
8 $\mathrm{reg}_{acc} \leftarrow \mathrm{reg}_{acc} + (\xi_{x,i,\mathcal{B}} >> (r-h))$
                /* $(\xi_{x,i,\mathcal{B}} >> (r-h)) = \mathrm{trunc}_h\left(\xi_{x,i,\mathcal{B}}\right)$ */
9 $\kappa \leftarrow \left(\mathrm{reg}_{acc} >> h\right)$
                /* $\mathrm{reg}_{acc} < 2^{h+1}$ */
10 **si** $\kappa = 1$ **alors**
11 $\mathrm{reg}_{acc} \leftarrow \left(\mathrm{reg}_{acc}\ \texttt{and}\ \left(2^h - 1\right)\right)$
                /* $\mathrm{reg}_{acc}\ \texttt{and}\ \left(2^h-1\right) = \mathrm{reg}_{acc} \bmod 2^h$ */
12 **pour** $j \leftarrow 1$ **à** ℓ **faire**
                /* en // dans $\mathcal{B}'$ */
13 $x'_j \leftarrow \left| x'_j + \xi_{x,i,\mathcal{B}} \times a_{i,j} - \kappa \left|M\right|_{m'_j} \right|_{m'_j}$

Remarque 1.7 *Le calcul de $\tilde{\kappa}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)$ mis en œuvre dans l'Algorithme 6 via l'utilisation de la variable d'accumulation reg_{acc} ne requiert que des additions d'entiers sur h bits, ainsi que des opérations très simples de décalage et de* `and` *bit à bit.*

L'Algorithme 6 se décline ainsi en deux versions, pour lesquelles il sera toujours supposé que l'entier h peut être choisi de manière à ce que $\Delta_{kw} \in [0, 1[$. La première, notée Bex_{kw}, n'utilise pas de terme de correction α_{kw}. Dans ce cas,

la conversion de l'entier x dans la base $\mathcal{B}'$ peut donner $|x|_{M'}$ ou $|x+M|_{M'}$. La seconde, notée Bex_{kwc}, suppose l'adjonction du terme correcteur α_{kw} et sera généralement utilisée sur une donnée x satisfaisant $x < (1-\alpha_{kw})\,M$. De ce fait, la conversion sera complètement réduite :

$$\forall x \in [\![0, (1-\alpha_{kw})\,M[\![,\ \text{Bex}_{kwc}\left(\mathcal{B}, \boldsymbol{x}_{\mathcal{B}}, \alpha_{kw}\right) = \varphi_{\mathcal{B}}^{-1}\left(\boldsymbol{x}_{\mathcal{B}}\right) = x.$$

Remarque 1.8 *Au vu de la première ligne du système d'Inéquations (1.24) avec $\alpha_{kw} = 0$, il est clair que $\text{Bex}_{kw}\left(\mathcal{B}, \mathcal{B}', x\right) = |x+M|_{M'}$ seulement si $x < \Delta_{kw}M$. Ainsi, comme souligné par Kawamura et al.,*

$$\forall x \in [\![0, M[\![,\ \text{Bex}_{kw}\left(\mathcal{B}, \boldsymbol{x}_{\mathcal{B}}\right) \in \{x, x+M\}\ \textit{et}\ \text{Bex}_{kw}\left(\mathcal{B}, \boldsymbol{x}_{\mathcal{B}}\right) \in [\![0, (1+\Delta_{kw})\,M[\![.$$

Remarque 1.9 *Lorsque la conversion corrigée Bex_{kwc} est utilisée sur un entier x quelconque de l'intervalle dynamique $[\![0, M[\![$, alors $\text{Bex}_{kwc}\left(\mathcal{B}, \boldsymbol{x}_{\mathcal{B}}, \alpha_{kw}\right) = x + \delta M$ avec $\delta \in \{-1, 0\}$. De plus, $\delta = -1$ seulement si $x \in [\![(1-\alpha_{kw})\,M, M[\![$.*

À opposer à l'inconvénient de la possible imprécision sur le résultat obtenu, cet algorithme a le double avantage d'être plus facilement parallélisable qu'une conversion Bex_{mrs} et de ne pas nécessiter l'adjonction d'un modulus redondant contrairement à Bex_{sk}. Ce second avantage est parfois essentiel puisqu'il n'est pas toujours possible d'obtenir le résidu redondant x_{sk} de x. C'est notamment ce qui se passe dans l'algorithme de réduction modulaire RNS détaillé dans la partie suivante 2.2.

Une architecture, dite Cox-Rower, a été proposée dans Kawamura et al. (2000), Nozaki et al. (2001). L'unité Cox est un additionneur-accumulateur réalisant le calcul de $\tilde{\kappa}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)$ en procédant de manière similaire au calcul mené dans l'Algorithme 6. Les Rowers quant à eux mettent en œuvre les calculs dans les canaux $\mathbb{Z}/m\mathbb{Z}$ des bases RNS. Des implantations matérielles sur FPGA fondées sur ce type d'architecture ont été réalisées, et démontrent la viabilité du RNS pour implanter efficacement des primitives cryptographiques (Guillermin 2010; 2011, Duquesne et Guillermin 2011, Yao et al. 2013).

La complexité de ces deux conversions reste similaire à celle d'une conversion de type Bex_{sk}, l'approche utilisée étant également basée sur l'Équation (1.5). Le coût total est le suivant :

$$\mathcal{C}\left(\text{Bex}_{kw,h}\left(\mathcal{B}, \mathcal{B}', ., .\right)\right) = n\ \texttt{MMEl}_{\mathcal{B}} + n\,\text{Add}_h + \ell n\ \texttt{MMEl}_{\mathcal{B}'} + \ell\,(2n-2)\ \texttt{AMEl}_{\mathcal{B}'}\ . \tag{1.25}$$

D'où, pour des tailles de moduli identiques β pour les deux bases, un coût en termes de multiplications modulaires élémentaires donné par :

$$\ell\,(n+1)\,\texttt{MMEl}_{\beta}. \tag{1.26}$$

Conversion accélérée Dans certaines situations, la conversion peut être allégée du calcul du coefficient $\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)$ afin d'accélérer les calculs (Bajard et al. 2001). Ceci a pour conséquence l'obtention d'un résultat compris dans $[\![0, nM[\![$. En effet, la conversion devient simplement le calcul des résidus de $\text{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)$ dans la base de destination. Le résultat obtenu est donc :

$$\text{Bex}_{crt}\left(\mathcal{B}, \boldsymbol{x}_{\mathcal{B}}\right) = x + \delta M,\ \delta \in [\![0, n-1]\!]. \tag{1.27}$$

Cette procédure, notée Bex_{crt}, est similaire à Bex_{sk} et son coût est :

$$\mathcal{C}\left(\mathrm{Bex}_{crt}\left(\mathcal{B},\mathcal{B}',.,.\right)\right) = n\ \texttt{MMEl}_{\mathcal{B}} + n\ell\ \texttt{MMEl}_{\mathcal{B}'} + \ell\,(n-1)\ \texttt{AMEl}_{\mathcal{B}'}. \tag{1.28}$$

Cette technique est par exemple mise en œuvre par Bajard et Imbert (2004) pour l'exponentiation modulaire RNS dans le cadre d'un RSA-RNS.

Remarque 1.10 *Les conversions de base décrites précédemment sont parallélisables sur les canaux de la base RNS de destination $\mathcal{B}'$. De ce fait, si l'ajout d'un canal dans $\mathcal{B}'$ modifie le coût global de la conversion, cela ne change pas le nombre d'étapes parallèles nécessaires. En revanche, ce n'est plus le cas si la base $\mathcal{B}$ est augmentée d'un canal supplémentaire. Un tel ajout accroît nécessairement le nombre d'étapes de calcul parallèles.*

Architecture Cox-Rower

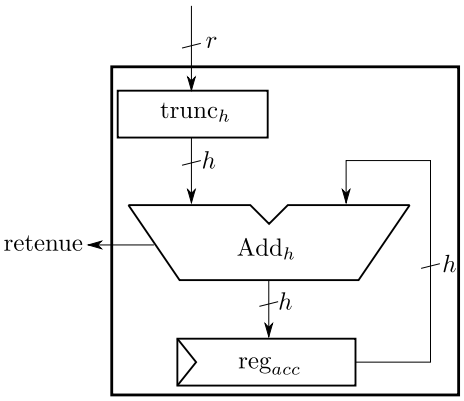


FIGURE 1.1 – *Unité Cox.*

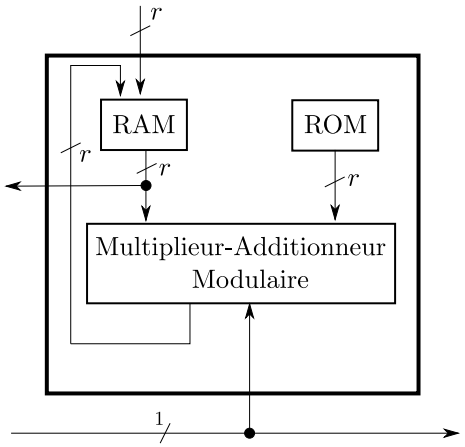


FIGURE 1.2 – *Unité Rower = canal RNS.*

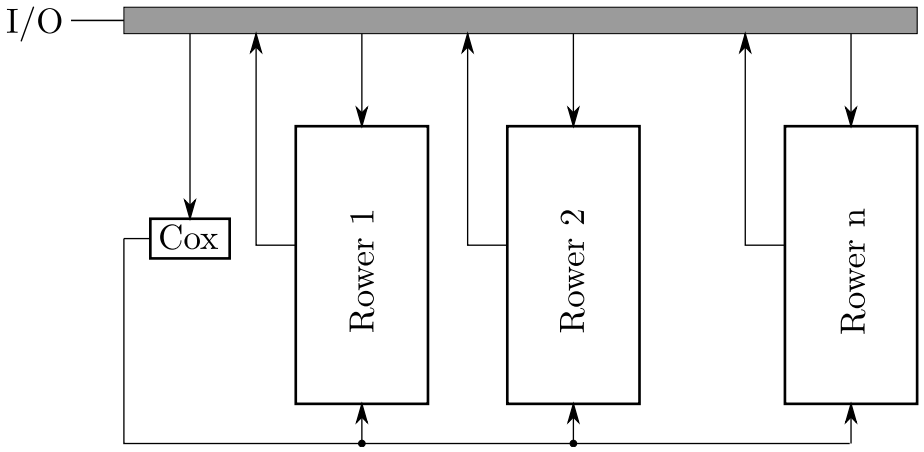


FIGURE 1.3 – *Principe d'une architecture dite Cox-Rower.*

Comme souligné précédemment, Kawamura et al. (2000) proposent une architecture dédiée au RNS (Fig. 1.3). Les Rowers (Fig. 1.2) permettent les calculs dans les canaux $\mathbb{Z}/m\mathbb{Z}$. Plus précisément, ils sont spécialisés pour effectuer efficacement des calculs du type $a \leftarrow |a + b \times c|_m$, opérations typiquement effectuées lors d'une conversion de base basée sur le TRC. Le Cox (Fig. 1.1) est l'unité réalisant le calcul de $\sum_{i=1}^{n} \mathrm{eval}_h\left(\xi_{x,i,\mathcal{B}}\right)$ (1.20), ce qui permet d'extraire la

valeur $\tilde{\kappa}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ et de l'envoyer à l'ensemble des Rowers. Il s'agit d'un simple additionneur-accumulateur sur h bits. La somme des retenues successives sortantes correspond à la valeur $\tilde{\kappa}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$. Lors d'une conversion de base, les entiers $\xi_{i,x,\mathcal{B}}$ sont transférés d'un Rower aux autres par un bus. Nozaki et al. (2001) modifient la précédente architecture en dotant chaque Rower de son propre Cox, et en privilégiant une interconnexion en anneau. Plus généralement, ce type d'architecture parallèle est au centre des recherches actuelles portant sur le RNS appliqué à la cryptographie asymétrique (Guillermin 2012, Bigou 2014), que ce soit pour du RSA (Ciet et al. 2003) ou encore de la cryptographie sur courbes elliptiques (Guillermin 2010) ou des couplages (Cheung et al. 2011).

1.3.2 Réduction modulaire en RNS

Disposer d'une réduction modulaire efficace est essentiel lorsqu'il s'agit de travailler dans des corps finis. L'état-de-l'art du calcul modulaire en RNS est basé sur une adaptation du principe de la réduction modulaire de Montgomery (1985).

Rappel sur le réduction modulaire de Montgomery

Algorithme 7 : Réduction modulaire de Montgomery, version digitale

Données : Une taille de mot β, quatre entiers x, t, p, u exprimés en base β tels que $\text{pgcd}(\beta, p) = 1$, $0 \leqslant p < \beta^n$, $0 \leqslant x < p\beta^t$, et un entier précalculable $u = |-p|_\beta^{-1} \in [\![0, \beta[\![$.

Résultat : $s \equiv \left|x\beta^{-1}\right|_p$, $s \in [\![0, 2p[\![$.

1 **début**
2 $s \leftarrow x$
3 **pour** $i \leftarrow 0$ **à** $t - 1$ **faire**
4 $q \leftarrow u \times s_0 \bmod \beta$
/* s_0 est le digit de poids faible de s en base β, i. e. $s \bmod \beta$ */
5 $(s \leftarrow s + p \times q)$
6 $s >>_\beta 1$
/* décalage d'un digit vers la droite : division par β */

Le principe de l'Algorithme 7 est de réduire l'entier x modulo p sans utiliser d'opération coûteuse de division, mais uniquement des additions, multiplications et décalages.

L'idée fondamentale est de rajouter à x une valeur z telle que $x + z$ reste congruente à x modulo p, tout en étant multiple de β. Selon la taille de z, en divisant $x + z$ par β, ce qui n'est qu'un simple décalage des digits, alors il est possible d'obtenir un résultat $s = \frac{x+z}{\beta}$ suffisamment proche de p. Pour que $x + z$ vérifie ces conditions, il est naturel de choisir $z = \left|-xp^{-1}\right|_\beta \times p$. Ainsi, $x + z < 2p\beta$, et donc $s < 2p$. Il est possible de mener la réduction à terme en exécutant une comparaison finale entre s et p. Néanmoins, lorsqu'il s'agit d'enchaîner les calculs dans $\mathbb{Z}/p\mathbb{Z}$, la réduction partielle dans $[\![0, 2p[\![$ garantit

la limitation de la croissance des résultats intermédiaires, ce qui est suffisant pour notre propos. Il est alors toujours possible d'effectuer cette comparaison uniquement sur le résultat final d'une exponentiation modulaire par exemple.

s n'est pas exactement le résultat souhaité puisque $s \equiv x\beta^{-1} \bmod p$. Lorsque cet algorithme de réduction est utilisé pour effectuer de nombreux calculs dans $\mathbb{Z}/p\mathbb{Z}$, une solution est de travailler sur une représentation différente des données en utilisant la représentation de Montgomery. Elle s'obtient en appliquant initialement à tout calcul l'automorphisme de $\mathbb{Z}/p\mathbb{Z}$ de multiplication par $|\beta|_p$. Si $(x, y) \in \mathbb{Z}/p\mathbb{Z}^2$, alors leurs représentants sont par définition $|x\beta|_p$ et $|y\beta|_p$. Notamment, il vient directement que cette représentation est stable par addition : $|x\beta|_p + |y\beta|_p = |(x+y)\beta|_p$. Un intérêt réel est que cette représentation est aussi stable par multiplication dans $\mathbb{Z}/p\mathbb{Z}$, dès lors que la réduction de Montgomery est utilisée. En effet, en réduisant $|x\beta|_p \times |y\beta|_p < p^2 < p\beta$, le résultat produit n'est autre que $\left|xy\beta^2 \times |\beta|_p^{-1}\right|_p = |xy\beta|_p$. La condition d'appliquer cet automorphisme préalablement aux calculs devient peu contraignante lorsqu'il s'agit d'effectuer des exponentiations modulaires, opération centrale dans bon nombre de cryptosystèmes.

Détaillons rapidement le coût de l'Algorithme 7. Il est clair que l'étape 4 est une multiplication entre deux entiers constitués d'un seul digit, soit une `EMul`. De même, l'étape suivante est constituée d'une multiplication entre l'entier p sur n digits, et q qui n'est qu'un mot simple, soit une $\mathrm{Mul}_\beta(n, 1) = n$ `EMul` et $n-1$ `EAdd`, ce qui donne alors un résultat sur $n+1$ digits. Suit alors une addition entre a, entier de $t+n$ mots, et un entier de $n+1$ digits, soit une majoration par $t+n$ `EAdd` en tenant compte des propagations de retenue. Ainsi, le coût d'une réduction modulaire d'un entier $x < p\beta^t$ par $p < \beta^n$ est majoré par :

$$\mathcal{C}_{\mathrm{RedMontDig}}(t, n) = t(n+1)\ \ \mathtt{EMul}_\beta + t(t+2n-1)\ \ \mathtt{EAdd}_\beta. \tag{1.29}$$

Il peut s'avérer avantageux, lors de calculs successifs dans un anneau $\mathbb{Z}/p\mathbb{Z}$, de ne pas effectuer une réduction modulo p systématiquement. S'il s'agit par exemple de calculer une somme $\sum_{i=1}^{k} a_i b_i \bmod p$, où $p < \beta^n$ et $a_i, b_i < p$ pour tout i, alors en appliquant une seule réduction finale sur la somme ce calcul nécessite au plus $(k+1)n^2 + \left(1 + \lceil \log_\beta(k) \rceil\right)(n+1)$ `EMul`. En comparaison, l'utilisation de k réductions successives amène à un coût total majoré par $kn^2 + kn(n+1) = 2kn^2 + kn$ `EMul`. Ce genre de schéma de calculs consistant en des sommes de produits privilégie donc une approche dite de « réduction fainéante ». Les sommes de produits apparaissant couramment dans les cryptosystèmes à base de courbes elliptiques et de couplages notamment, ce genre de schéma de réduction fainéante se révèle d'autant plus intéressant pour le cas du RNS puisque la complexité temporelle des produits $a_i b_i$ est alors linéaire en le nombre de β-digits des facteurs a_i et b_i (à cause de l'indépendance des multiplications menées dans les canaux RNS). Ce fait a été exploité pour optimiser les implantations RNS de telles fonctions cryptographiques (Bajard et al. 2006a, Duquesne et Guillermin 2011, Cheung et al. 2011, Bajard et al. 2013a).

Multiplication modulaire RNS

La plupart des systèmes cryptographiques modernes recourent très largement à l'utilisation de calculs dans de grands corps finis $\mathbb{F}_p$. Pour rendre le RNS compétitif dans le contexte de la cryptographie asymétrique, disposer d'un algorithme de multiplication modulaire efficace est crucial. Les algorithmes les plus efficaces (Bajard et al. 2001) reposent sur une adaptation de l'Algorithme 7 de réduction de Montgomery, où, pour réduire x modulo p, il s'agit foncièrement de calculer la division exacte $\frac{x+\left|-xp^{-1}\right|_\beta p}{\beta}$. Travaillant dans une base RNS $\mathcal{B}$, la première idée clef est de substituer la valeur β par M. De ce fait, le calcul de $q = \left|-xp^{-1}\right|_M$ se fait aisément sur les résidus de x et $u = \left|-p^{-1}\right|_M$ dans $\mathcal{B}$. Cependant, un problème se pose à l'étape suivante puisque le calcul de la somme $x + qp$ dans $\mathcal{B}$ cause la perte de toute l'information contenue dans les résidus de q. Et la division par M est bien sûr impossible à exécuter directement dans $\mathcal{B}$. Le second point clef est donc d'utiliser une base RNS auxiliaire $\mathcal{B}'$, première avec $\mathcal{B}$, au sein de laquelle il est possible de mener les calculs à terme. Les procédures de conversion de base sont utilisées pour calculer les résidus de q dans cette base annexe. Le schéma général de la procédure qui en découle est détaillé dans l'Algorithme 8, et est illustré par la Figure 1.4.

Algorithme 8 : RedModRNS $(\mathcal{B}, \mathcal{B}', x, p)$:

Données : Deux bases copremières $\mathcal{B}$ et $\mathcal{B}'$, un entier x représenté par ses résidus dans les deux bases, un modulus p représenté par ses résidus précalculés dans la base auxiliaire $\mathcal{B}'$, et deux procédures de conversion de base notées Bex_1 et Bex_2 ; les résidus précalculés de l'entier $\left|-p^{-1}\right|_M$ dans $\mathcal{B}$, et de $\left|M^{-1}\right|_{M'}$ dans $\mathcal{B}'$; toutes ces données vérifient les hypothèses de la Table 1.1.

Résultat : Les résidus dans $\mathcal{B} \cup \mathcal{B}'$ de $s \equiv \left|xM^{-1}\right|_p$.

début
- $\boldsymbol{q}_\mathcal{B} \leftarrow \left|x \times \left|-p^{-1}\right|_M\right|_M$ `/* en // dans` $\mathcal{B}$ `*/`
- $\widehat{\boldsymbol{q}}_{\mathcal{B}'} \leftarrow \text{Bex}_1(\mathcal{B}, \mathcal{B}', \boldsymbol{q}_\mathcal{B})$ `/* 1ère conversion de` q `vers` $\mathcal{B}'$ `*/`
- $\boldsymbol{t}_{\mathcal{B}'} \leftarrow \left|x + \widehat{q} \times p\right|_{M'}$ `/* en // dans` $\mathcal{B}'$ `*/`
- $\boldsymbol{s}_{\mathcal{B}'} \leftarrow \left|tM^{-1}\right|_{M'}$ `/* en // dans` $\mathcal{B}'$ `*/`
- $\boldsymbol{s}_\mathcal{B} \leftarrow \text{Bex}_2(\mathcal{B}', \mathcal{B}, \boldsymbol{s}_{\mathcal{B}'})$ `/* 2nde conversion de` s `vers` $\mathcal{B}$ `*/`

Remarque 1.11 *La valeur $\widehat{q} = \text{Bex}_1(\mathcal{B}, \boldsymbol{q}_\mathcal{B})$ peut être non complètement réduite modulo M, et donc être de la forme $q + \delta M$ avec $\delta \in \mathbb{N}$ suivant la conversion utilisée. Si la conversion est complètement réduite, alors $\widehat{q} = q$. Dans tous les cas, lorsque les tailles de M et M' sont convenablement choisies, alors il est toujours possible d'avoir $s = \frac{x+\widehat{q}p}{M} = \frac{x+qp}{M} + \delta p < (2+\delta)\, p < M'$. De plus, le résultat s vérifie bien $s \equiv xM^{-1} \bmod p$. La seconde conversion $\text{Bex}_2(\mathcal{B}', \mathcal{B}, \boldsymbol{s}_{\mathcal{B}'})$ doit être complètement réduite afin d'obtenir les résidus du même entier s à la fois dans la base $\mathcal{B}$ et dans la base $\mathcal{B}'$.*

Remarque 1.12 *Les conditions sur les bases $\mathcal{B}$ et $\mathcal{B}'$ imposées par $\mathcal{H}_{kw}$ dans la Table 1.1 garantissent que $\boldsymbol{s}_{\mathcal{B}'}$ est converti de manière exacte par Bex_{kwc}. En effet, vu la Remarque 1.8, $\widehat{q}$*

Hypothèses	$\mathcal{H}_{mrs}$	$\mathcal{H}_{kw}$	$\mathcal{H}_{sk}$
Bex_1	Bex_{mrs}	Bex_{kw}	Bex_{crt}
Bex_2	Bex_{mrs}	Bex_{kwc}	Bex_{sk}
redondance	$\times$	$\times$	$m_{sk} \geqslant \ell + 1$
coprimalités	$M \wedge M'p = 1$	$M \wedge M'p = 1$	$M \wedge m_{sk}M'p = 1, m_{sk} \wedge M' = 1$
taille x	$< \sigma p^2, \sigma \geqslant 4$	$< \sigma p^2, \sigma \geqslant 4$	$< (n+1)^2 p^2$
taille $\mathcal{B}$	$M > \sigma p$	$M > \frac{\sigma p}{(1-\Delta_{kw})}$	$M > (n+1)^2 p$
taille $\mathcal{B}'$	$M' > 2p$	$M' > \frac{2p}{(1-\alpha_{kw})}$	$M' > (n+1) p$
taille s	$< 2p$	$< 2p$	$< (n+1) p$

TABLE 1.1 – *Hypothèses pour l'Algorithme 8.*

vérifie $\widehat{q} < (1 + \Delta_{kw}) M$*, et il vient alors :*

$$\begin{aligned} s = \frac{x + \widehat{q}p}{M} &< \frac{\sigma p^2 + (1 + \Delta_{kw}) Mp}{M} \\ &< \frac{(1 - \Delta_{kw}) Mp + (1 + \Delta_{kw}) Mp}{M} = 2p \\ &< (1 - \alpha_{kw}) M'. \end{aligned}$$

De la même manière que pour l'Algorithme 7, la réduction modulaire réalisée par l'Algorithme 8 ne met pas en œuvre le test final sur la valeur s pour sa réduction complète dans $[\![0, p[\![$. Un tel test ne s'avère pas forcément utile tant que les calculs n'ont pas été menés à leur terme. La taille de la base RNS principale intervenant dans la réduction modulaire peut être calibrée de manière à ce qu'aucun dépassement de capacité ne soit possible si l'entrée de l'algorithme est le résultat d'une multiplication entre deux entiers ayant été réduit par l'algorithme de Montgomery par exemple. Ceci explique la condition $\sigma \geqslant 4$ dans la Table 1.1. Ceci permet par la même occasion de dériver un algorithme d'exponentiation modulaire de l'Algorithme 8.

Une comparaison entre s et p nécessiterait de passer par un système de représentation positionnel comme le MRS. Si Bex_2 est une conversion de type Bex_{mrs}, la comparaison devient bien sûr facilement réalisable à cause du caractère positionnel du MRS. Dans le cas contraire, une approche du même type que celles suggérées par Walter (1999), Hachez et Quisquater (2000), Gueron (2002) est possible. À savoir, si $M > 4p$, alors en appliquant une réduction sur une sortie précédente s de l'Algorithme 8 et vérifiant donc $s < 2p$, nous avons $\frac{s+pq}{M} < p + \frac{1}{2}$. Par conséquent, il ne reste plus qu'à vérifier une éventuelle égalité avec p, ce qui se réalise directement sur les résidus vu l'injectivité du RNS.

Remarque 1.13 *Comme la valeur q calculée dans la base M est un produit modulaire modulo M, il est inutile d'adjoindre un modulus redondant m_{sk} à la base $\mathcal{B}$. En effet, le résidu q_{sk} de q ne peut a priori être obtenu facilement. En revanche, cela devient possible pour s dans la base $\mathcal{B}'$, sous l'hypothèse d'avoir accès aux résidus x_{sk} et p_{sk} parmi les entrées de l'algorithme. Comme s est le résultat de la division exacte de $t = x + \widehat{q}p$ par M, alors s_{sk} s'obtient facilement par $\left|tM^{-1}\right|_{m_{sk}}$. Étant donné que la taille de $\mathcal{B}'$ vérifie $s < M'$, s_{sk} est effectivement une information redondante des résidus $\mathbf{s}_{\mathcal{B}'}$. Par conséquent, il est possible d'utiliser* Bex_{sk} *pour la seconde conversion.*

Le coût d'une réduction modulaire RNS est dépendant de celui des conver-

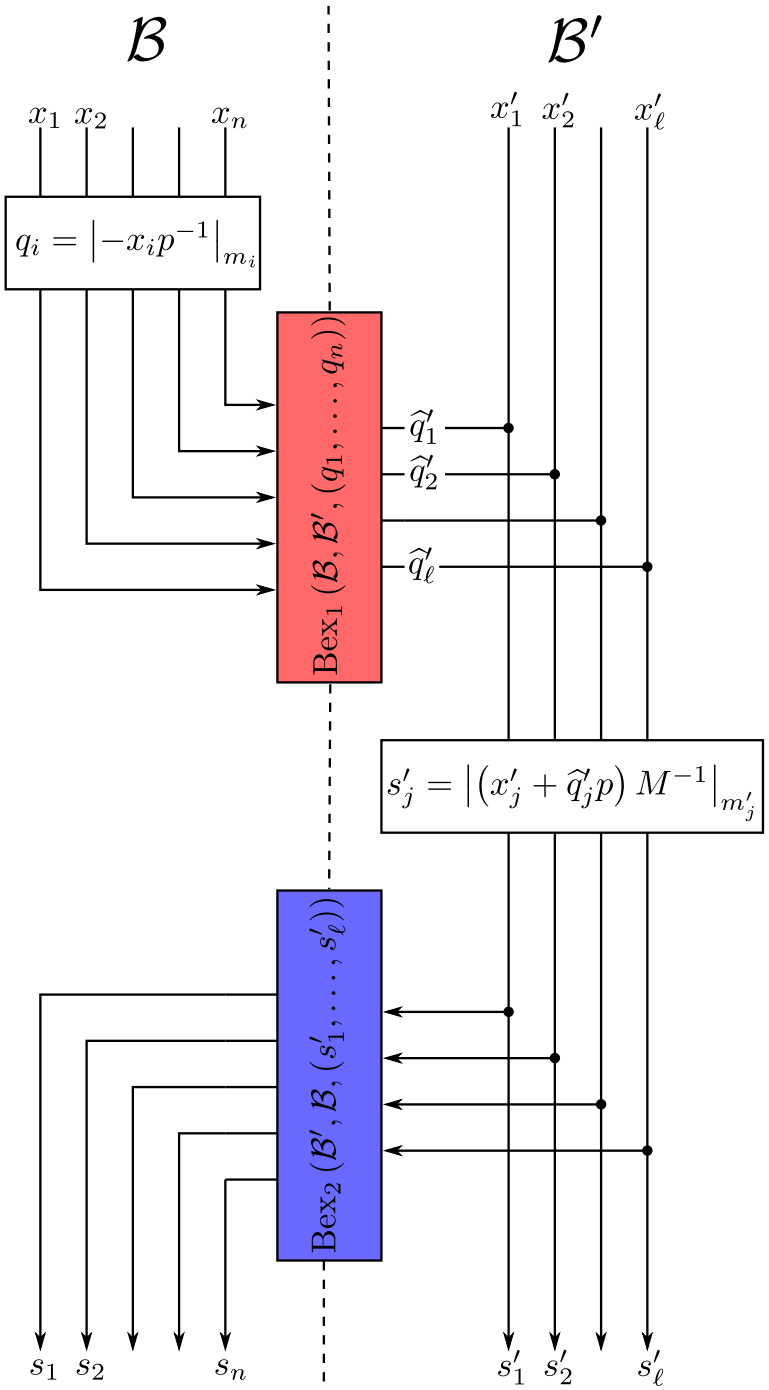


FIGURE 1.4 – *Illustration de l'Algorithme 8 de réduction modulaire RNS.*

sions de base utilisées, à savoir :

$$\mathcal{C}\left(\mathrm{Bex}_1\left(\mathcal{B}, \mathcal{B}', .\right)\right) + \mathcal{C}\left(\mathrm{Bex}_2\left(\mathcal{B}, \mathcal{B}', .\right)\right) + n \ \texttt{MMEl}_{\mathcal{B}} + 2\ell \ \texttt{MMEl}_{\mathcal{B}'} + \ell \ \texttt{AMEl}_{\mathcal{B}'}. \tag{1.30}$$

De nombreuses optimisations sont possibles dans le cas de l'exponentiation modulaire pour réduire le nombre total de multiplications nécessaires en en intégrant une partie dans les valeurs précalculées (Gandino et al. 2011; 2012). Par exemple, le calcul des coefficients $\xi_{q,i,\mathcal{B}}$ lorsque la première conversion de base Bex_1 est de type Bex_{crt} ou Bex_{kw} peut être court-circuité en utilisant les résidus précalculés $u_i = \left|-\left(pM_i\right)^{-1}\right|_{m_i}$. Si au final le coût reste tout de même

quadratique en le nombre de digits de p, l'avantage de la réduction modulaire RNS est sa flexibilité puisqu'elle s'applique à tout type de modulus p.

Suivant le choix des procédures de conversion de base, les conditions imposées sur les tailles des bases $\mathcal{B}$ et $\mathcal{B}'$ peuvent varier et ont en particulier pour but de garantir que la taille de l'entrée puisse être celle du produit de deux valeurs réduites, afin de permettre la réalisation efficace d'une multiplication modulaire. La Table 1.1 résume trois principaux ensembles d'hypothèses associées à l'Algorithme 8 correspondant à trois choix possibles pour le couple $(\mathrm{Bex}_1, \mathrm{Bex}_2)$ qui sont $(\mathrm{Bex}_{mrs}, \mathrm{Bex}_{mrs})$, $(\mathrm{Bex}_{crt}, \mathrm{Bex}_{sk})$ (Bajard et al. 2001), et $(\mathrm{Bex}_{\mathrm{kw}}, \mathrm{Bex}_{kwc})$ (Kawamura et al. 2000).

La flexibilité de la condition sur la taille de l'entrée x via le coefficient σ et sa répercussion sur la taille minimale de M dans $\mathcal{H}_{mrs}$, $\mathcal{H}_{sk}$ et $\mathcal{H}_{kw}$ est due à Guillermin (2010). Si les produits étaient systématiquement réduits modulo $2p$, $\sigma = 4$ suffirait. En effet, le produit de deux valeurs préalablement réduites est, de fait, plus petit que $4p^2$. Néanmoins, il se trouve, comme dans le cas d'une réduction de Montgomery classique, qu'il est plus intéressant pour certaines situations nécessitant le calcul de nombreuses sommes de produits de n'effectuer qu'une réduction finale.

La multiplication classique en RNS a en effet le précieux avantage d'être linéaire en le nombre de moduli des bases RNS, puisqu'elle ne nécessite que $n + \ell$ `MMEl`. Néanmoins, la multiplication modulaire reste, elle, quadratique à cause des opérations de conversion de base qui nécessitent $\mathcal{O}(n\ell)$ `MMEl`. Afin de tirer parti de l'efficacité du RNS dans l'exécution des opérations basiques d'addition et de multiplication, la « réduction fainéante » se révèle très intéressante, par exemple dans le cas d'algorithmes utilisés en cryptographie basée sur les courbes elliptiques comme souligné plus tôt. Un calcul du coût de la réduction modulaire d'une somme de produits, détaillé par la suite, permet de montrer l'influence de la linéarité de la multiplication en RNS.

Par soucis de simplicité, les bases $\mathcal{B}$ et $\mathcal{B}'$ sont supposées contenir un même nombre n de moduli de taille β. La taille n de ces bases est celle de $\log_\beta(p)$ (cf. Table 1.1). Ainsi, l'application d'une réduction fainéante en RNS, avec réduction modulaire de type Kawamura et al., pour le calcul d'une somme $\sum_{i=1}^{k} a_i b_i \bmod p$, où les termes a_i et b_i sont supposés réduits modulo p, nécessite donc $2kn$ `MMEl` pour les k produits de la somme, et $2n^2 + 5n$ `MMEl` pour la réduction modulaire, soit un coût total de

$$2n^2 + (2k+5)n \ \ \mathtt{MMEl}_\beta.$$

À titre de comparaison, la discussion précédente concernant le calcul du coût de l'Algorithme 7 montre qu'une approche standard nécessiterait dans ce cas

$$(k+1)n^2 + \left(1 + \lceil \log_\beta(k) \rceil\right)(n+1) \ \ \mathtt{EMul}_\beta.$$

Par la suite, la multiplication modulaire RNS naturellement associée à l'Algorithme 8 de réduction modulaire et appliquée aux résidus dans $\mathcal{B} \cup \mathcal{B}'$ de deux entiers x et y sera notée $\mathrm{MulModRNS}(\mathcal{B}, \mathcal{B}', x, y, p)$. La contrainte sur la taille des entiers x et y concerne alors simplement la taille du produit qui doit vérifier la borne donnée dans la Table 1.1.

Remarque 1.14 *La représentation de Montgomery d'un nombre x est obtenue par le calcul de la multiplication modulaire* $\mathrm{MulModRNS}\left(\mathcal{B}, \mathcal{B}', x, \left|M^2\right|_p, p\right)$.

Exemple 1.5 *Soit deux bases RNS $\mathcal{B} = \{3,5,7,11\}$, $\mathcal{B}' = \{2,13,17\}$. La multiplication modulaire dérivée de l'Algorithme 8 est utilisée pour calculer $\left|xyM^{-1}\right|_p = 156$ où $p = 211$, $x = 313$, $y = 215$. L'ensemble de ces données vérifie les hypothèses $\mathcal{H}_{mrs}$ de la Table 1.1 avec $\sigma = 4$. Le détail du calcul est donné par la Table 1.2. Nous y effectuons une comparaison finale pour obtenir une réduction complète modulo p.*

	$\mathcal{B}$				$\mathcal{B}'$		
	3	5	7	11	2	13	17
p			$\times$		1	3	7
x	1	3	5	5	1	1	7
y	2	0	5	6	1	7	11
xy	2	0	3	8	1	7	9
$\left\|-p^{-1}\right\|_M$	2	4	6	5		$\times$	
$q = \left\|-xyp^{-1}\right\|_M$	1	0	3	7		$\times$	
Coeff. MRS de q	1	3	0	5		$\times$	
$q' = \mathrm{Bex}_{mrs}(\mathcal{B},\mathcal{B}',\boldsymbol{q}_{\mathcal{B}})$			$\hookrightarrow$		1	2	8
pq'			$\times$		1	6	5
$xy + pq'$			$\times$		0	0	14
$s' = \frac{xy+pq'}{M}$			$\times$		0	0	3
Coeff. MRS de s'			$\times$		0	0	6
Coeff. MRS de p			$\times$		1	1	8
soustraction par p ?			$\times$			non	
$s = \mathrm{Bex}_{mrs}\left(\mathcal{B}',\mathcal{B},\boldsymbol{s}'_{\mathcal{B}'}\right)$	0	1	2	2		$\hookleftarrow$	
$\left\|xyM^{-1}\right\|_p$	0	1	2	2	0	0	3

TABLE 1.2 – *Détail des calculs de l'Exemple 1.5.*

1.4 RNS et arithmétique dans $\mathbb{F}_{p^s}$

Outre les corps finis premiers, la cryptographie asymétrique recourt à l'utilisation de corps finis de cardinal p^s avec p un nombre premier.

Une extension de corps finie de degré s de $\mathbb{F}_p$ peut être construite comme le corps de rupture sur $\mathbb{F}_p$ d'un polynôme irréductible $N(X)$ de degré s. Les corps de cardinal p^s étant isomorphes, le choix d'une représentation via le polynôme $N(X)$ détermine en pratique l'efficacité de l'arithmétique. Les éléments du corps $\mathbb{F}_{p^s}$ identifié au quotient $\mathbb{F}_p[X]/N(X)\,\mathbb{F}_p[X]$ sont représentables par des polynômes sur $\mathbb{F}_p$ de degré strictement inférieur à s. L'arithmétique dans $\mathbb{F}_{p^s}$ se réduit alors à une arithmétique modulaire sur des polynômes.

Concernant la cryptographie sur courbes elliptiques, le National Institute of Standards and Technology (NIST) suggère par exemple l'utilisation de corps binaires $\mathbb{F}_{2^s}$ où s varie de 163 à 571 iNI (1999). La cryptographie basées sur les couplages est aussi un domaine utilisateur de corps finis non premiers où, selon les degrés de sécurité recherchés, la taille de $\log_2(p^s)$ doit atteindre des tailles assez grandes, 960 à 18000, afin de rendre le problème du logarithme discret impraticable, avec un degré d'extension s pouvant varier de 2 à 36 (Freeman et al. 2006).

1.4.1 Représentation des éléments de $\mathbb{F}_{p^s}$

Représentation RNS

Lorsque la représentation polynomiale des éléments de $\mathbb{F}_{p^s}$ est privilégiée, il est possible de choisir n polynômes $m_1(X), \ldots, m_n(X)$ premiers entre eux tels que la somme de leur degré $d = \sum_{i=1}^{n} \deg(m_i)$ vérifie $d \geqslant s$. Dans ce cas, l'ensemble de ces polynômes définit une base de l'espace vectoriel sur $\mathbb{F}_p$ des polynômes de $\mathbb{F}_p[X]$ de degré au plus $d-1$, espace noté $\mathbb{F}_p[X]_d$. Le théorème des restes chinois établit l'existence de l'isomorphisme suivant :

$$\begin{array}{rccl} \varphi: & \mathbb{F}_p[X]_d & \rightarrow & \prod_{i=1}^{s} \mathbb{F}_p[X]/m_i(X)\mathbb{F}_p[X] \\ & A(X) & \mapsto & \left(|A(X)|_{m_1(X)}, \ldots, |A(X)|_{m_n(X)} \right). \end{array}$$

La preuve de cette version du théorème des restes chinois est du même acabit que celle du Théorème 1.1. Les polynômes $m_i(X)$ sont généralement choisis de manière à ce que l'arithmétique polynomiale modulo $m_i(X)$ soit efficace. Il est donc judicieux de les choisir le plus creux possible, comme c'est le cas des trinômes pour les corps binaires (Bajard et al. 2005), ou des polynômes de la forme $X^{d_i} + c_i$ (Jullien etS al. 2005).

Chaque polynôme $A(X)$ de degré au plus d, et en particulier chaque élément de $\mathbb{F}_{p^s}$, est représenté de manière univoque par ses résidus $(A_1(X, \ldots, A_n(X))$ où $A_i(X) = A(X) \bmod m_i(X)$ pour $i \in [\![1, n]\!]$. Les opérations arithmétiques s'effectuent toujours en parallèle sur les résidus. Néanmoins, les calculs polynomiaux se font modulo $\prod_{i=1}^{n} m_i(X)$. Il est donc nécessaire de disposer d'une opération de réduction modulaire afin de se ramener dans le quotient $\mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$. Pour ce faire, il faut définir des procédures de conversion de la base $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ vers une base copremière $\mathcal{B}' = \{m_1(X)', \ldots, m_\ell(X)'\}$, en se basant sur la technique classique d'interpolation polynomiale de Newton. Cette approche est le pendant de la conversion MRS du cas des RNS sur les entiers. Cette conversion se construit sur la donnée de la base de l'espace vectoriel des polynômes de degré au plus $d-1$ sur $\mathbb{F}_p$ définie par

$$\{1, m_1(X), m_1(X)m_2(X), \ldots, \prod_{i=1}^{n-1} m_i(X)\}.$$

Le calcul des coefficients d'un élément $A(X)$ est semblable à celui du calcul des coefficients MRS donnée par les Équations (1.9), et se déduit donc des coefficients RNS $(A_1(X), \ldots, A_n(X))$ de la manière suivante :

$$\begin{cases} \tilde{A}_1(X) = A_1(X) \\ \tilde{A}_2(X) = \left| \left(A_2(X) - \tilde{A}_1(X)\right) m_1(X)^{-1} \right|_{m_2(X)} \\ \qquad \vdots \\ \tilde{A}_n(X) = \left| \left(\ldots \left(A_n(X) - \tilde{A}_1(X)\right) m_1(X)^{-1} - \ldots - \tilde{A}_{n-1}(X) \right) m_{s-1}(X)^{-1} \right|_{m_s(X)}. \end{cases} \tag{1.31}$$

La reconstruction de $A(X)$ s'effectue par exemple par un schéma de Hörner :

$$\begin{aligned} A(X) &= \tilde{A}_1(X) + m_1(X)\left(\tilde{A}_2(X) + m_2(X)\left(\ldots + m_{n-1}(X)\tilde{A}_n(X)\right)\ldots\right) \\ &= \tilde{A}_1(X) + \tilde{A}_2 m_1(X) + \ldots + \tilde{A}_n(X) m_1(X)\ldots m_{n-1}(X). \end{aligned} \tag{1.32}$$

Le membre de droite de l'Équation (1.32) a effectivement son degré dominé par $d_n - 1 + \sum_{i=1}^{n-1} d_i = d - 1$. Une procédure de conversion se déduit immédiatement de cette approche, de manière analogue à l'Algorithme 4 décrivant la procédure Bex_{mrs} dans le cas des entiers.

Une autre approche utilise l'analogue pour le cas polynomial de l'approche constructive de la preuve du théorème des restes chinois pour les entiers donnée par l'Égalité (1.15). La formule est alors la suivante :

$$\forall A(X) \in \mathbb{F}_p[X]_d,\ A(X) = \sum_{i=1}^{n} \left| A_i(X) \prod_{j=1, j\neq i}^{n} m_j(X)^{-1} \right|_{m_i(X)} \times \prod_{j=1, j\neq i}^{n} m_j(X). \tag{1.33}$$

Remarque 1.15 *La réduction modulo $\prod_{j=i}^{n} m_i(X)$ du membre de droite de l'Équation (1.33) n'est pas requise. En effet, le degré du i-ième terme de cette somme est majoré par $d_i - 1 + \sum_{j=1, j\neq i}^{n} d_j = d - 1$. Ainsi, le polynôme reconstruit possède effectivement un degré strictement inférieur à d.*

Algorithme 9 : $\text{Bex}_{New}(\mathcal{B}, \mathcal{B}', \boldsymbol{A}_{\mathcal{B}} = (A_1(X), \ldots, A_n(X)))$

Données : $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ et $\mathcal{B}' = \{m'_1(X), \ldots, m'_\ell(X)\}$ deux bases RNS de polynômes copremières, $\boldsymbol{A}_{\mathcal{B}}$ les résidus dans $\mathcal{B}$ d'un polynôme $A(X)$ de degré $\leqslant d-1$.

Résultat : $\boldsymbol{A}_{\mathcal{B}'} = \left(A'_1(X), \ldots, A'_\ell(X)\right)$ les résidus dans $\mathcal{B}'$ de $A(X)$.

```
début
    (Ã_1(X), ..., Ã_n(X)) ← (A_1(X), ..., A_n(X))
    pour i ← 2 à n faire
                                        /* coeff. MRS, Éq. (1.31) */
        pour j ← i à n faire
                                        /* en parallèle dans B */
            Ã_j(X) ← (Ã_j(X) − Ã_{i−1}(X)) m_{i−1}^{−1}(X) mod m_j(X)
    (A'_1(X), ..., A'_ℓ(X)) ← (Ã_n(X), ..., Ã_n(X))
    pour i ← 1 à ℓ faire
                                        /* en parallèle dans B' */
        pour j ← n − 1 à 1 faire
            A'_i(X) ← (A'_i(X) × m_j(X) + Ã_j(X)) mod m'_i(X)
    retourner (A'_1(X), ..., A'_ℓ(X))
```

Représentation de Lagrange

Les éléments du corps fini vus comme des polynômes peuvent être représentés autrement que par leurs coefficients dans $\mathbb{F}_p$. Lorsque la caractéristique p et le degré s de l'extension vérifient $s < p$, chaque polynôme peut être représenté par ses valeurs prises en s points distincts $(e_1, \ldots, e_s)$ (Bajard et al. 2006b). Cette représentation dite de Lagrange a l'avantage de réduire l'arithmétique polynomiale modulaire modulo $N(X)$ à l'arithmétique sur le corps de base $\mathbb{F}_p$. Formellement, cette représentation s'appuie sur la représentation précédente appliquée avec la base de moduli $(X - e_1, \ldots, X - e_s)$. Le théorème des restes chinois établit dans ce cas l'existence de l'isomorphisme suivant :

$$\begin{array}{rccl} \varphi: & \mathbb{F}_p[X]_s & \rightarrow & \prod_{i=1}^{s} \mathbb{F}_p[X]/(X - e_i)\,\mathbb{F}_p[X] \\ & A(X) & \mapsto & \left(|A(X)|_{X-e_1}, \ldots, |A(X)|_{X-e_s}\right) = (A(e_1), \ldots, A(e_s)). \end{array}$$

Du théorème des restes chinois découle ainsi le fait que l'évaluation en s points distincts de $\mathbb{F}_p$ de tout polynôme de $\mathbb{F}_p[X]$ de degré inférieur ou égal à $s-1$ détermine de manière univoque le polynôme considéré parmi l'ensemble des polynômes de degré $\leqslant s-1$.

Une autre conséquence du théorème est que pour tout $(A(X), B(X)) \in \mathbb{F}_p[X]/N(X)\,\mathbb{F}_p[X]$, et toute opération $\star \in \{+, -, \times\}$, $A \star B$ s'effectue directement sur les résidus dans $\mathbb{F}_p$:

$$\varphi(A(X) \star B(X)) = (A(e_1) \star B(e_1), \ldots, A(e_s) \star B(e_s)).$$

L'isomorphisme φ est l'application de conversion de la représentation polynomiale à la représentation lagrangienne. L'application inverse peut être réalisée de diverses manières. Une première méthode est l'interpolation polynomiale de Lagrange. Elle est la simple adaptation de l'Équation (1.33) au cas présent. Pour tout s-uplet de points (e_i, a_i), où $a_i = A(e_i)$, le polynôme A est reconstitué par la formule suivante :

$$A(X) = \sum_{j=1}^{s} a_j \prod_{i=0, i\neq j}^{s} \frac{X - e_i}{e_j - e_i}. \tag{1.34}$$

L'Équation (1.34) permet de définir une procédure de conversion de base entre deux bases copremières $\mathcal{B} = \{X - e_1, \ldots, X - e_s\}$ et $\mathcal{B}' = \{X - e'_1, \ldots, X - e'_s\}$ sous l'hypothèse que $p \geqslant 2s$.

Il est possible d'optimiser l'efficacité de cette conversion en choisissant au mieux des points e_i et e'_j permettant de maîtriser la taille des coefficients de la matrice $\mathbf{\Omega}$ dans l'Algorithme 10 (Bajard et al. 2006b). Le principe de cette conversion de type produit matrice-vecteur est analogue aux conversion de base basées sur le TRC (cf. notamment Bex_{crt}, Partie 1.3.1) pour les RNS sur les entiers.

Cette technique de conversion ne s'applique que pour les corps $\mathbb{F}_{p^s}$ dont le corps premier contient assez de points distincts, en l'occurrence des corps vérifiant $p \geqslant 2s$.

Remarque 1.16 *Si la conversion* Bex_{Lag} *définie par l'Algorithme 10 est effectivement construite sur une approche similaire à celle des conversions basées sur le TRC pour les RNS sur les*

Algorithme 10 : $\mathrm{Bex}_{Lag}\left(\mathcal{B}, \mathcal{B}', A_{\mathcal{B}} = (a_1, \ldots, a_s)\right)$

Données : $\left(e_1, \ldots, e_s, e'_1, \ldots, e'_s\right) \in \mathbb{F}_p$ $2s$ points distincts, $\mathcal{B} = \{X - e_1, \ldots, X - e_s\}$ et $\mathcal{B} = \{X - e'_1, \ldots, X - e'_s\}$ deux bases RNS de polynômes copremières, $\boldsymbol{A}_{\mathcal{B}}$ les résidus dans $\mathcal{B}$ d'un polynôme $A(X)$ de degré $\leqslant s-1$, la matrice précalculée $\boldsymbol{\Omega} = \left(\prod_{k=0, k\neq j}^{s} \frac{e'_i - e_k}{e_j - e_k}\right)_{(i,j)\in[\![1,s]\!]^2}$.

Résultat : $A_{\mathcal{B}'} = \left(a'_1, \ldots, a'_s\right)$ les résidus dans $\mathcal{B}'$ de $A(X)$.

1 **début**

2 $\quad \begin{pmatrix} a'_1 \\ \vdots \\ a'_s \end{pmatrix} \leftarrow \boldsymbol{\Omega} \times \begin{pmatrix} a_1 \\ \vdots \\ a_s \end{pmatrix}$

3 $\quad$ **retourner** $\left(a'_1, \ldots, a'_s\right) = \left(A(e'_1), \ldots, A(e'_s)\right)$

entiers, elle possède à présent l'avantage de toujours être complètement réduite. Les problèmes liés au calcul du coefficient $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ de l'Équation (1.15) dans le cas des conversions Bex_{sk} *et* Bex_{kw} *n'ont pas de semblable dans le contexte de la conversion* Bex_{Lag}*. En effet, cette dernière opération retourne nécessairement un polynôme correctement réduit modulo $\prod_{i=1}^{s}(X - e_i)$, puisque la somme (1.34) (analogue à la somme* $\mathrm{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ *dans le cas entier) est bien de degré $< s$.*

Toujours sous l'hypothèse $p \geqslant 2s$, une seconde méthode de conversion se construit sur la technique d'interpolation polynomiale de Newton. La base MRS associée à la base RNS $\mathcal{B}$ est définie par l'ensemble de polynômes $\{1, X - e_1, \ldots, \prod_{i=1}^{s-1}(X - e_i)\}$. C'est une base de l'espace vectoriel des polynômes de $\mathbb{F}_p$ de degré au plus $s-1$, tout comme l'est l'ensemble des s polynômes de Lagrange construits précédemment $\left(\prod_{i=0, i\neq j}^{s} \frac{X - e_i}{e_j - e_i}\right)_{j\in[\![1,s]\!]}$. Les coefficients MRS du polynôme $A(X)$ sont dérivés des résidus $(a_1, \ldots, a_s)$ de la même manière que les Équations (1.9) explicitant le calcul des coefficients MRS :

$$\begin{cases} \tilde{a}_1 = a_1 \\ \tilde{a}_2 = \left|(a_2 - \tilde{a}_1)(X - e_1)^{-1}\right|_{X - e_2} = \left|(a_2 - \tilde{a}_1)(e_2 - e_1)^{-1}\right|_p \\ \quad \vdots \\ \tilde{a}_s = \left|\left(\ldots (a_n - \tilde{a}_1)(X - e_1)^{-1} - \ldots - \tilde{a}_{s-1}\right)(X - e_{s-1})^{-1}\right|_{X - e_s} \\ \quad = \left|\left(\ldots (a_n - \tilde{a}_1)(e_s - e_1)^{-1} - \ldots - \tilde{a}_{s-1}\right)(e_s - e_{s-1})^{-1}\right|_p . \end{cases} \tag{1.35}$$

La reconstruction de $A(X)$ s'effectue par exemple par un schéma de Hörner :

$$\begin{aligned} A(X) &= \tilde{a}_1 + (X - e_1)\left(\tilde{a}_2 + (X - e_2)\left(\ldots + \tilde{a}_s (X - e_{s-1})\right)\ldots\right) \\ &= \tilde{a}_1 + \tilde{a}_2 (X - e_1) + \ldots + \tilde{a}_s (X - e_1)\ldots(X - e_{s-1}) . \end{aligned} \tag{1.36}$$

Une procédure de conversion de base se déduit immédiatement de cette interpolation, et est le pendant de l'Algorithme 4 décrivant la procédure Bex_{mrs}.

1.4.2 Réduction modulaire de Montgomery dans $\mathbb{F}_{p^s}$

La multiplication modulaire de Montgomery a été adaptée aux corps binaires $\mathbb{F}_{2^s}$ par Koç et Acar (1998). Le principe reste généralisable à toute extension $\mathbb{F}_{p^s}$ (Bajard et al. 2006b). Si $R(X)$ est un polynôme premier à $N(X)$ et constitue le facteur de Montgomery, alors les représentations de Montgomery des polynômes $A(X)$ et $B(X)$ sont respectivement $A_R(X) = |A(X)R(X)|_{N(X)}$ et $B_R(X) = |B(X)R(X)|_{N(X)}$. La multiplication dans le quotient $\mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$ avec réduction de Montgomery consiste donc à calculer les quantités suivante :

$$Q(X) = \left|-A_R(X)B_R(X)N(X)^{-1}\right|_{R(X)},\ S(X) = \frac{A_R(X)B_R(X) + Q(X)N(X)}{R(X)}. \tag{1.37}$$

Par définition de Q, S est le résultat d'une division exacte. De plus, le résultat S vérifie bien $S(X) \equiv A(X)B(X)R(X) \bmod N(X)$. Le facteur R est encore une fois choisi de manière à ce que la réduction modulo R et la division par R soient peu coûteuses. Il est alors naturel de choisir $R(X) = X^s$. Dans ce cas, la réduction modulaire est une suppression des monômes de degré $\geqslant s$, et la division consiste à diminuer de s le degré des monômes de la quantité à diviser.

Algorithme 11 : Multiplication modulaire dans $\mathbb{F}_{p^s} \simeq \mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$

Données : $A(X)$ et $B(X)$ deux éléments de $\mathbb{F}_p[X]$ de degré $\leqslant s-1$.
Résultat : $A(X)B(X)X^{-s} \bmod N(X)$.

1 **début**
2 $\quad Q(X) \leftarrow -A_R(X)B_R(X)N(X)^{-1} \bmod X^s$
3 $\quad T(X) \leftarrow A(X)B(X) + Q(X)N(X)$
4 $\quad S(X) \leftarrow T(X)X^{-s}$
5 $\quad$ **retourner** $S(X)$

Remarque 1.17 *Contrairement au cas des entiers où une comparaison finale peut être nécessaire, la réduction modulaire de Montgomery définie par l'Algorithme 11 est complètement réduite. En effet, le degré du polynôme $T(X)$ vérifie :*

$$\deg(T) \leqslant \max(\deg(A) + \deg(B), \deg(Q) + \deg(N)).$$

Or, A et B ont par hypothèse un degré au plus $s-1$, N est de degré s, et par construction Q est de degré maximal $s-1$. Ainsi $\deg(T) \leqslant 2s-1$. *Et par suite,* $\deg(S) \leqslant s-1$. *Par conséquent, $S(X) = S(X) \bmod N(X)$.*

Lorsque les éléments A et B sont représentés en RNS dans deux bases copremières $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ et $\mathcal{B}' = \{m'_1(X), \ldots, m'_\ell(X)\}$, la multiplication modulaire RNS reprend le principe de l'Algorithme 8.

Par définition de Q, il vient $T(X) = A(X)B(X) + Q(X)N(X) \equiv 0 \bmod M(X)$. Ainsi, le polynôme S est défini par les résidus de la division exacte $\frac{T(X)}{M(X)} \equiv A(X)B(X)M(X)^{-1} \bmod N(X)$. De plus,

$$\deg(S) \leqslant \max(2s + 2\sigma - 2 - d, k-1) \leqslant k-1.$$

Algorithme 12 : Multiplication modulaire RNS dans $\mathbb{F}_{p^s} \simeq \mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$

Données : σ un paramètre entier, $\mathcal{B}$ et $\mathcal{B}'$ deux bases copremières avec $d = \deg(M) \geqslant s + 2\sigma - 1$ et $d' = \deg(M') \geqslant s$, $A(X)$ et $B(X)$ deux éléments de $\mathbb{F}_p[X]$ de degré $d_A, d_B < s + \sigma$ représentés par leurs résidus dans $\mathcal{B} \cup \mathcal{B}'$.

Résultat : les résidus dans $\mathcal{B} \cup \mathcal{B}'$ de $S(X) = A(X)B(X)M(X)^{-1} \bmod N(X)$.

début
- **pour** $i \leftarrow 1$ **à** n **faire**
  - `/* en parallèle dans` $\mathcal{B}$ `*/`
  - $Q_i(X) \leftarrow -A_i(X)B_i(X)N(X)^{-1} \bmod m_i(X)$
- $\left(Q'_1(X), \ldots, Q'_\ell(X)\right) \leftarrow \mathrm{Bex}_1\left(\mathcal{B}, \mathcal{B}', (Q_1(X), \ldots, Q_n(X))\right)$ `/* première interpolation */`
- **pour** $i \leftarrow 1$ **à** ℓ **faire**
  - `/* en parallèle dans` $\mathcal{B}'$ `*/`
  - $T'_i(X) \leftarrow \left(A'_i(X)B'_i(X) + Q'_i(X)N(X)\right) \bmod m'_i(X)$
  - $S'_i(X) \leftarrow T'_i(X)M(X)^{-1} \bmod m'_i(X)$
- $(S_1(X), \ldots, S_n(X)) \leftarrow \mathrm{Bex}_2\left(\mathcal{B}', \mathcal{B}, \left(S'_1(X), \ldots, S'_\ell(X)\right)\right)$ `/* seconde interpolation */`
- **retourner** $(S_1(X), \ldots, S_n(X)), \left(S'_1(X), \ldots, S'_\ell(X)\right)$

Comme la base $\mathcal{B}'$ vérifie $d' \geqslant k$, les résidus $\left(S'_1(X), \ldots, S'_\ell(X)\right)$ définissent complètement S. Finalement, $S(X) = A(X)B(X)M(X)^{-1} \bmod N(X)$.

L'Algorithme 12 s'applique évidemment au cas particulier de la représentation de Lagrange.

Conclusion

Ce chapitre a permis d'introduire la définition des systèmes de représentation par les restes, aussi dénommés RNS, comme conséquence du théorème des restes chinois. Nous avons expliqué comment l'utilisation de tels systèmes permet d'améliorer la complexité des opérations arithmétiques, d'une part par le parallélisme qui leur est naturellement associé, et d'autre part à cause de la flexibilité apportée par la liberté de choix des moduli qui peuvent, de par leurs propriétés, faciliter les opérations élémentaires `MME1` et `AME1`. Néanmoins, la réduction modulaire est une opération délicate en RNS. Les techniques de l'état-de-l'art reposent sur une adaptation de la réduction de Montgomery en faisant intervenir des procédures de conversion de base. L'efficacité de la réduction modulaire RNS qui en découle permet alors de contribuer à faire du RNS une arithmétique compétitive en termes de complexité pour le contexte de la cryptographie asymétrique, que ce soit pour des corps finis premiers ou non.

Néanmoins, les propriétés du RNS qui sont avantageusement exploitées pour la cryptographie asymétrique ne se limitent pas à une question de complexité pure. Par exemple, le fait de disposer d'une réduction modulaire « à la Montgomery » fait apparaître le fait de devoir utiliser les représentations de Montgomery. Celles-ci étant liées au produit M des moduli de la base RNS principale $\mathcal{B}$ utilisée pour la réduction, cette représentation offre une sorte de masquage des données, qui peut être rendu « aléatoire ». Pour ce faire, Bajard et al. (2004) montrent une manière astucieuse de changer la base $\mathcal{B}$, et conséquemment les représentations de Montgomery, à la volée lors d'une série de multiplications modulaires. Nous verrons un peu plus en détail par la suite comment cette technique fonctionne.

Outre ce nouvel aspect intéressant la protection des cryptosystèmes, le RNS offre également la possibilité de l'utilisation très simple de redondance via l'intégration de canaux supplémentaires, permettant ainsi de garantir un certain niveau d'intégrité des données traitées. Si ces RNS dits redondants ont été beaucoup étudiés jusqu'à présent, leur utilisation dans le cadre de calculs dans un grand corps fini $\mathbb{F}_p$ a été limitée par la nature de la réduction modulaire RNS. En effet, les conversions de base utilisées pour cette réduction modulaire brisent l'indépendance des résidus et, par suite, vont à l'encontre du principe des RNS redondants. Dans le chapitre qui suit, nous allons étudier dans quelle mesure il est possible de concilier réduction modulaire et RNS redondants, afin de contribuer à rendre le RNS toujours plus intéressant pour des applications cryptographiques.

Protection contre les attaques par faute

2

Sommaire

Ce chapitre introduit la notion de RNS redondants. Ceux-ci sont à la base de la construction d'une procédure de détection d'erreurs pouvant affecter des résidus. Nous énonçons les définitions essentielles dans un premier temps.

Puis nous détaillons les techniques effectives de détection de faute basées sur les RNS redondants, en exhibant les conditions que doivent respecter les moduli redondants pour garantir la détection de résidus erronés. Ces rappels et études préliminaires servent à établir un nouvel algorithme de réduction modulaire RNS doté d'une capacité de résistance aux attaques par injection de faute. Cette contribution originale permet d'élargir le cadre d'utilisation des RNS redondants, qui est initialement celui d'une arithmétique basique constituée d'additions et de multiplications, à un contexte d'arithmétique dans un corps fini.

2.1 Une arithmétique robuste pour la cryptographie asymétrique

2.1.1 Le RNS comme contre-mesure aux attaques par canaux auxiliaires

Les attaques par canaux auxiliaires (Side Channel Attacks, SCA) sont une sérieuse menace pour les cryptosystèmes implantés sur des matériels embarqués. Afin de recouvrer de l'information concernant des données secrètes, ces attaques reposent sur l'exploitation de caractéristiques mesurables depuis l'« extérieur », comme le temps d'exécution (Timing Attacks, Kocher (1996), Brumley et Boneh (2003)), ou encore la mesure de l'énergie consommée ou des émissions électromagnétiques (Simple Power Analysis). Ceci permet par exemple sur des implantations non protégées de RSA de lire directement les bits de l'exposant secret en exploitant la différence de consommation entre une multiplication et un carré modulaires. Pour contrer ce type d'attaque, il est possible de « lisser » les traces relevées en utilisant un algorithme adéquat d'exponentiation modulaire, comme l'échelle de Montgomery (Joye et Yen 2003).

D'autres types d'attaques se basent sur l'exploitation d'un ensemble de traces et consistent en une analyse statistique qui doit permettre d'extraire certaines corrélations entre les grandeurs mesurées et les données traitées en interne. C'est le cas par exemple de la Differential Power Analysis (Kocher et al. 1999; 2011, Coron 1999, Goubin 2002) concernant la mesure de consommation d'énergie, ou encore des attaques par analyse différentielle basées sur la mesure des émissions électromagnétiques (Gandolfi et al. 2001, Agrawal et al. 2003). Les contre-mesures classiques pour les attaques différentielles reposent en général sur un masquage aléatoire des données.

Dans ce contexte, les RNS constituent une arithmétique intéressante en tant qu'outil de protection contre les attaques par canaux auxiliaires. La remarque clef est que la représentation de Montgomery offre un masquage des données. Comme la multiplication modulaire nécessite l'utilisation de deux bases RNS copremières $\mathcal{B}$ et $\mathcal{B}'$ (supposées posséder le même nombre n de moduli par soucis de simplicité) et que la représentation de Montgomery associée à ce choix de bases est l'isomorphisme Λ_M de $\mathbb{Z}/m_1\mathbb{Z} \times \ldots \times \mathbb{Z}/m_{2n}\mathbb{Z}$ défini par :

$$\Lambda_M \circ \varphi_{\mathcal{B}\cup\mathcal{B}'}(x) = \varphi_{\mathcal{B}\cup\mathcal{B}'}\left(x \times |M|_p\right),$$

l'idée développée par Bajard et al. (2004) est d'ajouter un aléa à ce masquage en choisissant aléatoirement les bases $\mathcal{B}$ et $\mathcal{B}'$ parmi un ensemble prédéfini de $2n$ moduli $\{m_1, \ldots, m_{2n}\}$.

La qualité du masquage est notamment reliée au nombre de bases possibles pour le choix de $\mathcal{B}$, et donc aussi de $\mathcal{B}'$, qui est $\mathcal{C}_n^{2n} \sim \frac{2^{2n}}{\sqrt{\pi n}}$. Vu cette quantité, il n'est pas question de stocker les précalculs de toutes les valeurs $\left|M^2\right|_p$ possibles qui permettent de calculer les représentations de Montgomery. Bajard, Imbert, Liardet, et Teglia proposent une manière astucieuse pour calculer facilement ces représentations via une simple multiplication modulaire. Si $\mathcal{M}$ désigne le produit des $2n$ moduli disponibles et si deux bases $\mathcal{B}$ et $\mathcal{B}'$ sont choisies, alors nous avons l'égalité $\mathcal{M} = M \times M'$. Par suite :

$$\begin{aligned}\varphi^{-1}_{\mathcal{B}'\cup\mathcal{B}}\left(\text{MulModRNS}\left(\mathcal{B}', \mathcal{B}, x, |\mathcal{M}|_p, p\right)\right) &\equiv x \times \mathcal{M} \times (M')^{-1} \bmod p \\ &\equiv x \times M \bmod p.\end{aligned}$$

Pour obtenir la représentation de Montgomery de x associée à la base $\mathcal{B}$, il s'agit donc d'effectuer une multiplication modulaire entre x et $|\mathcal{M}|_p$ avec $\mathcal{B}'$ comme base principale.

Ce même type de démarche offre également la possibilité de tirages aléatoires intermédiaires au cours du calcul d'une exponentiation modulaire par exemple. Si $\mathcal{B}_\alpha$ et $\mathcal{B}'_\alpha$ sont les deux bases courantes, et $\mathcal{B}_\beta$ et $\mathcal{B}'_\beta$ les deux nouvelles bases, il s'agit donc de calculer :

$$\Lambda_{M_\beta} \circ \Lambda_{M_\alpha}^{-1} \circ \varphi_{\mathcal{B}_\alpha \cup \mathcal{B}'_\alpha} \left(x \times |M_\alpha|_p \right) = \varphi_{\mathcal{B}_\beta \cup \mathcal{B}'_\beta} \left(x \times |M_\beta|_p \right).$$

La procédure se fait en deux étapes. La première consiste à supprimer la première représentation de Montgomery :

$$\begin{aligned} y &= \varphi^{-1}_{\mathcal{B}_\alpha \cup \mathcal{B}'_\alpha} \left(\text{MulModRNS} \left(\mathcal{B}_\alpha, \mathcal{B}'_\alpha, x \times |M_\alpha|_p, 1, p \right) \right) \\ &\equiv x \times M_\alpha \times M_\alpha^{-1} \bmod p \\ &\equiv x \bmod p. \end{aligned} \tag{2.1}$$

Puis la seconde étape est le calcul de la nouvelle représentation :

$$\begin{aligned} z &= \varphi^{-1}_{\mathcal{B}'_\beta \cup \mathcal{B}_\beta} \left(\text{MulModRNS} \left(\mathcal{B}'_\beta, \mathcal{B}_\beta, y, |\mathcal{M}|_p, p \right) \right) \\ &\equiv y \times \mathcal{M} \times (M'_\beta)^{-1} \bmod p \\ &\equiv y \times M_\beta \bmod p \\ &\equiv x \times M_\beta \bmod p. \end{aligned} \tag{2.2}$$

L'efficacité de cette arithmétique résistante aux fuites (Leak Resistant Arithmetic, LRA) a été testée expérimentalement. Guillermin (2011) l'a par exemple intégrée dans l'implantation sur FPGA d'un RSA-CRT de 1024 bits avec échelle de Montgomery. Il a ainsi montré que le surcoût en terme de temps causé par l'intégration de cette technique de masquage reste négligeable ($+2,5\%$ cycles). Perin et al. (2014) ont également mis en application le LRA pour l'implantation d'un RSA et ont montré sa réelle efficacité contre des attaques de type SPA et DPA se basant sur les mesures des radiations électromagnétiques. Cette technique demande un compromis temps/surface concernant les nombreux précalculs inhérents au choix aléatoire des bases utilisées. Perin et al. ont été capables d'obtenir pour leur implantation un faible surcoût en temps, de l'ordre de 1%, pour un surcoût de mémoire de 92%.

2.1.2 Sensibilité des cryptosystèmes asymétriques aux attaques par injection de fautes

Complétant les SCA dans la panoplie du cryptanalyste, les attaques par faute peuvent être très efficaces. Historiquement, l'attaque dite de Bellcore proposée par Boneh et al. (1997) sur le cryptosystème RSA-CRT (Koç 1994) prouve le haut potentiel de dangerosité de ce type d'attaque, et illustre bien leur principe. Le fonctionnement du RSA-CRT est le suivant. Soit les paramètres privés constitués des grands nombres premiers p et q, et d'un exposant d, ainsi que les paramètres publics $N = pq$ et un entier e vérifiant $e \equiv d^{-1} \bmod \phi(N) = d^{-1} \bmod (p-1)(q-1)$. Si m est un message à signer

et h une fonction de hachage quelconque, alors la signature est obtenue en calculant dans un premier temps ses résidus modulo p et q :

$$\begin{cases} s_p = h(m)^{d \bmod (p-1)} \bmod p, \\ s_q = h(m)^{d \bmod (q-1)} \bmod q. \end{cases}$$

s est alors reconstruit par une simple application de la conversion MRS pour la base RNS à deux éléments $\{p, q\}$:

$$s = s_p + \left| \left(s_q - s_p\right) p^{-1} \right|_q \times p.$$

La stratégie de l'attaque par faute repose sur les constations suivantes. Si une faute perturbe le calcul du résidu s_q, donnant alors $\overline{s}_q = \left|s_q + e\right|_q$, alors la signature erronée a la forme suivante :

$$\overline{s} = s_p + \left| \left(s_q - s_p\right) p^{-1} + e p^{-1} \right|_q \times p = s + \left(\left| e p^{-1} \right|_q - \delta q \right) p, \ \delta \in \{0, 1\}.$$

Disposant d'un doublet $(s, \overline{s})$ de signatures d'un même message dont l'une d'elle a été altérée de la manière décrite précédemment, alors :

$$\overline{s} - s = \left(\left| e p^{-1} \right|_q - \delta q \right) p.$$

Par conséquent, le calcul de pgcd $(N, \overline{s} - s)$ révèle le facteur p et par suite la clef privée.

Une contre-mesure immédiate est de procéder à la vérification $s^e = h(m) \bmod N$ avant l'envoi de la signature s. Nombre de contre-mesures moins coûteuses ont été proposées, comme l'ajout de redondance permettant une vérification plus rapide (Shamir 1999, Aumüller et al. 2003, Vigilant 2008). Malgré les nombreuses études menées sur le sujet depuis l'attaque Bellcore, la problématique de la sensibilité du RSA-CRT contre les attaques par fautes fait toujours l'objet de recherches (Fouque et al. 2012).

Outre RSA, les cryptosystèmes basés sur les courbes elliptiques (Miller 1986, Koblitz 1987) ou les couplages (Page et Vercauteren 2006, Whelan et Scott 2007) sont tout autant concernés par les attaques par faute. Il s'agit donc d'un sujet sensible de la cryptographie en général, et de la cryptographie asymétrique en particulier.

L'objet de ce chapitre est de proposer une solution de protection basée sur une approche arithmétique exploitant les propriétés du RNS. La multiplication modulaire étant l'opération centrale des systèmes de cryptographie asymétrique précités, il va s'agir de proposer un nouvel algorithme doté d'une capacité de résistance à l'injection de fautes.

2.2 RNS redondants

L'étude des capacités de détection de faute des RNS redondants a été initiée par Cheney (1962). Le principe du test de cohérence, introduit par Watson et Hastings (1966), est désormais à la base des techniques classiques de détection de faute. Ce test est construit sur une opération de conversion de

base. Si l'état-de-l'art de la détection de faute en RNS s'est jusqu'à présent essentiellement appuyé sur l'hypothèse de conversions de base avec réduction complète, comme par exemple la procédure Bex_{mrs}, étudier l'impact du choix d'une autre conversion permettra une plus grande flexibilité dans l'utilisation des techniques de détection de fautes, ce qui sera de première importance lors de la création d'une multiplication modulaire en RNS redondant.

2.2.1 Définitions

Une définition générale d'un RNS redondant (RRNS) est donnée dans un premier temps, afin de poser un cadre formel précis. Par la suite, des conditions restrictives sur la redondance seront apportées lorsqu'elles seront nécessaires. Ces conditions dépendront du contexte qui sera alors précisé.

Définition 2.1 *Un RNS redondant est la donnée d'une base RNS $\mathcal{B} = \{m_1, \ldots, m_n\}$ dite principale, d'un entier $k \in \mathbb{N}$, et d'un ensemble de k moduli $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$. Par soucis de simplicité, M_R dénote l'entier $\prod_{i=1}^{k} m_{R,i}$.*

L'intervalle dynamique du RRNS est par définition l'ensemble $[\![0, \frac{MM_R}{M \wedge M_R}[\![$. Le sous-ensemble $[\![0, M[\![$ est appelé intervalle légitime, et $[\![M, \frac{MM_R}{M \wedge M_R}[\![$ est quant à lui dénommé intervalle illégitime.

Dans un tel système, tout nombre x de l'intervalle dynamique est représenté par ses résidus $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R})$. $\boldsymbol{x}_{\mathcal{B}}$ sont les résidus principaux de x, et $\boldsymbol{x}_{\mathcal{B}_R}$ sont ses résidus redondants. Le RRNS construit sur $\mathcal{B}$ et $\mathcal{B}_R$ est complètement défini par l'application suivante :

$$\begin{array}{rccl} \varphi_{\mathcal{B}\cup\mathcal{B}_R} : & [\![0, \frac{MM_R}{M \wedge M_R}[\![& \rightarrow & \mathcal{R}_M \times \mathcal{R}_{M_R} = \prod_{i=1}^{n} \mathbb{Z}/m_i\mathbb{Z} \times \prod_{i=1}^{k} \mathbb{Z}/m_{R,k}\mathbb{Z} \\ & x & \mapsto & (\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R}) = (\varphi_{\mathcal{B}}(x), \varphi_{\mathcal{B}_R}(x)). \end{array} \tag{2.3}$$

Remarque 2.1 *L'entier k donné dans la définition d'un RRNS va permettre de déterminer la capacité de détection d'erreur du système. Par capacité est entendu le nombre maximal d'erreurs détectables simultanément.*

La notion d'intervalle dynamique du RRNS se justifie de la manière suivante. Dans la Définition 2.1, si l'espace d'états total $\mathcal{R}_M \times \mathcal{R}_{M_R}$ est de cardinal MM_R, il n'a aucune raison *a priori* d'être en bijection avec l'intervalle $[\![0, MM_R[\![$. Le point essentiel dans la construction d'un RRNS est que l'intervalle dynamique $[\![0, M[\![$ de la base RNS principale $\mathcal{B}$ reste complètement et de manière univoque décrit dans l'espace d'états total, et que les caractéristiques de stabilité de $\mathcal{B}$ sous l'action des opérations arithmétiques élémentaires ne sont pas affectées. Le Théorème 2.1 s'attache à éclaircir ce point. Dans un premier temps, certaines propriétés directement déduites de la définition précédente sont énumérées dans le lemme suivant.

Lemme 2.1 *Vu la Définition 2.1, les propriétés suivantes sont vérifiées.*

1. *En identifiant $[\![0, \frac{MM_R}{M \wedge M_R}[\![$ à l'anneau $\mathbb{Z}/\frac{MM_R}{M \wedge M_R}\mathbb{Z}$, alors $\varphi_{\mathcal{B}\cup\mathcal{B}_R}$ est un morphisme d'anneaux.*
2. *Pour tout $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R}) \in \mathcal{R}_M \times \mathcal{R}_{M_R}$, $\text{Card}\left(\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}_R}(\{(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R})\}) \cap [\![0, M[\![\right) \in \{0, 1\}$.*

Démonstration. 1. La preuve que $\varphi_{\mathcal{B}\cup\mathcal{B}_R}$ est un morphisme est une conséquence du fait que $\frac{MM_R}{M\wedge M_R}$ est divisible par le ppcm de l'ensemble des moduli $\mathcal{B}\cup\mathcal{B}_R$. En effet, par définition, $MM_R = \prod_{i=1}^{n} m_i \prod_{j=1}^{k} m_{R,j}$. Ainsi, pour tout $i \in [\![1,n]\!]$ et tout $j \in [\![1,k]\!]$ nous déduisons alors immédiatement les égalités et l'implication suivantes :

$$\begin{cases} \dfrac{MM_R}{M\wedge M_R} = M\dfrac{M_R}{M\wedge M_R} = m_iM_i\dfrac{M_R}{M\wedge M_R} \in m_i\mathbb{Z} \\ \dfrac{MM_R}{M\wedge M_R} = m_{R,j}M_{R,j}\dfrac{M}{M\wedge M_R} \in m_{R,j}\mathbb{Z} \end{cases}$$
$$\Rightarrow \frac{MM_R}{M\wedge M_R} \in \bigcap_{i=1}^{n} m_i\mathbb{Z} \cap \bigcap_{j=1}^{k} m_{R,j}\mathbb{Z} = \text{ppcm}\left(m_1,\ldots,m_n,m_{R,1},\ldots,m_{R,k}\right)\mathbb{Z}.$$

2. $\mathcal{B}$ étant une base RNS, la seconde assertion découle du fait que $\varphi_{\mathcal{B}}$ est injective.

□

La seconde assertion du lemme précédent est une conséquence du fait que dans la définition que nous avons donnée des RRNS, les moduli redondants n'ont pas été supposés premiers entre eux deux à deux ni premiers avec les moduli principaux. Cela implique donc notamment que $\varphi_{\mathcal{B}\cup\mathcal{B}_R}$ peut ne pas être surjective. Mais sa restriction sur l'intervalle légitime $[\![0,M[\![$ reste par contre bien injective.

Nous avons fait le choix de ne pas supposer que $\mathcal{B}_R$ est une base RNS, et qui plus est qui serait première avec $\mathcal{B}$, afin de détailler précisément dans quelle mesure cela impacterait les techniques standards de détection de faute en RNS. L'intuition voudrait que cela ne serve pas notre cause, puisque d'un point de vue informationnel, il s'agit simplement d'une duplication d'une information déjà portée par les résidus principaux et/ou par plusieurs résidus redondants. D'ailleurs, à notre connaissance, les résultats de l'état-de-l'art font peu cas de ces considérations. La conclusion que nous tirerons de notre analyse abondera effectivement dans ce sens.

Le théorème suivant va permettre de poser les bases du principe du test de cohérence que nous avons évoqué précédemment.

Théorème 2.1 *Soit $\mathcal{B}\cup\mathcal{B}_R$ un RNS redondant. Alors pour tout triplet d'entiers $(x,y,z) \in [\![0, \frac{MM_R}{M\wedge M_R}[\![^3$ vérifiant $z = x\star y \bmod \frac{MM_R}{M\wedge M_R}$ avec $\star \in \{+,\times,-,\div\}$,*

$$z \in [\![0,M[\![\Rightarrow \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\{\left(\varphi_{\mathcal{B}}(x)\star\varphi_{\mathcal{B}}(y), \varphi_{\mathcal{B}_R}(x)\star\varphi_{\mathcal{B}_R}(y)\right)\}\right) = \{z\}. \tag{2.4}$$

Démonstration. Si $z \in [\![0,M[\![$, alors par définition de $\varphi_{\mathcal{B}\cup\mathcal{B}_R}$, qui de plus est un morphisme (assertion 1 du Lemme 2.1),

$$\varphi_{\mathcal{B}\cup\mathcal{B}_R}(z) = \left(\varphi_{\mathcal{B}}(x\star y), \varphi_{\mathcal{B}_R}(x\star y)\right) = \left(\varphi_{\mathcal{B}}(x)\star\varphi_{\mathcal{B}}(y), \varphi_{\mathcal{B}_R}(x)\star\varphi_{\mathcal{B}_R}(y)\right).$$

L'assertion 2 du Lemme 2.1 permet ensuite de conclure. □

Le Théorème 2.1 fournit un moyen de vérifier à partir des résidus dans $\mathcal{B}\cup\mathcal{B}_R$ que le résultat d'une suite d'opérations arithmétiques appartient à l'intervalle légitime. En effet, en notant $(z_{\mathcal{B}}, z_{\mathcal{B}_R})$ les résidus de $z = x\star y \bmod \frac{MM_R}{M\wedge M_R}$,

alors l'Équation 2.4 se réécrit :

$$\left[\varphi_{\mathcal{B}}^{-1}\left(\{z_{\mathcal{B}}\}\right) + M\mathbb{Z}\right] \cap \left[\varphi_{\mathcal{B}_R}^{-1}\left(\{z_{\mathcal{B}_R}\}\right) + M_R\mathbb{Z}\right] \cap [\![0, \frac{MM_R}{M \wedge M_R}[\![= \{z\}.$$

Vu l'injectivité de $\varphi_{\mathcal{B}}$, il vient alors :

$$z \in [\![0, M[\![\Rightarrow [\{z\} + M\mathbb{Z}] \cap \left[\varphi_{\mathcal{B}_R}^{-1}\left(\{z_{\mathcal{B}_R}\}\right) + M_R\mathbb{Z}\right] \cap [\![0, \frac{MM_R}{M \wedge M_R}[\![= \{z\}.$$

Or, par hypothèse, z appartient à l'intervalle dynamique $[\![0, M[\![$ de $\mathcal{B}$. Ainsi, $z = \varphi_{\mathcal{B}}^{-1}(z_{\mathcal{B}})$, et l'implication suivante vient immédiatement :

$$z \in [\![0, M[\![\Rightarrow \{z_{\mathcal{B}_R}\} = \varphi_{\mathcal{B}_R}\left(\{\varphi_{\mathcal{B}}^{-1}(z_{\mathcal{B}})\}\right). \tag{2.5}$$

L'implication (2.5) permet de définir un test fondamental dans l'utilisation des RRNS, appelé test de cohérence (« consistency check » dans la littérature anglophone).

Définition 2.2 *Soit $\mathcal{B} \cup \mathcal{B}_R$ un RNS redondant et $(z_{\mathcal{B}}, z_{\mathcal{B}_R})$ des résidus dans ce système. Le test de cohérence associé à la base $\mathcal{B} \cup \mathcal{B}_R$ et appliqué aux résidus $(z_{\mathcal{B}}, z_{\mathcal{B}_R})$ est défini par la Procédure TestCoherence.*

Procédure TestCoherence($\mathcal{B}$,$\mathcal{B}_R$,$x_{\mathcal{B}}$,$x_{\mathcal{B}_R}$)

```
1 début
2     x̃_{B_R} ← φ_{B_R}(φ_B^{-1}(x_B));
3     retourner x̃_{B_R} == x_{B_R}
```

En calculant $\tilde{z}_{\mathcal{B}_R} = \varphi_{\mathcal{B}_R}\left(\varphi_{\mathcal{B}}^{-1}(z_{\mathcal{B}})\right)$, l'inégalité $\tilde{z}_{\mathcal{B}_R} \neq z_{\mathcal{B}_R}$ permet alors de conclure que l'ensemble de résidus $(z_{\mathcal{B}}, z_{\mathcal{B}_R})$ ne correspond pas à un nombre de l'intervalle légitime. Ce test d'égalité justifie la notion d'intervalle légitime en la raccrochant à la notion de cohérence entre les résidus principaux et redondants.

Par la suite, les conditions sur la redondance seront affinées de manière à ce que, *a contrario*, si un tel test de cohérence réussit alors cela signifiera que les résidus testés sont effectivement ceux d'un nombre de l'intervalle légitime. Autrement dit, une réciproque à l'implication 2.4 du Théorème 2.1 pourra être construite, ce qui permettra d'achever la création d'une procédure de détection de faute.

2.2.2 Modèle de faute unique

Le modèle de faute considéré est défini dans cette partie. Le choix du modèle théorique se justifie par le fait qu'un canal RNS est assimilé à la structure mathématique d'anneau quotient $\mathbb{Z}/m\mathbb{Z}$ qu'il représente. Il est le modèle utilisé dans la majeure partie de l'état-de-l'art sur les RNS redondants (Mandelbaum 1972, Barsi et Maestrini 1973, Yau et L. 1973, Etzel et Jenkins 1980, Krishna et al. 1992). La manière dont une telle faute modifie l'entier de l'intervalle dynamique représenté par ces résidus va être détaillée.

Dans un premier temps, le concept de faute localisée est défini. Celui-ci justifie la pertinence du modèle théorique utilisé pour simuler les effets d'attaques physiques par injection de faute sur un matériel implantant une architecture RNS, de type Cox-Rower par exemple.

Définition 2.3 *Dans un système RNS, une faute localisée est une perturbation affectant une valeur appartenant à un canal $\mathbb{Z}/m\mathbb{Z}$, où m est un modulus du (des) RNS utilisé(s).*

Pour la suite de la discussion, les données initiales sont un RRNS $\mathcal{B} \cup \mathcal{B}_R$, $\mathcal{B} = \{m_1, \ldots, m_n\}$ et $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$, et un ensemble de résidus $(\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_{\mathcal{B}_R})$ associé à un nombre $x \in [\![0, \frac{MM_R}{M \wedge M_R}[\![$.

Définition 2.4 *Soit $i \in [\![1, n]\!]$ un indice. Une faute sur le résidu x_i de x est un entier $e_i \in]\!]0, m_i[\![$. L'ensemble des résidus de x affecté par ladite faute est noté $(\overline{\boldsymbol{x}}_\mathcal{B}, \overline{\boldsymbol{x}}_{\mathcal{B}_R})$, où $\overline{x}_j = x_j$ pour tout $j \in [\![1, n]\!] \backslash \{i\}$, $\overline{x}_{R,z} = x_{R,z}$ pour tout $z \in [\![1, k]\!]$, et*

$$\overline{x}_i = |x_i + e_i|_{m_i} = x_i + e_i - \delta_i m_i \in [\![0, m_i[\![, \text{ où } e_i \in]\!]1, m_i[\![\text{ et } \delta_i \in \{0, 1\}. \tag{2.6}$$

De la même manière, un résidu redondant peut être affecté par une faute, et les notations restent similaires.

Remarque 2.2 *La pertinence de ce modèle de faute repose sur l'indépendance fonctionnelle des unités RNS au sein desquelles sont réalisés les calculs menés dans les anneaux $\mathbb{Z}/m_i\mathbb{Z}$. Ce modèle simule une perturbation localisée sur une seule de ces unités.*

À partir de ces définitions, seuls les résidus pouvant contenir au plus une erreur sur l'un d'entre eux vont être considérés. L'hypothèse suivante fixe précisément le contexte de la suite de l'étude.

Hypothèse 2.1 *De l'entier x issu de l'intervalle légitime $[\![0, M[\![$ et de ses résidus $(\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_{\mathcal{B}_R})$ dans le RRNS $\mathcal{B} \cup \mathcal{B}_R$, seuls sont connus les résidus suivants :*

$$(\overline{\boldsymbol{x}}_\mathcal{B}, \overline{\boldsymbol{x}}_{\mathcal{B}_R}) = \begin{cases} (\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_{\mathcal{B}_R}) \\ \textit{ou} \\ (\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_{\mathcal{B}_R}) + 1 \textit{ faute.} \end{cases} \tag{2.7}$$

Autrement dit, ceux-ci diffèrent de $(\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_{\mathcal{B}_R})$ sur au plus un résidu.

La Proposition 2.1 suivante montre comment une faute unique sur un ensemble de résidus transforme l'entier qu'ils représentent. En allant plus loin, la Proposition 2.2 précisera l'ensemble exact de tous les entiers représentables par les résidus de x affectés par une faute unique, ce qui permettra de fixer des conditions nécessaires et suffisantes sur M_R rendant possible la détection de toute faute unique en utilisant la procédure du test de cohérence de la Définition 2.2.

Proposition 2.1 *Soit $i \in [\![1, n]\!]$ un indice, $\overline{\boldsymbol{x}}_\mathcal{B}$ les résidus de $x \in [\![0, M[\![$ dont le i-ème est affecté par une faute $e_i \in]\!]0, m_i[\![$, et $\overline{x} = \varphi_\mathcal{B}^{-1}(\overline{\boldsymbol{x}}_\mathcal{B}) \in [\![0, M[\![$ l'entier de $[\![0, M[\![$ représenté par les résidus principaux erronés $\overline{\boldsymbol{x}}_\mathcal{B}$ de x. Alors,*

$$\exists a_i \in]\!] - m_i, m_i[\![\backslash \{0\}, \ \overline{x} = x + a_i M_i \in [\![0, M[\![. \tag{2.8}$$

De plus, $a_i \equiv e_i M_i^{-1} \bmod m_i$.

Démonstration. Notons y l'entier $y = |\overline{x} - x|$. En particulier, nous avons donc $y \in [\![0, M[\![$. y est de plus non nul. Pour montrer cette affirmation, il suffit d'exhiber un résidu non nul de y. Deux cas de figure sont à considérer. Si $\overline{x} < x$, alors $y = x - \overline{x}$ et donc $y_i = |x_i - \overline{x}_i|_{m_i} = |x_i - (x_i + e_i)|_{m_i} = |-e_i|_{m_i} \neq 0$. De la même manière, si $x < \overline{x}$, alors il vient $y_i = |\overline{x}_i - x_i|_{m_i} = |e_i|_{m_i} \neq 0$.

y étant un élément de l'intervalle dynamique de $\mathcal{B}$, il est par conséquent complètement défini par ses résidus dans $\mathcal{B}$. Or, pour tout $j \in [\![1, n]\!] \backslash \{i\}$, $y_i = 0$. Ainsi, M_i divise y. Écrivant alors $y = bM_i$, comme $y \in [\![0, M[\![$ il est clair que $b \in [\![0, m_i[\![$. Ainsi, $b = |b|_{m_i}$. Il suffit donc de calculer son résidu modulo m_i, soit, vu la valeur de y_i calculée précédemment :

$$|b|_{m_i} = \left|yM_i^{-1}\right|_{m_i} = \left|yM_i^{-1}\right|_{m_i} = \left|\pm e_i M_i^{-1}\right|_{m_i}.$$

Ainsi, si $x < \overline{x}$, alors $y = \left|e_i M_i^{-1}\right|_{m_i} M_i$. Par suite, $\overline{x} = x + \left|e_i M_i^{-1}\right|_{m_i} M_i \in [\![0, M[\![$. De même, si $\overline{x} < x$ alors $y = \left|-e_i M_i^{-1}\right|_{m_i} M_i$ et donc $\overline{x} = x - \left|-e_i M_i^{-1}\right|_{m_i} M_i = x + \left(\left|e_i M_i^{-1}\right|_{m_i} - m_i\right) M_i \in [\![0, M[\![$.

□

L'injection d'une faute unique sur les résidus $\boldsymbol{x}_{\mathcal{B}}$ de x n'est finalement rien d'autre que le résultat de l'addition à x modulo M d'un entier de la forme suivante :

$$e = \varphi_{\mathcal{B}}^{-1}(0, \ldots, 0, e_i, 0, \ldots, 0) = \left|e_i M_i^{-1}\right|_{m_i} M_i.$$

Ainsi,

$$\overline{x} = |x + e|_M = x + \left(\left|e_i M_i^{-1}\right|_{m_i} - \delta m_i\right) M_i,\ \delta \in \{0, 1\}.$$

Proposition 2.2 *Soit $i \in [\![1, n]\!]$ et $x \in [\![0, M[\![$. Alors tout entier $z \in [\![0, M[\![$ tel que $x - z \in M_i\mathbb{Z}$ et $z \neq x$ peut être obtenu des résidus de x affectés d'une erreur non nulle sur le i-ème d'entre eux.*

Démonstration. Par hypothèse, il existe un entier a_i tel que $z = x + a_i M_i$ avec $0 < |a_i| < m_i$. Ainsi, en considérant la faute $e_i = |a_i M_i|_{m_i}$ sur le résidu x_i, la construction de l'entier $\overline{x}$ issu de $\overline{x}_{\mathcal{B}}$ établie dans la preuve de la Proposition 2.1 donne :

$$\overline{x} = x + \left(|a_i|_{m_i} - \delta m_i\right) M_i \in [\![0, M[\![,$$

où $\delta \in \{0, 1\}$. Or, suivant le signe de a_i, $|a_i|_{m_i} = a_i$ si $a_i \geqslant 0$, et $|a_i|_{m_i} = m_i + a_i$ sinon. Par conséquent, $\overline{x} = x + (\delta' m_i + a_i - \delta m_i) M_i$ avec $\delta, \delta' \in \{0, 1\}$ et $\delta' = 1$ seulement si $a_i < 0$.

Par suite, $|\overline{x} - z| = |\delta' - \delta| M$, et comme $(\overline{x}, z) \in [\![0, M[\![^2$, il en découle que $\delta' = \delta$, et donc que $\overline{x} = z$. □

L'ensemble des entiers de l'intervalle dynamique atteignables par une faute sur le i-ième résidu de x est illustré par la Figure 2.1.

2.2.3 Redondance nécessaire et suffisante pour la détection des fautes uniques

Étant donné un RRNS $\mathcal{B} \cup \mathcal{B}_R$, le principe de détection de faute sur les résidus principaux le plus efficace repose sur la procédure `TestCoherence`

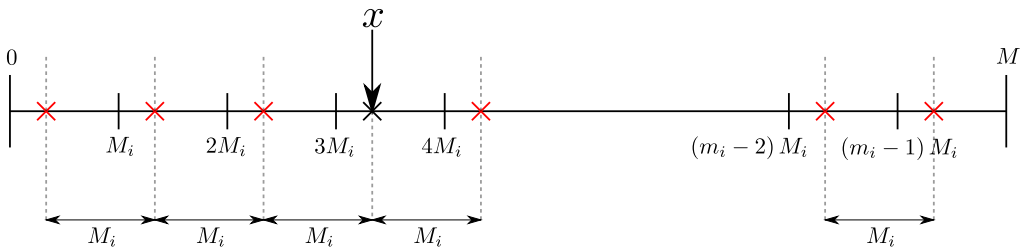


FIGURE 2.1 – *Illustration de l'ensemble des valeurs (marquées d'une croix rouge) atteignables par une faute sur le i-ième résidu de x.*

introduite dans la Définition 2.2, et définie initialement par Watson et Hastings (1966). Comme précisé précédemment, l'incohérence d'un ensemble de résidus $(\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_R)$ signifie que l'entier x n'est pas un élément de l'intervalle légitime $[\![0, M[\![$.

Certains auteurs des premiers résultats sur les RNS redondants (Mandelbaum 1972, Etzel et Jenkins 1980) avaient adopté une approche plutôt brutale, à savoir la reconstruction complète de l'entier défini par les résidus $(\overline{\boldsymbol{x}}_\mathcal{B}, \overline{\boldsymbol{x}}_R)$. En effet, constatant que lorsque $\mathcal{B} \cup \mathcal{B}_R$ constitue une base RNS toute faute unique sur le i-ième résidu de $(\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_R)$ donne les résidus $(\overline{\boldsymbol{x}}_\mathcal{B}, \overline{\boldsymbol{x}}_R)$ d'un nombre de la forme $\overline{x} = \varphi^{-1}_{\mathcal{B}\cup\{m_R\}}\left((\overline{\boldsymbol{x}}_\mathcal{B}, \overline{\boldsymbol{x}}_R)\right) = x + a_i M_i M_R$, alors si la redondance M_R est suffisamment grande, tout nombre de cette forme doit appartenir à l'intervalle illégitime. Il suffit alors de reconstruire le nombre $\overline{x}$ et de le comparer à M.

Le test de cohérence n'est en fait qu'une vision purement RNS de cette approche. En effet, comme l'entier $\overline{x}$ est supposé être un élément de l'intervalle légitime lorsqu'il est vérifié, il doit être complètement défini par ses résidus dans la base principale $\mathcal{B}$. Ainsi, l'inégalité $\varphi_{\mathcal{B}_R} \circ \varphi^{-1}_\mathcal{B}\left(\overline{\boldsymbol{x}}_\mathcal{B}\right) \neq \overline{\boldsymbol{x}}_R$ signifie bien que $\overline{x}$ n'est pas dans l'intervalle légitime.

En pratique, la composée d'applications $\varphi_{\mathcal{B}_R} \circ \varphi^{-1}_\mathcal{B}$ est réalisée par une opération de conversion de base. Le choix de cette opération va influer sur la forme de la redondance à utiliser pour assurer la détection de toute faute unique. Afin de garantir la cohérence entre des résidus principaux et redondants intègres, la conversion de base doit utiliser une réduction complète. C'est le cas par exemple de la conversion Bex_{mrs}, qui a été intensivement étudiée et utilisée dans l'état-de-l'art sur les RNS redondants où la conversion de base est toujours considérée complètement réduite (Mandelbaum 1972, Etzel et Jenkins 1980, Yau et L. 1973, Krishna et al. 1992). Mais c'est aussi le cas de Bex_{sk}, ainsi que de Bex_{kwc} pour un sous-ensemble de l'intervalle légitime. L'utilisation de ces conversions dans le cadre de la multiplication modulaire RNS justifie une étude détaillée de leur impact si elles sont mises en œuvre pour le test de cohérence.

Dans un premier temps, l'application $\varphi_{\mathcal{B}_R} \circ \varphi^{-1}_\mathcal{B}$ est considérée en dehors de tout choix pratique d'opérateur de changement de base afin de déterminer quelle redondance est nécessaire et suffisante pour la détection dans un contexte purement théorique. Si ces résultats apparaissent dans nombre des travaux précédemment cités, ils sont ici énoncés dans le cadre très général où M_R n'est pas choisi *a priori* premier à M. Ceci permettra de montrer que si cette condition n'est pas nécessaire pour garantir la détection des fautes, le fait de ne pas l'imposer ne fait également rien perdre de la capacité de détection.

Théorème 2.2 *Soit $\mathcal{B} = \{m_1, \ldots, m_n\}$ une base RNS, m_R un modulus et $i \in [\![1, n]\!]$ un indice. Alors*

$$\forall (x, e) \in [\![0, M[\![\times]\!]0, m_i[\![,$$
$$\varphi_{\mathcal{B}}^{-1}\left(\left(x_1, \ldots, x_{i-1}, |x_i + e|_{m_i}, x_{i+1}, \ldots, x_n\right)_{\mathcal{B}}\right) \bmod m_R \neq x \bmod m_R$$
$$\Leftrightarrow$$
$$m_R \geqslant m_i \left(m_R \wedge M_i\right). \tag{2.9}$$

Démonstration.

- Prouvons la suffisance par contraposition. Nous supposons donc qu'il existe une faute unique sur les résidus principaux de x pour laquelle $\varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) \bmod m_R = x \bmod m_R = x_R$. Notant $\overline{x}$ l'entier $\overline{x} = \varphi_{\mathcal{B}}^{-1}\left(\left(x_1, \ldots, x_{i-1}, |x_i + e|_{m_i}, x_{i+1}, \ldots, x_n\right)_{\mathcal{B}}\right) \in [\![0, M[\![$ où $e \in \mathbb{Z}$ est l'erreur supposée affecter le résidu x_i, avec $|e|_{m_i} \neq 0$, nous supposons donc l'existence d'un entier b tel que $\overline{x} - x = bm_R$. Par la Proposition 2.1 il existe $a_i \in]\!] - m_i, m_i[\![$ tel que $\overline{x} = x + a_i M_i$. Par conséquent, $bm_R = a_i M_i$. Ainsi, $\frac{m_R}{m_R \wedge M_i}$ divise a_i, ce qui implique que $m_R < m_i \left(m_R \wedge M_i\right)$.
- Prouvons la nécessité par contraposition également. Donnons-nous pour ce faire un indice $i \in [\![1, n]\!]$ pour lequel il est supposé que $m_R < m_i \left(m_R \wedge M_i\right)$. Alors en posant $x = 0$ et $\overline{x}_i = \left|\frac{m_R}{m_R \wedge M_i} M_i\right|_{m_i}$,

$$\overline{x} = \varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = \left|\overline{x}_i M_i^{-1}\right|_{m_i} M_i = \left|\frac{m_R}{m_R \wedge M_i}\right|_{m_i} M_i.$$

Comme $\frac{m_R}{m_R \wedge M_i} < m_i$, il vient alors :

$$\overline{x} = \frac{m_R}{m_R \wedge M_i} M_i = m_R \frac{M_i}{m_R \wedge M_i} \equiv 0 \bmod m_R \equiv x \bmod m_R.$$

La preuve est alors achevée.

□

Remarque 2.3 *La Condition (2.9) montre que pour avoir une redondance minimale, m_R doit être choisi premier à la base principale $\mathcal{B}$.*

L'influence du choix d'une opération de conversion de base pour la mise en œuvre pratique de la détection peut amener à modifier la Condition (2.9). La technique pratique de détection est décrite par la Procédure DetectOneErr, qui est simplement la mise en œuvre de la Procédure TestCoherence en utilisant une opération de conversion de base.

Il est supposé que la faute éventuelle est apparue avant la procédure de détection. Le problème soulevé par l'hypothèse de la modification d'une valeur dans un canal $\mathbb{Z}/m_i\mathbb{Z}$ pendant la conversion de base sera discuté dans la Partie 2.3.1.

Procédure DetectOneErr(Bex,$\mathcal{B}$,m_R,$\boldsymbol{x}_{\mathcal{B}}$,$x_R$)

```
1 début
2     x̂_R ← Bex(B, {m_R}, x_B);
3     retourner x̂_R == x_R
```

2.2.4 Théorème fondamental de détection pratique de faute unique

Cette partie fournit la description de l'ensemble des valeurs que peut retourner une conversion de base lorsqu'elle est appliquée à un ensemble de résidus affecté par une faute. Ceci va permettre d'établir quelle est la redondance suffisante, voire nécessaire, garantissant la détection pratique de toute faute unique.

Test de cohérence basé sur Bex_{mrs}

Faute sur un résidu x_i La conversion de base utilisant le MRS a l'avantage d'être toujours complètement réduite malgré la présence d'une faute. En effet, si $\overline{x}_i = |x_i + e_i|_{m_i} \neq x_i$ et $\overline{x}_j = x_j$ pour tout $j \in [\![1, n]\!] \backslash \{i\}$, alors la proposition suivante est vérifiée :

Proposition 2.3 *Si e_i parcourt l'ensemble $[\![1, m_i - 1]\!]$ et si $\overline{x} = \mathrm{Bex}_{mrs}(\mathcal{B}, \overline{\boldsymbol{x}}_{\mathcal{B}})$, alors la quantité $\frac{\overline{x}-x}{M_i}$ parcourt l'ensemble $[\![-\xi_{x,i,\mathcal{B}}, m_i - 1 - \xi_{x,i,\mathcal{B}}]\!] \backslash \{0\}$, où il est rappelé que $\xi_{x,i,\mathcal{B}} = \left|x_i M_i^{-1}\right|_{m_i}$.*

Démonstration. L'opération Bex_{mrs} convertit complètement l'entier $\overline{x}$ défini par les résidus $\overline{\boldsymbol{x}}_{\mathcal{B}}$. Ainsi, par la Proposition 2.1, $\overline{x} = x + a_i M_i \in [\![0, M[\![$ avec $a_i = \left|e_i M_i^{-1}\right|_{m_i} - \delta m_i \neq 0$ et où l'entier $\delta \in \{0, 1\}$ est défini de manière à ce que l'égalité $a_i + \xi_{x,i,\mathcal{B}} = \left|e_i M_i^{-1} + \xi_{x,i,\mathcal{B}}\right|_{m_i}$ soit vérifiée. Par conséquent, lorsque e_i parcourt l'ensemble $[\![1, m_i - 1]\!]$, alors $\left|e_i M_i^{-1} + \xi_{x,i,\mathcal{B}}\right|_{m_i}$ parcourt $[\![0, m_i[\![\backslash \{\xi_{x,i,\mathcal{B}}\}$ et donc $a_i + \xi_{x,i,\mathcal{B}}$ parcourt $[\![-\xi_{x,i,\mathcal{B}}, m_i - 1 - \xi_{x,i,\mathcal{B}}]\!] \backslash \{0\}$, ce qui conclut la preuve. □

Faute sur un coefficient MRS $\tilde{x}_i$ La conversion Bex_{mrs} introduisant la représentation alternative MRS, l'effet d'une faute injectée sur un coefficient MRS est analysé. Un tel effet est différent de celui d'une faute sur un résidu. En effet, lorsqu'une faute affecte un résidu RNS de la quantité $\boldsymbol{x}$ à convertir, elle perturbe tous les coefficients MRS $\tilde{x}_j$ pour lesquels $j \geqslant i$. Mais la représentation MRS obtenue au final est exactement celle de la quantité $\overline{x}$ représentée par les résidus RNS $\overline{\boldsymbol{x}}_{\mathcal{B}}$.

La quantité de redondance nécessaire et suffisante pour détecter ce type d'erreur découle de la Proposition 2.4 suivante.

Proposition 2.4 *Soit $i \in [\![1, n]\!]$ tel que $\overline{\tilde{x}}_i = |\tilde{x}_i + e_i|_{m_i}$. Lorsque e_i parcourt l'ensemble $[\![1, m_i - 1]\!]$, la quantité $\frac{\overline{x}-x}{m_1 \dots m_{i-1}}$, où $\overline{x}$ est obtenu par la Formule (1.11) de reconstruction d'un entier à partir de ses résidus MRS dans la base MRS $\{1, m_1, m_1 m_2, \dots, m_1 \dots m_{n-1}\}$ utilisée avec les coefficients MRS erronés de x, parcourt l'ensemble $[\![-\tilde{x}_i, m_i - 1 - \tilde{x}_i]\!] \backslash \{0\}$.*

Démonstration. Par définition de $\overline{x}$,

$$\overline{x} = \sum_{j=1}^{n} \tilde{x}_j m_1 \dots m_{j-1} + e m_1 \dots m_{i-1} = x + e m_1 \dots m_{i-1}, \text{ où } e = |\tilde{x}_i + e_i|_{m_i} - \tilde{x}_i.$$

L'application $e_i \mapsto |\tilde{x}_i + e_i|_{m_i} - \tilde{x}_i$ étant une bijection de $[\![1, m_i - 1]\!]$ sur $[\![-\tilde{x}_i, m_i - 1 - \tilde{x}_i]\!] \backslash \{0\}$, ceci conclut la preuve. □

Test de cohérence basé sur Bex_{sk}

De même que l'opération Bex_{mrs}, la conversion Bex_{sk} peut être utilisée pour effectuer une détection de faute, puisque elle est complète lorsque les résidus sont intègres. Néanmoins, la situation est différente en présence d'une faute puisqu'un modulus redondant supplémentaire m_{sk} est adjoint à la base principale $\mathcal{B}$. Et comme celui-ci ne joue pas le même rôle que les autres moduli de $\mathcal{B}$ durant la conversion, la condition nécessaire et suffisante (2.9) doit être adaptée en conséquence. Pour ce faire, l'effet créé par une faute sur un résidu de la valeur calculée par Bex_{sk} est étudié en détail.

Faute sur le résidu x_{sk} Si $\overline{x}_{sk} = |x_{sk} + e_{sk}|_{m_{sk}} \neq x_{sk}$ et $\overline{x}_i = x_i$ pour tout $i \in [\![1, n]\!]$, alors les égalités suivantes sont vérifiées :

$$\begin{aligned}
\mathrm{sum}_{\mathcal{B}}\left(\overline{\boldsymbol{x}}_{\mathcal{B}}\right) &= \mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right), \\
\kappa_{\mathcal{B}}\left(\overline{\boldsymbol{x}}_{\mathcal{B}}\right) &= \left(\left(\mathrm{sum}_{\mathcal{B}}\left(\overline{\boldsymbol{x}}\right) - \overline{x}_{sk}\right) M^{-1}\right) \bmod m_{sk} \\
&= \left(\left(\mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - x_{sk} - e_{sk}\right) M^{-1}\right) \bmod m_{sk} \\
&= \left(\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - e_{sk} M^{-1}\right) \bmod m_{sk} \\
&= \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) - \left|e_{sk} M^{-1}\right|_{m_{sk}} + \delta_{sk} m_{sk} \in [\![0, m_{sk}[\![, \text{ où } \delta_{sk} \in \{0, 1\}.
\end{aligned} \tag{2.10}$$

Par conséquent,

$$\begin{cases}
\overline{x} = \mathrm{Bex}_{sk}\left(\mathcal{B} \cup \{m_{sk}\}, \left(\boldsymbol{x}_{\mathcal{B}}, \overline{x}_{sk}\right)\right) = x + a_{sk} M, \\
a_{sk} = \left|e_{sk} M^{-1}\right|_{m_{sk}} - \delta_{sk} m_{sk}, \text{ avec } \delta_{sk} \in \{0, 1\}.
\end{cases} \tag{2.11}$$

Par une telle faute, les résidus $(\boldsymbol{x}, \overline{x}_{sk})$ représentent donc un nombre $\overline{x} \in [\![M, m_{sk} M[\![$, congru à x modulo M. La proposition suivante précise la forme de $\overline{x}$.

Proposition 2.5 *Si une faute non nulle affecte x_{sk} uniquement, alors $|\overline{x} - x|$ est un multiple de M. De plus, lorsque e_{sk} parcourt l'ensemble $[\![1, m_{sk}[\![$, la quantité $\frac{|\overline{x} - x|}{M}$, où $\overline{x}$ désigne la valeur $\overline{x} = \mathrm{Bex}_{sk}\left(\mathcal{B} \cup \{m_{sk}\}, \left(\boldsymbol{x}_{\mathcal{B}}, |x_{sk} + e_{sk}|_{m_{sk}}\right)\right)$, parcourt l'ensemble :*

$$[\![1, \max\left(\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right), m_{sk} - 1 - \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)\right)]\!].$$

Démonstration. Vu les Équations (2.10) et (2.11), il est clair que lorsque e_{sk} parcourt l'ensemble $[\![1, m_{sk}[\![$, alors $\kappa_{\mathcal{B}}\left(\overline{\boldsymbol{x}}_{\mathcal{B}}\right) = \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) + a_{sk}$ où a_{sk} parcourt $[\![-\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right), m_{sk} - 1 - \kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right)]\!] \backslash \{0\}$. Ceci achève la preuve. □

Remarque 2.4 *En particulier, lorsque $\kappa_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) = 0$, alors $\overline{x}$ est de la forme $x + aM$ où a peut prendre toute valeur de $[\![1, m_{sk} - 1]\!]$.*

Faute sur un résidu x_i**, pour** $i \in [\![1, n]\!]$ Si $\overline{x}_i = |x_i + e_i|_{m_i} \neq x_i$, $\overline{x}_j = x_j$ pour tout $j \in [\![1, n]\!] \backslash \{i\}$ et $\overline{x}_{sk} = x_{sk}$ alors :

$$\mathrm{sum}_{\mathcal{B}}\left(\overline{\boldsymbol{x}}_{\mathcal{B}}\right) = \mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) + \left(\left|e_i M_i^{-1}\right|_{m_i} - \delta_i m_i\right) M_i = \mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}\right) + e M_i \tag{2.12}$$

avec $e = \left(\left|e_i M_i^{-1}\right|_{m_i} - \delta_i m_i\right) \in]\!] - m_i, m_i [\![\backslash \{0\}$ et $\delta_i \in \{0,1\}$ est tel que $e = \left|(x_i + e_i) M_i^{-1}\right|_{m_i} - \left|x_i M_i^{-1}\right|_{m_i}$.

De plus,

$$\begin{aligned} \kappa_{\mathcal{B}}(\overline{\boldsymbol{x}}_{\mathcal{B}}) &= \left(\left(\mathrm{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) + eM_i - x_{sk}\right) M^{-1}\right) \bmod m_{sk} \\ &= \left(\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) + e m_i^{-1}\right) \bmod m_{sk} \\ &= \kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) + f \end{aligned} \tag{2.13}$$

avec $f = \left|e m_i^{-1}\right|_{m_{sk}} - \delta_{sk} m_{sk} \in]\!] - m_{sk}, m_{sk} [\![$, et $\delta_{sk} \in \{0,1\}$.

Nous obtenons alors :

$$\overline{x} = \mathrm{Bex}_{sk}(\mathcal{B} \cup \{m_{sk}\}, \boldsymbol{x}_{\mathcal{B}}, \overline{x}_{sk}) = x + (e - f m_i) M_i. \tag{2.14}$$

Il découle des constatations précédentes que $|e - f m_i| < m_i + (m_{sk} - 1) m_i = m_i m_{sk}$, et $|e - f m_i|_{m_{sk}} = 0$. Ainsi, il existe un entier $c \in]\!]0, m_i[\![$ tel que $|e - f m_i| = c m_{sk}$.

Nous venons de montrer la proposition suivante.

Proposition 2.6 *Quelle que soit la faute non nulle e_i sur le résidu x_i, la quantité $\overline{x} - x$ est un multiple de $m_{sk} M_i$. De plus, $0 < \frac{|\overline{x} - x|}{m_{sk} M_i} < m_i$.*

Test de cohérence basé sur $\mathrm{Bex}_{kw,h}$

La conversion de base $\mathrm{Bex}_{kw,h}$ peut introduire une correction lors du calcul approché de $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ via l'ajout dans le registre d'accumulation du Cox d'une valeur α_{kw}. Ceci garantit une conversion complète dès que x vérifie la condition $x < (1 - \alpha_{kw}) M$. Dans ce cas précis, un test de cohérence peut être effectué sous l'hypothèse que tout ensemble de résidus intègres $(\boldsymbol{x}_{\mathcal{B}}, x_R)$ représente un entier de l'intervalle $[\![0, (1 - \alpha_{kw}) M[\![$.

Nous rappelons que cette conversion consiste précisément à calculer les quantités suivantes :

$$\mathrm{sum}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) = \sum_{i=1}^{n} \xi_{x,i,\mathcal{B}} M_i \text{ et } \tilde{\kappa}_{\mathcal{B}}(\boldsymbol{x}) = \lfloor \sum_{i=1}^{n} \frac{\mathrm{trunc}_h(\xi_{x,i,\mathcal{B}})}{2^h} + \alpha_{kw} \rfloor \tag{2.15}$$

où $\xi_{x,i,\mathcal{B}} = \left|x_i M_i^{-1}\right|_{m_i}$.

Une faute sur le résidu x_i peut être étudiée par son effet sur la quantité $\xi_{x,i,\mathcal{B}}$, ce qui ne change pas le fond de l'analyse. Considérons donc que nous avons :

$$\overline{\xi}_{x,i,\mathcal{B}} = \xi_{x,i,\mathcal{B}} + e_i, \text{ avec } e_i \in [\![-\xi_{x,i,\mathcal{B}}, m_i - \xi_{x,i,\mathcal{B}} - 1]\!] \backslash \{0\}.$$

Une telle faute va agir sur les résultats des calculs de $\mathrm{sum}_{\mathcal{B}}(\overline{\boldsymbol{x}})$ et de $\kappa_{\mathcal{B}}(\overline{\boldsymbol{x}})$. Ainsi, la quantité obtenue par la conversion a la forme $\overline{x} = x + e_i M_i + \delta M$, où δ peut prendre pour valeurs 0 ou ± 1.

Proposition 2.7 *Lorsque e_i parcourt l'ensemble $[\![-\xi_{x,i,\mathcal{B}}, m_i - \xi_{x,i,\mathcal{B}} - 1]\!]$, la quantité $\frac{|\overline{x} - x|}{M_i}$, où $\overline{x} = \mathrm{Bex}_{kwc}(\mathcal{B}, \overline{\boldsymbol{x}}_{\mathcal{B}}, \alpha_{kw})$, est strictement majorée par m_i et est non nulle.*

Démonstration. Nous réutilisons la Formule (2.15). Si $e_i > 0$, alors nous avons :

$$\text{trunc}_h\left(\overline{\xi}_{x,i,\mathcal{B}}\right) = \frac{\xi_{x,i,\mathcal{B}} + e_i - |\xi_{x,i,\mathcal{B}} + e_i|_{2^{r-h}}}{2^{r-h}} = \text{trunc}_h\left(\xi_{x,i,\mathcal{B}}\right) + \text{trunc}_h\left(e_i\right) + \delta,$$

où $\delta \in \{0,1\}$ ($\delta = 1$ si, et seulement si, $|\xi_{x,i,\mathcal{B}}|_{2^{r-h}} + |e_i|_{2^{r-h}} \geqslant 2^{r-h}$). Ainsi, $\text{eval}_h\left(\overline{\xi}_{x,i,\mathcal{B}}\right) = \text{eval}_h\left(\xi_{x,i,\mathcal{B}}\right) + \text{eval}_h\left(e_i\right) + \delta$. Par conséquent,

$$\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{x}}\right) = \lfloor \sum_{j=1}^{n} \text{eval}_h\left(\xi_{x,j,\mathcal{B}}\right) + \text{eval}_h\left(e_i\right) + \frac{\delta}{2^h} + \alpha_{kw} \rfloor.$$

De la même manière, si $e_i < 0$ alors :

$$\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{x}}\right) = \lfloor \sum_{j=1}^{n} \text{eval}_h\left(\xi_{x,j,\mathcal{B}}\right) - \text{eval}_h\left(|e_i|\right) - \frac{\delta}{2^h} + \alpha_{kw} \rfloor,$$

où $\delta = 1$ si, et seulement si, $|\xi_{x,i,\mathcal{B}}|_{2^{r-h}} < |e_i|_{2^{r-h}}$. Étant donné que $\text{eval}_h\left(|e_i|\right) \leqslant 1 - \frac{1}{2^h}$, alors il vient que $\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{x}}\right) = \kappa_{\mathcal{B}}\left(\overline{\boldsymbol{x}}\right) + \mu$ avec $\mu \in \{0, \pm 1\}$. De plus, il vient immédiatement que $\mu = 1$ seulement si $e_i < 0$, et $\mu = -1$ seulement si $e_i > 0$.

Finalement, comme nous avons de plus que $\text{sum}_{\mathcal{B}}\left(\overline{\boldsymbol{x}}_{\mathcal{B}}\right) = \text{sum}_{\mathcal{B}}\left(\overline{\boldsymbol{x}}_{\mathcal{B}}\right) + e_i M_i$, alors nous déduisons finalement de ce qui précède que $\overline{x} = x + \left(e_i + \mu m_i\right) M_i$, avec $|e_i + \mu m_i| < m_i$. De plus, comme e_i vérifie $0 < |e_i| < m_i$ (cf. (2.15)), alors $e_i + \mu m_i \neq 0$. □

Théorème fondamental pour la détection de faute unique

Théorème 2.3 *Soit $\mathcal{B} \cup \{m_R\}$ un RRNS, où un modulus redondant m_{sk} est adjoint à la base principale $\mathcal{B}$ si nécessaire. Le test de cohérence basé sur* Bex_{mrs} *(respectivement* Bex_{sk} *ou* Bex_{kwc}*) permet de détecter toute faute unique sur tout ensemble de résidus $\left(\boldsymbol{x}_{\mathcal{B}}, x_{m_R}\right)$ si, et seulement si, (respectivement si)*

$$\forall m \in \mathcal{B},\ m_R \geqslant m \times \left(m_R \wedge \frac{M}{m}\right). \tag{2.16}$$

Démonstration. Les Propositions 2.3, 2.5, 2.6 et 2.7 montrent que, dans tous les cas, $|\text{Bex}\left(\mathcal{B}, \overline{\boldsymbol{x}}\right) - x| = a\frac{M}{m}$ avec $0 < a < m$. Une telle quantité ne peut alors être un multiple de m_R. En effet, si nous supposons l'existence d'un entier $b \neq 0$ tel que $a\frac{M}{m} = bm_R$, alors cela implique que $\frac{m_R}{m_R \wedge \frac{M}{m}}$ divise a, et est donc strictement inférieur à m. L'hypothèse que $m_R \geqslant m \times \left(m_R \wedge \frac{M}{m}\right)$ est alors contredite, ce qui prouve la suffisance.

Nous prouvons la nécessité par contraposition pour le cas de Bex_{mrs}. Soit donc $m \in \mathcal{B}$ tel que $m_R < m \times \left(m_R \wedge \frac{M}{m}\right)$. Alors la Proposition 2.3 appliquée avec $x = 0$ montre qu'il existe une faute sur le résidu $|x|_m$ telle que

$$|\overline{x} - x| = \frac{m_R}{m_R \wedge \frac{M}{m}} \times \frac{M}{m} = m_R \times \frac{M}{m \times \left(m_R \wedge \frac{M}{m}\right)} \in m_R\mathbb{Z}.$$

Ce qui conclut la preuve. □

Remarque 2.5 *Vu la Proposition 2.4 concernant les fautes localisées affectant un coefficient MRS, la Condition (2.16) est suffisante pour détecter l'effet de toute faute sur un unique coefficient MRS lorsque* Bex_{mrs} *est utilisé.*

Les résultats fondamentaux concernant la détection de faute dans les RNS redondants sont désormais posés. Il s'agit par la suite d'étudier dans quelle mesure les RRNS sont utilisables pour protéger efficacement la multiplication modulaire RNS.

2.3 Vers une multiplication modulaire résistante aux fautes uniques

Disposer d'un algorithme de multiplication modulaire résistant aux attaques par canaux cachés et aux attaques par injection de faute est primordial en cryptographie. L'état-de-l'art de la cryptanalyse démontre que la multiplication modulaire est un point très sensible, puisqu'elle est l'opération cœur de cryptosystèmes comme le RSA ou le protocole d'échange de clés de Diffie-Hellman. Dans l'optique de faire du RNS un candidat de premier plan pour une arithmétique adaptée à la cryptographie et à ses contraintes, la question de fournir un algorithme de multiplication modulaire protégé efficacement contre les attaques par canaux auxiliaires est donc centrale.

Dans un premier temps, la création d'une arithmétique résistante aux fuites (Bajard et al. 2004) basée sur le RNS a permis de poser les premiers jalons de ce projet. La possibilité de choisir aléatoirement les bases RNS $\mathcal{B}$ et $\mathcal{B}'$ utilisées permet un masquage des données traitées via la représentation de Montgomery. Le point central est qu'il a été montré qu'il est possible de changer les bases RNS à la volée très efficacement puisque le calcul des nouvelles représentations de Montgomery qui découlent de ce nouveau choix de bases se réalise simplement par le biais de deux multiplications modulaires RNS consécutives (2.1) et (2.2). Ceci permet notamment de créer un algorithme d'exponentiation modulaire enrichi d'un masquage aléatoire et évolutif des données.

Dans un second temps, nous avons vu que les RNS redondants sont une solution efficace pour la protection contre les injections de fautes sur les résidus. Suivant le niveau de protection adopté et relié directement à la quantité de redondance introduite, la détection a cependant le désavantage d'avoir un coût non négligeable qui est celui de la conversion de base utilisée pour la procédure de détection. Dans un contexte d'optimisation des calculs cryptographiques, il est donc naturel de chercher à insérer au mieux les procédures de détection dans les schémas de calcul afin de limiter leur influence sur le temps total des calculs.

L'enjeu, résumé dans les deux objectifs suivants, est de trouver une solution complète et efficace de protection de l'exponentiation modulaire contre les attaques par canaux cachés et par injection de fautes.

Objectif 2.1 **Étudier dans quelle mesure les RNS redondants peuvent être utilisés pour protéger la multiplication modulaire RNS contre les fautes, et ceci de manière à ce que les procédures de détection aient le minimum d'impact en termes de temps et de surface.**

Objectif 2.2 **Trouver une solution de protection contre les attaques par injection de fautes compatible avec l'arithmétique résistante aux fuites LRA.**

2.3.1 Adéquation du modèle de faute unique pour la multiplication modulaire RNS

Dans le contexte de la multiplication modulaire, la Remarque 2.2 (p. 53) concernant la pertinence du modèle de faute localisée ne tient plus à cause de la présence des conversions de base. En revanche, l'Hypothèse 2.2 qui suit, dont la pertinence sera discutée, se révèlera suffisante pour le présent propos quant à la protection de la multiplication modulaire contre les fautes sur un résidu.

Hypothèse 2.2 *Soit* Bex *un opérateur de conversion de base et* $\boldsymbol{x}_{\mathcal{B}'} = \text{Bex}\,(\mathcal{B}, \mathcal{B}', \boldsymbol{x}_{\mathcal{B}})$. *Alors toute faute localisée injectée durant la conversion de base sur une valeur d'un canal RNS* $\mathbb{Z}/m_i\mathbb{Z}$ *l'est avant toute propagation de ladite valeur vers les canaux de la base de destination* $\mathcal{B}'$. *Dans ce cas, l'effet d'une telle faute peut être modélisé par une faute unique sur un résidu de* $\boldsymbol{x}_{\mathcal{B}}$.

Conversions de base basées sur le MRS

La pertinence de l'Hypothèse 2.2 dans le cas de Bex_{mrs} peut ne pas apparaître immédiatement. En effet, le calcul des coefficients MRS induit de nombreuses dépendances entre les canaux de la base RNS de départ $\mathcal{B}$.

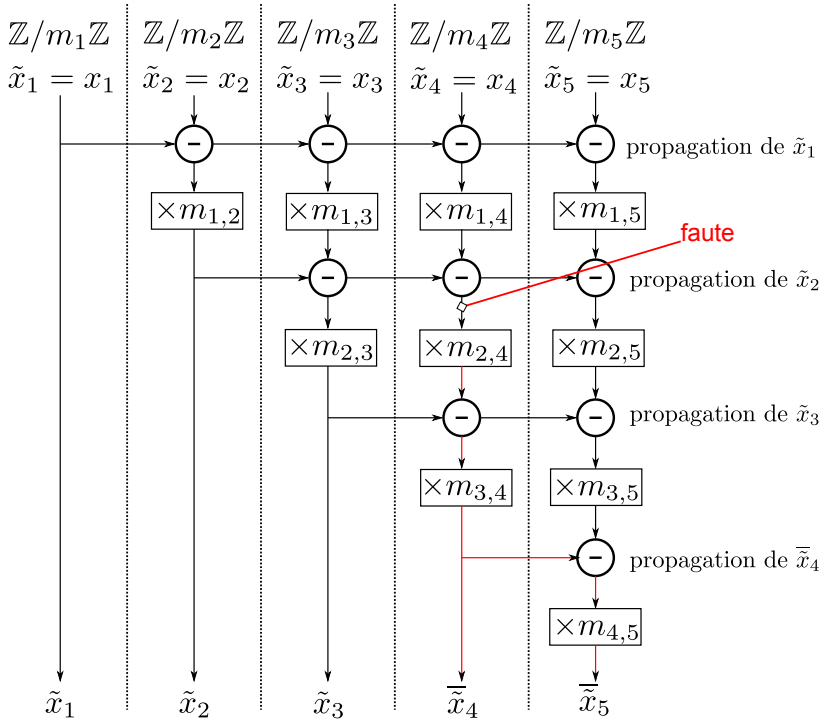


FIGURE 2.2 – *Injection d'une faute localisée lors du calcul de coefficients MRS, et propagation de la perturbation.*

La Figure 2.2 illustre la propagation d'une faute apparaissant lors d'un calcul de coefficient MRS dans le canal $\mathbb{Z}/m_4\mathbb{Z}$. La faute affecte alors les calculs des coefficients MRS consécutifs $\tilde{x}_5, \ldots, \tilde{x}_n$. Nous rappelons que la notation $m_{i,j}$ désigne l'inverse modulaire $\left|m_i^{-1}\right|_{m_j}$.

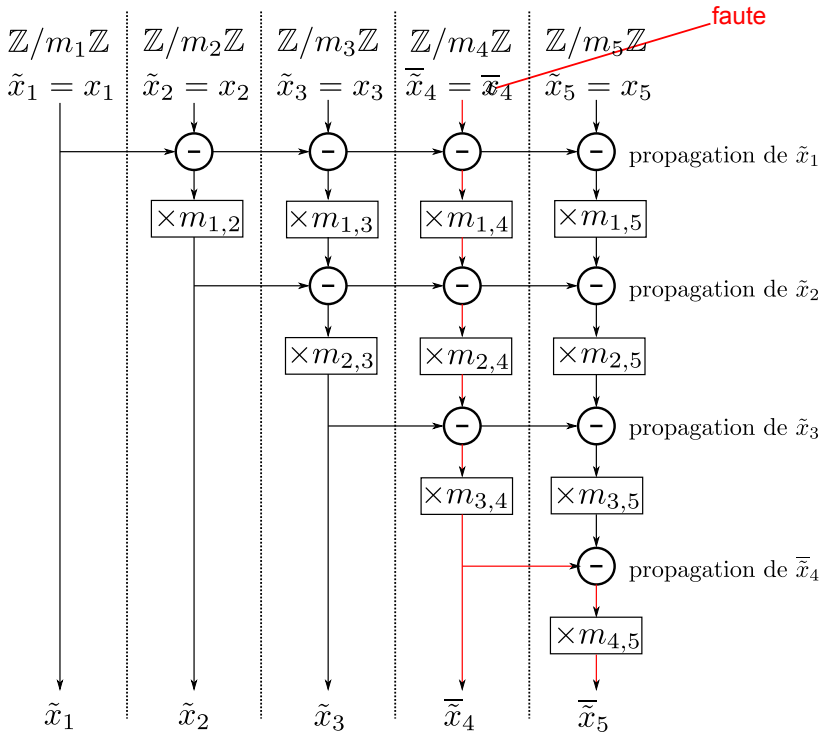


FIGURE 2.3 – *Propagation sur les coefficients MRS d'une perturbation causée par une faute unique sur un résidu dans la base de départ $\mathcal{B}$.*

La Figure 2.3 quant à elle montre comment une faute injectée sur le résidu x_4 avant le calcul des coefficients MRS propage de manière équivalente des perturbations sur les calculs des coefficients $\tilde{x}_4, \tilde{x}_5, \ldots, \tilde{x}_n$.

Dans le contexte de la Figure 2.2, l'erreur considérée apparaît après la propagation du coefficient $\tilde{x}_2$ vers les canaux RNS $\mathbb{Z}/m_i\mathbb{Z}$ pour $i \geqslant 3$. Ainsi, la valeur contenue dans le canal $\mathbb{Z}/m_4\mathbb{Z}$ après l'apparition d'une faute $e \in]\!]0, m_4[\!]$ a la forme suivante :

$$\left|\left(\left(x_4 - \tilde{x}_1\right) m_{1,4} - \tilde{x}_2\right) m_{2,4} + e\right|_{m_4}.$$

En continuant le calcul de $\tilde{x}_4$, puis de $\tilde{x}_5$, viennent les équations suivantes :

$$\begin{cases} \overline{\tilde{x}}_4 = \left|\tilde{x}_4 + e m_{3,4}\right|_{m_4} = \tilde{x}_4 + f \in [\![0, m_4[\![\\ \overline{\tilde{x}}_5 = \left|\tilde{x}_5 - f m_{4,5}\right|_{m_5}. \end{cases} \tag{2.17}$$

Les deux valeurs erronées $\overline{\tilde{x}}_4$ et $\overline{\tilde{x}}_5$ peuvent être obtenues à partir d'une faute sur le résidu x_4 avant la conversion de base. Notant $\overline{x}_4 = \left|x_4 + e'\right|_{m_4}$, le calcul des coefficients MRS donne en particulier :

$$\begin{cases} \overline{\tilde{x}}_4 = \left|\tilde{x}_4 + e' m_{1,4} m_{2,4} m_{3,4}\right|_{m_4} = \tilde{x}_4 + f' \in [\![0, m_4[\![, \\ \overline{\tilde{x}}_5 = \left|\tilde{x}_5 - f' m_{4,5}\right|_{m_5}. \end{cases} \tag{2.18}$$

Ainsi, la faute $e' = \left|e m_1 m_2\right|_{m_4}$ rend équivalents les deux systèmes d'équations (2.17) et (2.18).

Si l'Hypothèse 2.2 n'est pas vérifiée et que la valeur du coefficient MRS $\tilde{x}_4$ est modifiée après sa propagation vers le canal $\mathbb{Z}/m_5\mathbb{Z}$, nous obtenons alors $\overline{\tilde{x}}_4$ et $\tilde{x}_5$. Un tel cas de figure n'est pas toujours modélisable par une faute sur le

résidu x_4 injectée avant la conversion. En effet, une telle faute devrait donner $f' \equiv 0 \bmod m_5$ dans le système (2.18). Mais si $m_5 > m_4$, ce cas ne peut se présenter puisque par définition $f' \in]\!]0, m_4[\![$.

Dans les deux situations illustrées et discutées précédemment, une telle faute localisée affectera l'ensemble des résidus dans la base $\mathcal{B}'$. Mais le fait de pouvoir modéliser l'effet par une faute préconversion dans la base de départ se révèlera essentiel pour la détection de fautes lors d'une multiplication modulaire.

Conversions de base basées sur le TRC

Le cas des conversions de base construites sur l'Équation (1.15) est plus direct. L'Hypothèse 2.2 exprime seulement la nécessité que les valeurs $\xi_{x,i,\mathcal{B}}$ soient propagées équitablement vers les canaux de la base RNS de destination $\mathcal{B}'$. Durant une telle conversion, aucune dépendance entre les canaux de la base de départ $\mathcal{B}$ n'entre en jeu. Ainsi, toute faute localisée durant la conversion se résume à une faute unique pré- ou post-conversion.

Dans le cas d'une conversion Bex_{kwc}, le cas d'une faute modifiant la valeur du registre d'accumulation du Cox sera étudié, et n'est donc pas écarté pour le moment.

2.3.2 Catégories de fautes - Localisation

L'Hypothèse 2.2 permet d'introduire 4 catégories générales de fautes uniques pouvant apparaître lors d'une multiplication modulaire. Ces catégories sont illustrées par la Figure 2.4 et décrites ci-après.

Catégorie 1 Une faute est dite de catégorie 1 lorsque son effet est le même que celui d'une faute unique sur les résidus $\boldsymbol{q}_{\mathcal{B}}$ avant la procédure de conversion de base Bex_1. L'ensemble des fautes de catégorie 1 est celui des fautes localisées dans la base $\mathcal{B}$. C'est le cas par exemple d'une faute sur un résidu x_i de l'entrée $\boldsymbol{x}_{\mathcal{B}}$ de la multiplication modulaire, ou bien sur un résidu $\left|-p^{-1}\right|_{m_j}$ de la valeur précalculée $\left|-p^{-1}\right|_M$ ainsi que toute faute injectée durant la première conversion de base et pouvant se ramener à une faute sur $\boldsymbol{q}_{\mathcal{B}}$.

Catégorie 2 La catégorie 2 regroupe l'ensemble des fautes localisées dans la base auxiliaire $\mathcal{B}'$ et apparaissant en amont de la procédure Bex_2. Ce peut être le cas d'une faute sur un résidu précalculé ou issu des calculs intermédiaires menés entre les deux conversions de base Bex_1 et Bex_2. C'est aussi la catégorie des fautes localisées injectées durant Bex_2 et représentables par une faute pré-conversion.

Catégorie 3 La catégorie 3 est celle des fautes uniques sur les résidus $\boldsymbol{s}_{\mathcal{B}}$ lorsque ceux-ci ne sont pas destinés à être réutilisés comme entrée de l'algorithme de réduction modulaire redondant nouvellement proposé. Dans le cas contraire, une telle faute deviendra une faute de catégorie 1. Le but de la création de cette catégorie est de décrire comment assurer l'intégrité de l'ensemble des résidus d'une sortie de la réduction modulaire.

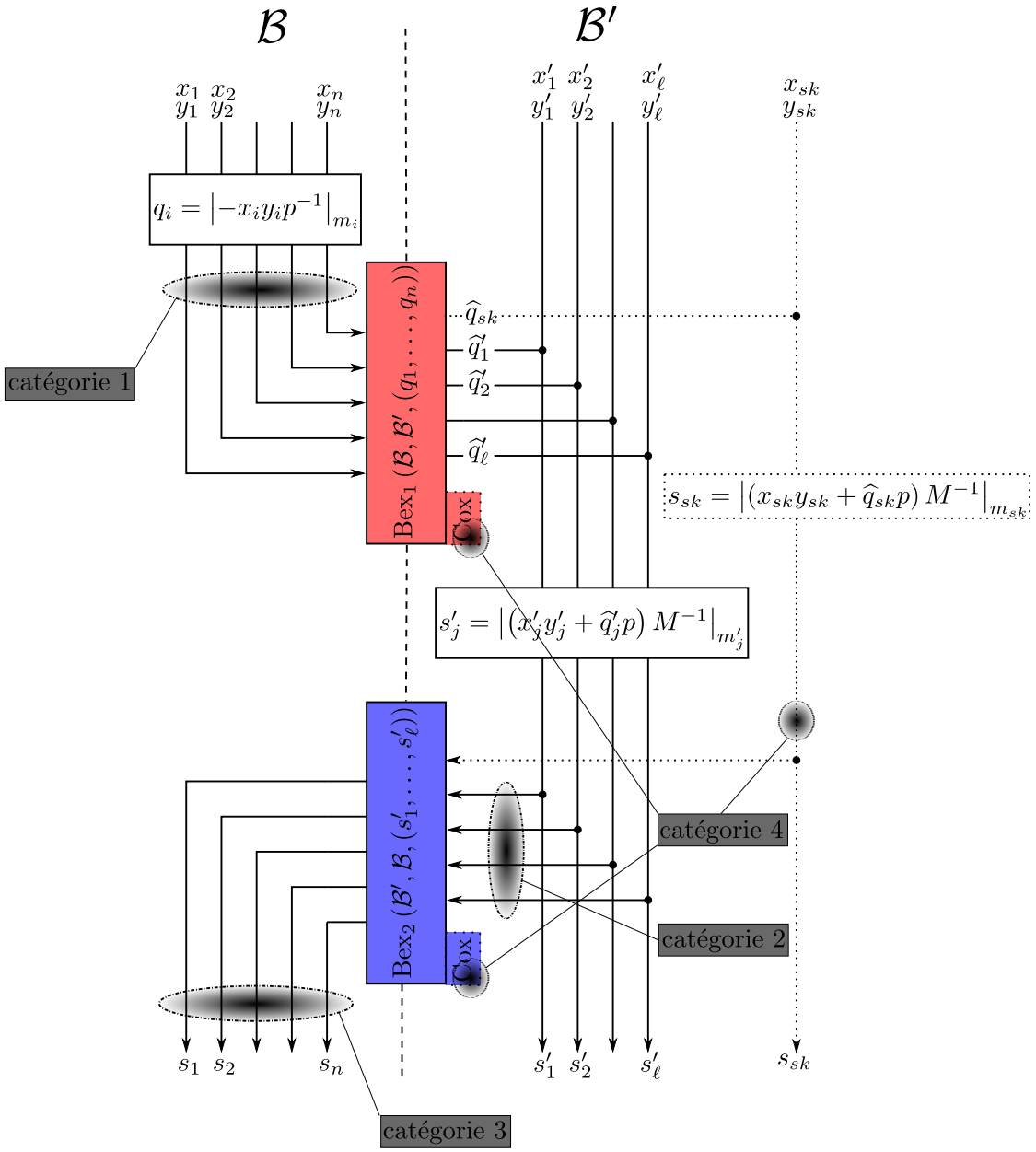


FIGURE 2.4 – *Catégorisation des fautes uniques durant une multiplication modulaire avec indication du matériel supplémentaire (Cox, canal $\mathbb{Z}/m_{sk}\mathbb{Z}$) selon les conversions de base utilisées.*

Catégorie 4 La catégorie 4 concerne les fautes localisées dans le canal $\mathbb{Z}/m_{sk}\mathbb{Z}$ lorsque $\mathrm{Bex}_2 = \mathrm{Bex}_{sk}$, ou sur l'unité Cox lorsque les conversions utilisées sont celles de Kawamura et al. (2000).

2.3.3 Algorithme de réduction modulaire RNS avec capacité de détection de faute unique

La création d'un algorithme de réduction modulaire RNS intégrant un canal RNS redondant permettant la détection des fautes de catégorie 1, 2, 3 et 4 est motivée par l'Objectif 2.1. Une question fondamentale concernant l'efficacité de l'intégration d'une procédure de détection dans l'Algorithme 8 de réduction modulaire RNS apparaît naturellement et est formulée ci-après.

Question 2.1 **La procédure de détection d'erreur étant basée sur une conversion de base qui est une opération coûteuse, comment et dans quelle mesure est-il possible d'exploiter les deux conversions Bex_1 et Bex_2 pour mettre en œuvre la procédure `DetectOneErr` afin de détecter les fautes des catégories 1, 2, 3 et 4 ?**

Une idée naturelle est de chercher à utiliser Bex_1 pour détecter les fautes de catégorie 1, et Bex_2 pour détecter les fautes de catégorie 2 et 4. Mais un problème surgit rapidement.

De l'impossible détection certaine des fautes de catégorie 1 par Bex_1

En supposant disposer d'un canal redondant $\mathbb{Z}/m_R\mathbb{Z}$, la détection des fautes de catégorie 1 requiert de pouvoir calculer le résidu redondant q_R de q. Mais de par la définition même de q, nous ne connaissons pas de moyen simple pour obtenir ce résidu redondant uniquement à partir de calculs dans le seul canal $\mathbb{Z}/m_R\mathbb{Z}$. Même en supposant donnés les résidus x_R et $\left|-p^{-1}\right|_{m_R}$, alors nous ne savons obtenir $q_R = q \bmod m_R = \left|-xp^{-1}\right|_M \bmod m_R$ sans procéder à une conversion de base. Ceci s'explique par la présence de la réduction modulo M apparaissant dans la construction de q. L'équivalence suivante souligne ce problème de ne pouvoir obtenir q_R avec certitude à partir de la seule donnée des résidus de x et p dans $\mathcal{B} \cup \{m_R\}$:

$$\left|-xp^{-1}\right|_M \bmod m_R = \left|-xp^{-1}\right|_{m_R} \Leftrightarrow \left|-xp^{-1}\right|_{Mm_R} \in [\![0, M[\![. \tag{2.19}$$

Par conséquent, Bex_1 ne peut être utilisée seule pour mettre en œuvre une procédure de détection des fautes de catégorie 1. Cela s'avère problématique, car une faute de catégorie 1 non détectée avant la première conversion de base va affecter tous les résidus $\boldsymbol{q}_{\mathcal{B}'}$, et par conséquent tous les résidus $s'_1, \ldots, s'_\ell$.

Algorithme proposé, preuve de correction

L'Algorithme 13 de multiplication modulaire en RNS redondant prend en considération les constatations précédentes. Le principe de l'algorithme est illustré par la Figure 2.5. Seule la seconde conversion de base est exploitée dans le cadre d'une procédure de détection de faute. Le Théorème 2.4 prouve en fait que la solution proposée permet notamment la détection des fautes catégorie 1, 2 et 4 définies précédemment, ainsi que les fautes de catégories 5, qui sont par définition celles affectant une valeur du canal redondant $\mathbb{Z}/m_R\mathbb{Z}$. Le traitement du cas des fautes de catégorie 3 sera discuté par la suite.

Théorème 2.4 *Soit $\mathcal{H}$ un ensemble d'hypothèses de la Table 1.1 (p. 34) et m_R un modulus premier à $\mathcal{B}$ et $\mathcal{B}'$ et vérifiant la condition suivante :*

$$\forall m \in \mathcal{B}, \forall m' \in \mathcal{B}',\ m_R > \begin{cases} m, m' \text{ si } \mathcal{H} \in \{\mathcal{H}_{mrs}, \mathcal{H}_{kw}\}, \\ m_{sk} m, m' \text{ si } \mathcal{H} = \mathcal{H}_{sk}. \end{cases} \tag{2.20}$$

Alors l'Algorithme 13 détecte toutes les fautes de catégorie 1, 2 et 5. De plus, la détection des fautes de catégorie 4 est aussi assurée pour $\mathcal{H} = \mathcal{H}_{sk}$.

Démonstration. **Catégorie 2** L'enjeu pour la détection des fautes de catégorie 2 est de pouvoir calculer le résidu $s \bmod m_R$. Par définition, s est défini par

Algorithme 13 : MulModRRNS $(\mathcal{B}, \mathcal{B}', m_R, x, y, p)$:

Données :

- trois bases copremières $\mathcal{B} = \{m_1, \ldots, m_n\}$, $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$ et $\{m_R\}$
- deux entiers x, y représentés par leurs résidus $\boldsymbol{x}_{\mathcal{B}\cup\mathcal{B}'\cup\{m_R\}}$ et $\boldsymbol{y}_{\mathcal{B}\cup\mathcal{B}'\cup\{m_R\}}$ dans les trois bases
- un modulus p représenté par ses résidus précalculés $\boldsymbol{p}_{\mathcal{B}'\cup\{m_R\}}$ dans la base $\mathcal{B}' \cup \{m_R\}$
- les résidus précalculés de l'entier $\left|-p^{-1}\right|_M$ dans $\mathcal{B}$ et de $\left|M^{-1}\right|_{M'm_R}$ dans $\mathcal{B}' \cup \{m_R\}$
- toutes ces données vérifient également les hypothèses de la Table 1.1 (p. 34).

Résultat : les résidus de $s \equiv xyM^{-1} \bmod p$ dans $\mathcal{B} \cup \mathcal{B}' \cup \{m_R\}$

début
  pour $i \leftarrow 1$ **à** n **faire**
    `/* en // dans les canaux de` $\mathcal{B}$ `*/`
    $q_i \leftarrow x_i \times y_i \times \left|-p^{-1}\right|_{m_i} \bmod m_i$
  $\left(\widehat{q}'_1, \ldots, \widehat{q}'_\ell, \widehat{q}_R\right) \leftarrow \mathrm{Bex}_1\left(\mathcal{B}, \mathcal{B}' \cup \{m_R\}, (q_1, \ldots, q_n)\right)$
    `/* conversion de` $\mathcal{B}$ `vers` $\mathcal{B}' \cup \{m_R\}$ `*/`
  pour $j \leftarrow 1$ **à** ℓ **faire**
    $t'_j \leftarrow x'_j \times y'_j + \widehat{q}'_j \times p'_j \bmod m'_j$
  $t_R \leftarrow x_R \times y_R + \widehat{q}_R \times p_R \bmod m_R$
    `/* en // dans les canaux de` $\mathcal{B}' \cup \{m_R\}$ `*/`
  pour $j \leftarrow 1$ **à** ℓ **faire**
    $s'_j \leftarrow t'_j \times \left|M^{-1}\right|_{m'_j} \bmod m'_j$
  $s_R \leftarrow t_R \times \left|M^{-1}\right|_{m_R} \bmod m_R$
    `/* en // dans les canaux de` $\mathcal{B}' \cup \{m_R\}$ `*/`
  $(s_1, \ldots, s_n, \widehat{s}_R) \leftarrow \mathrm{Bex}_2\left(\mathcal{B}', \mathcal{B} \cup \{m_R\}, \left(s'_1, \ldots, s'_\ell\right)\right)$
    `/* conversion de` $\mathcal{B}'$ `vers` $\mathcal{B} \cup \{m_R\}$ `*/`
  si $s_R \neq \widehat{s}_R$ **alors**
    `/* procédure de détection */`
    retourner `Faute détectée.`
  sinon
    retourner $\left(s_1, \ldots, s_n, s'_1, \ldots, s'_\ell, s_R\right)$

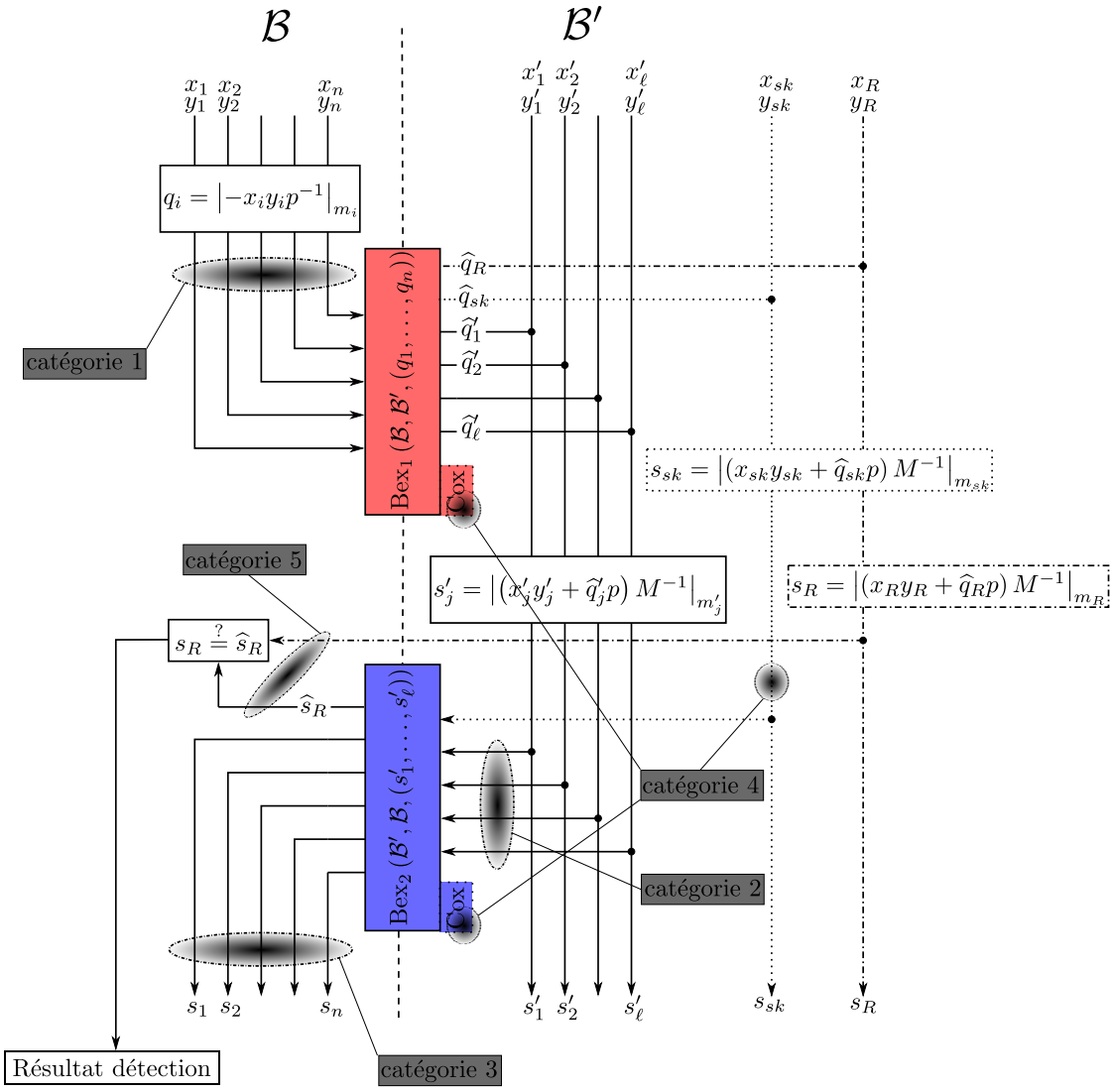


FIGURE 2.5 – *Multiplication modulaire en RNS redondant.*

les résidus dans $\mathcal{B}'$ de la quantité $\left|tM^{-1}\right|_{M'}$, où $t = xy + p\widehat{q}$. $\widehat{q}$ désigne la valeur obtenue par $\text{Bex}_1(\mathcal{B}, \boldsymbol{q}_\mathcal{B})$. Par construction de q et donc de $\widehat{q}$, t est un multiple de M : $xy + p\widehat{q} \equiv 0 \bmod M$. Ainsi, $\left|(xy + p\widehat{q})\, M^{-1}\right|_{M'}$ est une division exacte réalisée en RNS. Ensuite, la base auxiliaire $\mathcal{B}'$ est toujours choisie de manière à ce qu'un dépassement de capacité de t sur la base étendue $\mathcal{B} \cup \mathcal{B}'$ soit impossible. Autrement dit, $\frac{t}{M} < M'$. Par conséquent, $\varphi_{\mathcal{B}'}^{-1}(\boldsymbol{s}_{\mathcal{B}'}) = \frac{t}{M}$. L'équation suivante vient alors immédiatement :

$$s_R = \varphi_{\mathcal{B}'}^{-1}(\boldsymbol{s}_{\mathcal{B}'}) \bmod m_R = \left|tM^{-1}\right|_{M'} \bmod m_R = \frac{xy + p\widehat{q}}{M} \bmod m_R. \quad (2.21)$$

L'Équation (2.21) fournit le calcul du résidu redondant s_R de s avant la conversion de base Bex_2. Ce calcul nécessite de connaître les résidus redondants x_R et y_R des entrées x et y, ainsi que ceux des entiers p et $\left|M^{-1}\right|_{m_R}$. De plus, la conversion de $\boldsymbol{q}_\mathcal{B}$ par Bex_1 doit se faire de la base $\mathcal{B}$ vers la base étendue $\mathcal{B}' \cup \{m_R\}$. Dans ce cas, comme Bex_2 est complète lorsqu'elle est appliquée sur des résidus intègres, il devient possible de l'utiliser pour détecter les fautes de catégorie 2.

Finalement, si m_R vérifie la Condition (2.20), le Théorème 2.3 (p. 60) garantit la détection de toutes les fautes de catégorie 2.

Catégorie 4 La preuve pour la détection des faute sur le canal $\mathbb{Z}/m_{sk}\mathbb{Z}$ est identique à celle de la catégorie 2 et utilise le Théorème 2.3.

Catégorie 1 Par définition du test de cohérence, toute faute de catégorie 1 est détectée si, et seulement si, $|s_R - \widehat{s}_R|_{m_R} \neq 0$. De manière équivalente, il faut et il suffit que l'inégalité $|(s_R - \widehat{s}_R) M|_{m_R} \neq 0$ soit vérifiée dès qu'une faute de catégorie 1 apparaît, puisque M est premier avec m_R.

Sans perte de généralité, nous supposons par exemple que le résidu q_1 est erroné. Sa nouvelle valeur est notée $\overline{q}_1 = |q_1 + e_1|_{m_1}$. Ces résidus erronés définissent l'entier $\overline{q} = \varphi_{\mathcal{B}}^{-1}\left(\left(\overline{q}_1, q_2, \ldots, q_n\right)\right)$, et la valeur retournée par la première conversion est notée $\widehat{\overline{q}} = \mathrm{Bex}_1\left(\mathcal{B}, \overline{\boldsymbol{q}}_{\mathcal{B}}\right)$. Il faut remarquer que malgré la présence de la faute, le calcul de t ne provoque aucun dépassement de capacité :

$$t = xy + \widehat{\overline{q}}p \in [\![0, MM'[\![.$$

Cependant, t n'est désormais plus un multiple de M, puisque

$$\varphi_{\mathcal{B}}(t) = (e_1 p, 0, \ldots, 0)_{\mathcal{B}}.$$

Par conséquent, le résidu redondant de s a pour nouvelle valeur :

$$s_R = \left|tM^{-1}\right|_{m_R} = \left|\varphi_{\mathcal{B}\cup\mathcal{B}'}^{-1}\left(\left((e_1 p, 0, \ldots, 0)_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'}\right)\right) M^{-1}\right|_{m_R}. \tag{2.22}$$

Ensuite, l'effet de la faute modifie également les résidus de s dans $\mathcal{B}'$. Par construction, ces résidus sont précisément $\boldsymbol{s}_{\mathcal{B}'} = \varphi_{\mathcal{B}'}\left(\left|tM^{-1}\right|_{M'}\right)$. Nous pouvons alors écrire :

$$\varphi_{\mathcal{B}'}^{-1}\left(\boldsymbol{s}_{\mathcal{B}'}\right) = \frac{\varphi_{\mathcal{B}\cup\mathcal{B}'}^{-1}\left((\boldsymbol{0}_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'})\right)}{M}. \tag{2.23}$$

De ce fait, si Bex_2 est choisie parmi Bex_{sk} ou Bex_{kwc}, il est possible que la conversion ne soit plus complète et renvoie une valeur de la forme suivante :

$$\mathrm{Bex}_2\left(\mathcal{B}', \boldsymbol{s}_{\mathcal{B}'}\right) = \varphi_{\mathcal{B}'}^{-1}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + \mu M' \tag{2.24}$$

où les valeurs possibles de l'entier μ sont détaillées par la suite.

Finalement, la procédure de détection teste la nullité de

$$\left|\varphi_{\mathcal{B}\cup\mathcal{B}'}^{-1}\left(\left((e_1 p, 0, \ldots, 0)_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'}\right)\right) M^{-1} - \varphi_{\mathcal{B}'}^{-1}\left(\boldsymbol{s}_{\mathcal{B}'}\right) - \mu M'\right|_{m_R} \stackrel{?}{=} 0.$$

Comme précisé précédemment, le test de détection est équivalent au test suivant :

$$\varphi_{\mathcal{B}\cup\mathcal{B}'}^{-1}\left(\left((e_1 p, 0, \ldots, 0)_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'}\right)\right) - \varphi_{\mathcal{B}\cup\mathcal{B}'}^{-1}\left((\boldsymbol{0}_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'})\right) - \mu MM' \stackrel{?}{\equiv} 0 \bmod m_R. \tag{2.25}$$

Un entier $\delta \in \{0,1\}$ est introduit et est défini comme l'entier rendant vraie l'égalité suivante :

$$\begin{aligned}\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\left((e_1 p,0,\ldots,0)_{\mathcal{B}},\boldsymbol{t}_{\mathcal{B}'}\right)\right) &= \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\left((e_1 p,0,\ldots,0)_{\mathcal{B}},\mathbf{0}_{\mathcal{B}'}\right)\right)\\ &\quad+\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left((\mathbf{0}_{\mathcal{B}},\boldsymbol{t}_{\mathcal{B}'}\right)) - \delta MM'.\end{aligned}$$

Par suite, l'Équation (2.25) se réécrit :

$$\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\left((e_1 p,0,\ldots,0)_{\mathcal{B}},\mathbf{0}_{\mathcal{B}'}\right)\right) - (\mu+\delta)\,MM' \stackrel{?}{\equiv} 0 \bmod m_R. \tag{2.26}$$

$\mathcal{B}\cup\mathcal{B}'$ étant une base RNS, la Proposition 2.1 (p. 53), concernant la forme de l'entier représenté par des résidus atteints d'une faute unique, donne l'existence d'un entier $a_1 \in]\!]-m_1, m_1[\![\setminus\{0\}$ tel que :

$$\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\left((e_1 p,0,\ldots,0)_{\mathcal{B}},\mathbf{0}_{\mathcal{B}'}\right)\right) = a_1 M_1 M'.$$

Plus précisément, $a_1 > 0$ et nous avons même plus précisément $a_1 = \left|e_1 p\left(M_1 M'\right)^{-1}\right|_{m_1}$. Comme $M_1 M'$ est premier avec m_R, il suffit de s'assurer que sous les conditions du Théorème 2.4 l'assertion suivante est vraie :

$$\left|e_1 p\left(M_1 M'\right)^{-1}\right|_{m_1} - (\mu+\delta)\,m_1 \neq 0 \bmod m_R. \tag{2.27}$$

Pour ce faire, les valeurs possibles de $\mu+\delta$ suivant la conversion de base choisie sont maintenant détaillées.

Bex_{mrs} **:** La conversion est toujours complète, ce qui signifie que $\mu = 0$. Ainsi, $0 < \left|\left|e_1 p\left(M_1 M'\right)^{-1}\right|_{m_1} - \delta m_1\right| < m_1 < m_R$, et le test détecte la faute e_1.

Bex_{sk} **:** Le contexte est différent de celui du Théorème 2.3 concernant la détection d'une faute en utilisant Bex_{sk}. En effet, ici tous les résidus de $\boldsymbol{s}_{\mathcal{B}'}$ sont potentiellement modifiés par la faute e_1. Donc la conversion renvoie une valeur de la forme $\varphi^{-1}_{\mathcal{B}'}(\boldsymbol{s}_{\mathcal{B}'}) + \mu M'$ où μ est une valeur de $[\![-\kappa_{\mathcal{B}'}(\boldsymbol{s}_{\mathcal{B}'}), m_{sk}-1-\kappa_{\mathcal{B}'}(\boldsymbol{s}_{\mathcal{B}'})]\!]$. De ce fait, seule l'inéquation suivante reste garantie :

$$0 < \left|\left|e_1 p\left(M_1 M'\right)^{-1}\right|_{m_1} - (\delta+\mu)\,m_1\right| \leqslant m_1 - 1 - (m_{sk}-1)\,m_1 = m_{sk} m_1 - 1.$$

L'Hypothèse (2.20) du présent théorème donnant $m_R > m_{sk} m_1$ garantit donc bien la détection de la faute e_1.

Bex_{kwc} **:** À cause du coefficient de correction α_{kw} utilisé pour garantir une conversion complète pour les valeurs intègres s vérifiant $s < (1-\alpha_{kw})\,M'$, μ peut prendre pour seules valeurs 0 ou -1. De plus nous avons l'implication $\delta = 0 \Rightarrow \mu = 0$. En effet, si $\delta = 0$ alors :

$$\begin{aligned}\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\left((e_1 p,0,\ldots,0)_{\mathcal{B}},\boldsymbol{t}_{\mathcal{B}'}\right)\right) &= \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\left((e_1 p,0,\ldots,0)_{\mathcal{B}},\mathbf{0}_{\mathcal{B}'}\right)\right)\\ &\quad+\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left((\mathbf{0}_{\mathcal{B}},\boldsymbol{t}_{\mathcal{B}'}\right))\\ &= t.\end{aligned}$$

De plus, vu la Remarque 1.8 (p. 29), $\widehat{\widehat{q}} < (1+\Delta_{kw})\,M$. Ainsi, les hypothèses $\mathcal{H}_{kw}$ garantissent alors que :

$$t = xy + \widehat{\widehat{q}}p < (1-\Delta_{kw})\,Mp + (1+\Delta_{kw})\,Mp = 2Mp < (1-\alpha_{kw})\,MM'.$$

Par conséquent,

$$\varphi_{\mathcal{B}'}^{-1}\left(\boldsymbol{s}_{\mathcal{B}'}\right) = \frac{\varphi_{\mathcal{B}\cup\mathcal{B}'}^{-1}\left(\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'}\right)\right)}{M} < (1-\alpha_{kw})\,M'.$$

Cette inégalité implique dans ce cas une conversion complète, c'est-à-dire $\mu = 0$. Ainsi, $\delta + \mu \in \{0,1\}$.
Finalement, il vient :

$$0 < \left| \left| e_1 p \left(M_1 M'\right)^{-1} \right|_{m_1} - (\delta+\mu)\,m_1 \right| < m_1 < m_R,$$

inégalités prouvant que la détection réussit.

Catégorie 5 La modification par injection de faute d'une des deux valeurs s_R ou $\widehat{s}_R$ dans le canal $\mathbb{Z}/m_R\mathbb{Z}$ implique trivialement la détection.

□

Fautes de catégorie 4 sur le Cox

La protection contre les fautes de catégorie 4 correspondant à une perturbation de la valeur calculée par le Cox peut suivre différentes stratégies, comme par exemple une redondance de cette unité de traitement. S'agissant d'un simple additionneur-accumulateur, le coût reste négligeable. Si un seul Cox est utilisé, une altération non permanente (cf. 2.4.1 concernant les fautes permanentes et transitoires) la valeur contenue dans le registre d'accumulation lors de la première conversion Bex_{kw} peut provoquer l'apparition d'un terme $\pm M$ dans le résultat de la conversion, via la possible apparition d'une retenue sortante supplémentaire ou de la disparition d'une retenue initiale. Ainsi, nous obtenons un résultat de la forme $\widehat{q} = q + \delta M$ où $\delta \in \{-1,0,1,2\}$. Les nouveaux cas de figure sont ceux avec $\delta \in \{-1,2\}$. Il est à noter que $\widehat{q}$ satisfait la condition $-M \leqslant \widehat{q} < (2+\Delta_{kw})\,M$, puisqu'en l'absence de faute $\mathrm{Bex}_{kw}\left(\mathcal{B}, \boldsymbol{q}_{\mathcal{B}}\right)$ est dans $[\![0, (1+\Delta_{kw})\,M[\![$ (cf. Remarque 1.8, p. 29).

Un premier problème est lié à la possible négativité de $\widehat{q}$. Ce cas peut provoquer un dépassement de capacité de $t = xy + \widehat{q}p$ qui peut ne plus appartenir à l'intervalle dynamique $[\![0, MM'[\![$ du système RNS total $\mathcal{B}\cup\mathcal{B}'$. En effet, t vérifie les inégalités suivantes :

$$-Mp \leqslant t = xy + \widehat{q}p < (1-\Delta_{kw})\,Mp + (2+\Delta_{kw})\,Mp = 3Mp.$$

Il est à noter cependant que t vérifie toujours $t \equiv xy \bmod p$. De ce fait, une faute sur le Cox ne modifie pas la congruence modulo p du résultat, mais des précautions doivent être prises soit pour détecter un dépassement de capacité et donc détecter la faute, soit pour contrecarrer l'effet d'une telle faute via une correction automatique si possible.

Si $t \in [\![-Mp, 0[\![$, alors $s \bmod M' = \left|tM^{-1}\right|_{M'} = \frac{t}{M} + M'$ et $s_R = s \bmod m_R = \left|tM^{-1}\right|_{m_R}$. Dans ce cas, comme $s \bmod M'$ n'a plus la garantie d'appartenir à

l'intervalle $[\![0, (1-\alpha_{kw})\, M'[\![$ pour lesquelles la seconde conversion Bex_{kw} est toujours complète, il devient possible d'obtenir :

$$\text{Bex}_{kw}\left(\mathcal{B}', \boldsymbol{s}_{\mathcal{B}'}\right) = \frac{t}{M} + M' - \delta M'$$

où δ peut prendre pour valeur 1. Dans ce cas, la faute est corrigée par la seconde conversion et le test de détection réussit puisque celui-ci n'est que la comparaison de la valeur $\frac{t}{M} + M' - \delta M' \bmod m_R = \frac{t}{M} \bmod m_R$ obtenue via la conversion à la valeur $s_R = \left|tM^{-1}\right|_{m_R}$. Le résultat obtenu au final est l'ensemble des résidus dans $\mathcal{B} \cup \mathcal{B}' \cup \{m_R\}$ de $\left|xyM^{-1}\right|_p - p$. Il est donc possible de décider d'ajouter systématiquement $+p$ à la sortie de la multiplication modulaire. Or, comme il a été souligné précédemment, $t < 3Mp$. Au final, le résultat s obtenu, s'il reste congru à $\left|xyM^{-1}\right|_p$ modulo p, appartient à l'intervalle $[\![0, 4p[\![$. Il suffit de modifier dans les hypothèses $\mathcal{H}_{kw}$ la taille de M' via la nouvelle condition $\frac{4p}{1-\alpha_{kw}} < M'$, et de choisir $\sigma \geqslant 16$.

Si la faute sur le Cox apparaît durant la seconde conversion Bex_{kwc}, il devient possible d'obtenir la valeur suivante :

$$\text{Bex}_{kw}\left(\mathcal{B}', \boldsymbol{s}_{\mathcal{B}'}\right) = \frac{t}{M} + \delta M',$$

avec $\delta \in \{-1, 0, 1\}$. Comme M' est premier avec m_R, le test de détection donnera :

$$\left|s_R - tM^{-1} + \delta M'\right|_{m_R} = \left|\delta M'\right|_{m_R} \neq 0 \Leftrightarrow \delta \neq 0.$$

Donc, si la faute modifie effectivement le résultat de la conversion, elle sera détectée, sinon le résultat obtenu reste correct.

Finalement, nous venons de montrer la Proposition 2.8 suivante.

Proposition 2.8 *Soit les hypothèses $\mathcal{H}_{kw}$ modifiées de manière à ce que $4p < (1-\alpha_{kw})M'$ et $\sigma \geqslant 16$. Alors l'Algorithme 13, utilisé avec un seul Cox et modifié de manière à ce que $+p$ soit automatiquement rajouté aux résidus de la sortie s dans la base totale $\mathcal{B} \cup \mathcal{B}' \cup \{m_R\}$, a la capacité de prévenir l'effet de toute faute localisée sur le Cox injectée durant* Bex_1*, et celle de détecter toute faute similaire injectée durant* Bex_2 *si elle rend cette conversion incomplètement réduite. Le résultat $s \equiv xyM^{-1} \bmod p$ obtenu est alors dans l'intervalle $[\![0, 4p[\![$.*

De plus, la capacité de détection des fautes de catégorie 1, 2 et 5 reste inchangée dès lors que m_R vérifie la Condition (2.20) du théorème précédent.

Une stratégie différente suggérée par (Nozaki et al. 2001) consiste à allouer un Cox par Rower. Dans ce cas, une faute localisée n'atteint par définition qu'un seul Cox. Lors de la première conversion, si le Rower contenant le Cox atteint est celui affecté au canal $\mathbb{Z}/m_R\mathbb{Z}$, cela revient à considérer le résidu $\overline{\hat{q}}_R = \hat{q}_R + \delta_R M$ avec $\delta_R \in \{-1, 0, 1\}$. Ainsi, le test de cohérence compare $\widehat{s}_R$ (obtenu de $\boldsymbol{s}_{\mathcal{B}'}$ par la seconde conversion) à la valeur perturbée $\overline{s}_R = s_R + \delta_R p = \widehat{s}_R + \delta_R p$. Une telle faute de catégorie 5 est détectée dès que $\delta_R \neq 0$. Si le Rower est affecté à un canal $\mathbb{Z}/m'_j\mathbb{Z}$, alors la faute sur le Cox provoque la modification du résidu s'_j en $\overline{s}'_j = \left|s'_j + \delta_j M\right|_{m'_j} \neq s'_j$. C'est exactement le cas d'une faute de catégorie 2, qui sera donc détectée. Une telle faute sur un Cox durant la seconde conversion provoque, de la même manière, soit l'apparition d'une faute de catégorie 5, soit celle d'une faute de catégorie 3.

Finalement, aucune modification des hypothèses $\mathcal{H}_{kw}$ n'est nécessaire pour la détection des fautes sur un Cox lorsque chaque Rower possède le sien.

Gestion des fautes de catégorie 3

Sous l'hypothèse qu'une faute localisée est injectée durant la seconde conversion Bex_2, avec pour effet de modifier un résidu de $\boldsymbol{s}_{\mathcal{B}}$, la procédure de détection de l'Algorithme 13 garantit seulement l'intégrité des résidus de la sortie s dans la base $\mathcal{B}' \cup \{m_R\}$. Si cette sortie est destinée à être convertie en binaire, il suffit d'utiliser les résidus $\boldsymbol{s}_{\mathcal{B}'}$. En effet, $s < M'$ et donc $s = \varphi_{\mathcal{B}'}^{-1}(\boldsymbol{s}_{\mathcal{B}'}) = \mathrm{Bex}_2(\mathcal{B}', \boldsymbol{s}_{\mathcal{B}'})$. Dans le cas où il est nécessaire de vérifier l'intégrité des résidus $\boldsymbol{s}_{\mathcal{B}}$, il est à noter que le résidu s_R est disponible et n'est, par hypothèse, pas affecté par une faute (le test de cohérence de l'Algorithme 13 ayant réussi). Il suffit alors d'étudier dans quelle mesure il reste possible d'utiliser une procédure de détection suivant la conversion utilisée.

Bex_{mrs} **:** La procédure de conversion peut être utilisée sur $(\boldsymbol{s}_{\mathcal{B}}, s_R)$ sans condition particulière.

Bex_{sk} **:** Disposant du résidu s_{sk} non affecté par une faute par hypothèse (de la même manière que s_R), les conditions du Théorème 2.3 sont réunies. La procédure de détection est donc possible.

Bex_{kwc} **:** Il est nécessaire de pouvoir garantir que la conversion utilisée est complètement réduite lorsqu'aucune faute n'affecte un résidu de $\boldsymbol{s}_{\mathcal{B}}$ de manière à ce qu'on ne puisse pas avoir $\mathrm{Bex}_{kwc}(\mathcal{B}, \{m_R\}, \boldsymbol{s}_{\mathcal{B}}, \alpha_{kw}) = |s_R - M|_{m_R}$ en l'absence de faute, ce qui rendrait sinon caduque la procédure de détection de faute.

Si $\boldsymbol{s}_{\mathcal{B}}$ est intègre, alors par hypothèse $\varphi_{\mathcal{B}}^{-1}(\boldsymbol{s}_{\mathcal{B}}) = s < 2p$. Par conséquent, il suffit d'introduire la condition suivante :

$$2p < (1 - \alpha_{kw})\, M. \tag{2.28}$$

Dans ce cas, $\mathrm{Bex}_{kwc}(\mathcal{B}, \boldsymbol{s}_{\mathcal{B}}, \alpha_{kw}) = \varphi_{\mathcal{B}}^{-1}(\boldsymbol{s}_{\mathcal{B}}) = s$. En pratique, le choix des paramètres n et $r = \lceil \log_2(m_i) \rceil$ permet d'avoir $\Delta_{kw} \leqslant \alpha_{kw} \leqslant \frac{1}{2}$ en conservant un paramètre h petit (cf. l'Équation (1.23) (p. 27) qui exhibe un majorant de Δ_{kw}). Or par hypothèse $\frac{4p}{1-\Delta_{kw}} < M$. De ce fait, s'il est possible d'avoir $\alpha_{kw} \leqslant \frac{1}{2}$, alors $\frac{2p}{1-\alpha_{kw}} \leqslant \frac{4p}{1-\Delta_{kw}}$. Cette condition permet ainsi la détection des fautes de catégorie 3 sur $\boldsymbol{s}_{\mathcal{B}}$, et ceci sans modifier l'hypothèse sur la taille de M dans $\mathcal{H}_{kw}$, en procédant au test d'égalité suivant :

$$\mathrm{Bex}_{kwc}(\mathcal{B}, \boldsymbol{s}_{\mathcal{B}}, \{m_R\}, \alpha_{kw}) - s_R \stackrel{?}{=} 0 \bmod m_R.$$

Surcoût dû à la procédure de détection

L'introduction du canal redondant $\mathbb{Z}/m_R\mathbb{Z}$ induit un surcoût en terme d'opérations basiques `MME1` et `AME1`, ainsi qu'une opération de comparaison entre résidus redondants qui ne consiste finalement qu'en un `XOR` bit à bit. Comme m_R peut être choisi de manière à avoir une taille identique à celle des moduli de $\mathcal{B}$ et $\mathcal{B}'$, ce surcoût correspond donc à celui d'un canal RNS non redondant.

hypothèses	$\mathcal{H}_{mrs}$	$\mathcal{H}_{sk}$	$\mathcal{H}_{kw}$
Mult.	$(n+\ell+1)$ MME1	$(n+\ell+4)$ MME1	$(n+\ell+5)$ MME1
Add.	$(n+\ell-1)$ AME1	$(n+\ell)$ AME1	$(n+\ell)$ AME1

TABLE 2.1 – *Surcoût de la procédure de détection des fautes uniques de catégorie 1, 2 et 4 en termes d'opérations élémentaires.*

Néanmoins, il est à noter que lors des deux conversions le canal redondant fait partie de la base de destination. Comme souligné dans la Remarque 1.10 (p. 30), l'ajout de ce canal n'augmente donc pas le nombre d'étapes parallèles nécessaires pour exécuter la multiplication modulaire en RNS redondant (Algo. 13) par rapport à la multiplication modulaire RNS dérivée de l'Algorithme 8.

Par conséquent, l'intégration du canal $\mathbb{Z}/m_R\mathbb{Z}$ pour la détection des fautes de catégorie 1, 2, 4 et 5 dans une architecture parallèle dédiée n'entraîne aucune augmentation du temps d'exécution de la multiplication modulaire.

En revanche, la détection des fautes de catégorie 3 nécessite une conversion de base supplémentaire effectuée sur les résidus obtenus par Bex_2 dans $\mathcal{B}$. Le surcoût en terme d'opérations élémentaires est donné par les Équations (1.12), (1.17) et (1.25), et est résumé dans la Table 2.2 suivante. Cependant, cette catégorie reste assez singulière. D'une part, les résidus dans la base $\mathcal{B}'$ suffisent à définir entièrement le résultat de la multiplication modulaire et sont garantis intègres par la procédure de détection. Ceux-ci suffisent donc s'il s'agit de convertir le résultat dans un autre système de numération. D'autre part, en s'inscrivant dans un cadre plus général de RNS redondant, l'ensemble des résidus $(s_{\mathcal{B}}, s_{\mathcal{B}'}, s_R)$ peut n'être qu'un résultat intermédiaire d'une série de calculs dont le résultat final serait, de toute façon, analysé une nouvelle fois par une procédure de détection. La remarque suivante est donc pertinente :

Remarque 2.6 *La solution de protection de la multiplication modulaire contre les attaques par injection de faute unique proposée par l'Algorithme 13 et exploitée dans le contexte de parallélisation naturel du RNS satisfait les attentes fixées par l'Objectif2.1.*

hypothèses	$\mathcal{H}_{mrs}$	$\mathcal{H}_{sk}$	$\mathcal{H}_{kw}$
Mult.	$\frac{(n+2)(n-1)}{2}$ MME1	$(2n+1)$ MME1 $+(n+1)$ $\text{Mul}_{\log_2(n)}$	$2n$ MME1
Add.	$\frac{(n+2)(n-1)}{2}$ AME1	n AME1 $+ n$ $\text{Add}_{\log_2(n)}$	$(2n-2)$ AME1 $+n$ Add_h

TABLE 2.2 – *Surcoût de la procédure de détection des fautes uniques de catégorie 3 en termes d'opérations élémentaires.*

2.3.4 Adéquation de l'Algorithme 13 avec la contre-mesure LRA (Bajard et al. 2004)

La capacité de détection de faute proposée par l'Algorithme 13 peut être facilement associée à une arithmétique résistante aux fuites. Les étapes de calcul (2.1) et (2.2) nécessaires au changement de représentation de Montgomery

sont de simples multiplications modulaires. Elles peuvent donc être réalisées via l'Algorithme 13.

Comme il est nécessaire que le modulus redondant soit supérieur à tous les autres moduli, celui-là ne peut donc faire parti de l'ensemble de moduli constituant le « pool » parmi lequel les bases $\mathcal{B}$ et $\mathcal{B}'$ sont choisies aléatoirement. La robustesse offerte par le masquage aléatoire n'en est en rien affectée.

Un changement de représentation à la volée pour un changement des bases $\mathcal{B}_\alpha \cup \mathcal{B}'_\alpha$ en $\mathcal{B}_\beta \cup \mathcal{B}'_\beta$ s'effectue donc toujours de la même manière :

$$\begin{aligned} y &= \varphi^{-1}_{\mathcal{B}_\alpha \cup \mathcal{B}'_\alpha \cup \{m_R\}} \left(\text{MulModRRNS} \left(\mathcal{B}_\alpha, \mathcal{B}'_\alpha, m_R, x \times |M_\alpha|_p, 1, p \right) \right) \\ &\equiv x \times M_\alpha \times M_\alpha^{-1} \bmod p \\ &\equiv x \bmod p, \\ z &= \varphi^{-1}_{\mathcal{B}'_\beta \cup \mathcal{B}_\beta \cup \{m_R\}} \left(\text{MulModRRNS} \left(\mathcal{B}'_\beta, \mathcal{B}_\beta, m_R, y, |\mathcal{M}|_p, p \right) \right) \\ &\equiv y \times \mathcal{M} \times (M'_\beta)^{-1} \bmod p \\ &\equiv y \times M_\beta \bmod p \\ &\equiv x \times M_\beta \bmod p. \end{aligned} \tag{2.29}$$

Les changements de représentation sont ainsi également protégés contre les attaques par injection de fautes.

Remarque 2.7 *L'Algorithme 13 remplit donc les conditions satisfaisant l'Objectif 2.2.*

2.4 Concernant une adaptation à l'architecture Cox-Rower

Cette partie donne des pistes pour adapter une architecture Cox-Rower sur laquelle implanter l'Algorithme 13. Pour ce faire, nous discutons de l'adéquation du modèle de faute théorique introduit précédemment aux contraintes réelles imposées par les techniques d'attaque par faute sur un matériel physique. Enfin, nous étudions les effets d'un modèle de faute plus fin intégrant les largeurs des registres stockant les valeurs traitées par les Rowers.

2.4.1 Considérations pragmatiques sur la pertinence du modèle de faute théorique

En vue d'étudier la faisabilité d'une implantation matérielle de la solution de détection proposée, nous devons d'abord considérer dans quelle mesure les effets physiques des méthodes d'injection de faute réellement exploitées dans l'état de l'art peuvent être raisonnablement représentés dans notre propre modèle de faute.

Comme souligné par Barenghi et al. (2012) la méthodologie de recherche de contre-mesures aux attaques par fautes qui consiste à s'abstraire des contraintes physiques et à étudier le modèle mathématique sous-jacent sur lequel est construit la primitive implantée peut largement être sujette à critique. Cependant, il est important de préciser qu'une telle approche présente au moins l'avantage d'apporter une preuve formelle de la détection d'un certain type d'erreurs. Par suite, même si le contexte d'une implantation physique apporte son lot de facteurs difficilement interprétables dans le modèle théorique, il n'en demeure pas moins qu'une attaque physique est garantie inopérante de

l'instant où le résultat qu'elle produit sur les données traitées est effectivement pris en compte par ce même modèle théorique. Cette constatation, bien que triviale, se révèle particulièrement pertinente dans le cas du RNS à cause de la « segmentation » des calculs dans des unités qui, dans le cas d'une architecture Cox-Rower, sont physiquement distinctes.

De la représentation des effets de fautes physique dans le modèle théorique

L'architecture Cox-Rower de Kawamura et al. (2000) dispose d'un certain nombre de cellules de traitement, les Rowers, chacune étant dédiée aux calculs dans les canaux RNS « théoriques » $\mathbb{Z}/m\mathbb{Z}$. En pratique, dans le cas de la réduction modulaire RNS utilisant deux bases $\mathcal{B}$ et $\mathcal{B}'$, chaque Rower est généralement alloué à au moins un modulus de $\mathcal{B}$ et un modulus de $\mathcal{B}'$. Comme les calculs dans $\mathcal{B}$ et $\mathcal{B}'$ se font de manière séquentielle lors d'une réduction modulaire (*i.e.* selon un schéma $\mathcal{B} \rightarrow \mathcal{B}' \rightarrow \mathcal{B}$), la pertinence de la restriction du nombre de Rowers à $\max(\mathrm{Card}(\mathcal{B}), \mathrm{Card}(\mathcal{B}'))$ s'impose d'elle-même. En revanche, lorsque les contraintes de surface ne permettent pas d'avoir autant de Rowers, chacun d'eux peut alors être affecté à plusieurs canaux de chacune des deux bases.

Un état de l'art fournit par Barenghi et al. (2012) détaille les moyens dont peut disposer un attaquant pour injecter des fautes dans l'ensemble des valeurs traitées par système physique embarquant l'implantation d'une primitive cryptographique. La puissance de l'adversaire s'évalue essentiellement par les moyens financiers dont il dispose pour monter son attaque. Par exemple, ceux-ci influent directement sur la capacité à viser une zone précise du matériel (cf. tableau 1 de Barenghi et al. (2012)).

Plus généralement, les effets à prendre en compte sont multiples (Otto 2004) : localisation spatiale, contrôle du timing, faute permanente ou transitoire, etc. Cependant, ce genre de classification ne peut véritablement s'appliquer tel quel à nos propos. En fait, il s'agit de voir qu'il nous suffit de considérer que toute faute induite par une attaque physique est intégrable à notre modèle de l'instant où l'effet d'une telle faute peut se réduire à une faute sur une sortie d'un Rower. Par exemple, une attaque modifiant tout ou partie de précalculs contenus dans une ROM, de manière permanente ou non, s'intègre à notre modèle de l'instant où les valeurs modifiées ne concernent qu'un canal RNS $\mathbb{Z}/m\mathbb{Z}$. Que de telles fautes soient permanentes ou transitoires n'a pas véritablement de conséquence à cause du fait que la détection est réalisée systématiquement à la fin de chaque réduction modulaire. Même dans le cas d'une réduction fainéante, où une réduction modulaire n'intervient qu'après le calcul d'une somme plus ou moins grande, l'effet d'une faute permanente est aisément contenu grâce au fait que l'indépendance des canaux RNS n'est brisée qu'au moment d'une réduction modulaire.

Les considérations précédentes mettent également en lumière la nécessité de l'Hypothèse 2.2 (p. 62). Lors d'une conversion de base, chaque Rower distribue sa sortie aux autres. Cette phase peut être réalisée simplement via l'utilisation d'un registre à décalages envoyant séquentiellement les valeurs. Cependant, un problème surgit du fait que tout ou partie des sorties distribuées transitent par de mêmes registres. Dans ce cas précis, une faute modifiant de manière permanente ne serait-ce qu'un bit de registre va possiblement impacter sur la valeur de plusieurs résidus, brisant de fait le modèle de faute unique.

L'utilisation d'un multiplexeur, ou la redondance de la totalité des registres sont des solutions de protection possibles.

Pour résumer, la pertinence du cadre fixé par notre modèle de faute repose sur l'indépendance des unités RNS lors des calculs arithmétiques de base, ainsi que sur le fait que notre détection intervient systématiquement à chaque fois que cette indépendance est levée, à savoir lors d'une réduction modulaire. Les solutions de protection pouvant permettre de lever l'Hypothèse 2.2 ne feront pas nécessairement partie du cœur des propos qui suivront. En revanche, l'adaptation de notre solution de détection de faute doit absolument intégrer les conséquences de la contrainte inévitable de représentation des résidus en binaire.

De la pertinence d'une protection par détection

Outre l'approche par détection, Yen et al. (2003) et Guilley et al. (2010) prônent une solution de protection par résilience. Par définition, un système résilient va absorber la faute, laquelle doit alors infecter suffisamment le résultat final pour que celui-ci ne contienne aucune information exploitable. Ce type de méthode permet notamment d'éviter tout blocage du système, stratégie typiquement appliquée avec une méthode par détection. En revanche, il convient de s'assurer qu'une telle dissémination ne puisse provoquer d'autres types de fuite d'information récupérable via d'autres canaux.

La contre-mesure développée par Guilley et al. (2010) se situe au niveau matériel, tandis que celle proposée par Yen et al. (2003) est au niveau algorithmique et concerne le protocole RSA-CRT. L'argument avancé par ces derniers pour éviter les approches par détection se fonde sur le fait que la logique de détection peut potentiellement être elle-même mise hors-jeu par une attaque par faute. Il est certes difficile de contourner ce problème. Il paraît guère raisonnable par exemple de chercher à s'en prémunir en dupliquant le module de détection, puisque comme souligné par Guilley et al. (2010) il faudrait alors procéder de même concernant le module dupliqué afin de le protéger, et ainsi de suite. Ce désagrément semble donc inhérent à ce type de méthode, et doit faire l'objet d'une attention particulière dans la stratégie de défense appliquée. Cependant, si ce genre de faute est seulement transitoire, alors le modèle que nous utilisons veut qu'une telle faute apparaisse nécessairement seule lors d'une réduction modulaire. Par conséquent, dans ce cas, aucun résidu n'est supposé être fauté, et la conséquence est nulle. La faute étant transitoire et notre technique de détection s'effectuant « à la volée », la prochaine réduction modulaire sera toujours dotée de la capacité à détecter une nouvelle faute unique.

Enfin, la possibilité de détection « à la volée » présente le précieux avantage de donner le choix à l'utilisateur de stopper le protocole presque immédiatement après l'apparition d'une faute. Cela prévient ainsi toute analyse postérieure qui pourrait exploiter les effets de l'altération des calculs (récupération d'un résultat modifié, de traces de consommation, etc).

2.4.2 Raffinement du modèle de faute

Comme souligné par Guillermin (2011), une faute affectant une valeur traitée par un Rower se définit par l'effet qu'elle produit sur la valeur du registre

de sortie dudit Rower qui stocke les quantités $\xi_{i,x,\mathcal{B}}$ ou $\xi_{j,x,\mathcal{B}'}$ lors des changements de base. La taille r de ce registre détermine la taille des moduli des bases RNS utilisées : $2^{r-1} < m < 2^r$ pour tout $m \in \mathcal{B} \cup \mathcal{B}'$.

Définition 2.5 *Si r est la largeur du registre de sortie d'un Rower, une faute matérielle sur une valeur x d'un canal RNS $\mathbb{Z}/m\mathbb{Z}$ est un entier $e \in [\![-x, 2^r - x[\![$.*

Si la faute e est telle que $x + e \in [\![0, m[\![$, alors le modèle de faute théorique donné par la Définition 2.4 est immédiatement applicable. Il reste à analyser comment détecter les cas où $x + e \in [\![m, 2^r[\![$. Cette analyse est l'objet du Théorème 2.5. Avant de l'énoncer, nous rappelons certaines notations introduites lors de la présentation de la conversion de Kawamura et al. (2000) (p. 26). La fonction $\text{eval}_h(\xi)$ renvoie les h bits de poids fort en considérant que ξ s'écrit sur r bits. $\Delta_{kw} \in [0, 1[$ est un majorant de l'erreur commise par l'utilisation de cette fonction eval_h lorsqu'elle est utilisée pour estimer le coefficient $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ lors d'une conversion de base d'un entier x, c'est-à-dire un majorant de $\sum_{i=1}^{n} \left(\frac{\xi_{x,i,\mathcal{B}}}{m_i} - \text{eval}_h(\xi_{x,i,\mathcal{B}}) \right)$. Et $\alpha_{kw} \in [\Delta_{kw}, 1[$ est un coefficient permettant de corriger cette erreur afin de garantir une conversion complète lorsque $x < (1 - \alpha_{kw}) M$.

Théorème 2.5 *Soit r la taille du registre de sortie des Rowers et celle des moduli des bases $\mathcal{B}$ et $\mathcal{B}'$, et xy l'entrée de l'Algorithme 13 (p. 67) de multiplication modulaire vérifiant $xy < \sigma p^2$. Soit les deux ensembles d'hypothèses suivants :*

$$\mathcal{H}_1 : \begin{cases} \sigma \geqslant 4 \\ \Delta_{kw} < \min\limits_{i=1,..,n} \text{eval}_h(m_i) \\ \sigma p < \left(\min\limits_{i=1,..,n} \text{eval}_h(m_i) - \Delta_{kw} \right) M \\ 2p < (1 - \alpha_{kw}) M' \end{cases} \tag{2.30}$$

$$\mathcal{H}_2 : \begin{cases} \sigma \geqslant 9 \\ \sigma p < \left(1 + \max\limits_{i=1,..,n} \text{eval}_h(m_i) - \Delta_{kw} \right) M \\ 3p < (1 - \alpha_{kw}) M' \end{cases}$$

Alors l'Algorithme 13 utilisé avec les hypothèses $\mathcal{H}_1$ ou $\mathcal{H}_2$ et avec un modulus redondant vérifiant :

$$m_R > \max\left(m \in \mathcal{B} \cup \mathcal{B}', \max\{a \in [\![0, 2^r[\![\mid \text{eval}_h(a) \geqslant 1 + \Delta_{kw} - \alpha_{kw}\} \right) \tag{2.31}$$

détecte les fautes matérielles de catégorie 1, 2 et 5. De plus, si aucune faute n'est injectée et si $\mathcal{H}_1$ (resp. $\mathcal{H}_2$) est vérifié, la sortie s de la multiplication modulaire est dans $[\![0, 2p[\![$ (resp. $[\![0, 3p[\![$).

Enfin, sous l'hypothèse que tout modulus m' de $\mathcal{B}'$ vérifie $2^r - m' < 2^c$, alors si :

$$\Delta_{kw} + \frac{1}{2^{r-c}} + \frac{1}{2^h} - \frac{1}{2^r} \leqslant \alpha_{kw} < 1, \tag{2.32}$$

il suffit que la redondance vérifie la condition suivante :

$$m_R > \max\left(m \in \mathcal{B} \cup \mathcal{B}' \right). \tag{2.33}$$

Remarque 2.8 *Dans l'énoncé du théorème, nous passons sous silence les fautes de catégorie 3, pour lesquelles il est nécessaire de recourir à une conversion de base spécialement dédiée à la détection de ces fautes. Les fautes de catégorie 4 sont également omises. Nous prenons le parti de considérer le Cox protégé par une redondance en matériel, en dotant par exemple chaque Rower d'un Cox.*

Démonstration. La détection des fautes de catégorie 5 demeure encore une fois évidente.

Catégorie 1 Sans perte de généralité, nous considérons une faute sur le premier canal :

$$\overline{\xi}_{1,q,\mathcal{B}} = \xi_{1,q,\mathcal{B}} + e_1 = m_1 + f_1,\ f_1 \in [\![0, 2^r - m_1[\![\subset [\![0, m_1[\![. \tag{2.34}$$

En effet, si $\overline{\xi}_{1,q,\mathcal{B}} < m_1$, alors le modèle de faute théorique s'applique, et la détection a été prouvée. Nous nous intéressons donc au cas $m_1 \leqslant \overline{\xi}_{1,q,\mathcal{B}} < 2^r$. L'entier de $[\![0, M[\![$ représenté par les coefficients $\left(\overline{\xi}_{1,q,\mathcal{B}}, \xi_{2,q,\mathcal{B}}, \ldots, \xi_{n,q,\mathcal{B}}\right)$ a pour résidus $\overline{\boldsymbol{q}}_{\mathcal{B}} = \left(|f_1 M_1|_{m_1}, q_2, \ldots, q_n\right)$ dans la base $\mathcal{B}$. Notons alors $\overline{q} = \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right)$. Ceci signifie que $\overline{\boldsymbol{q}}_{\mathcal{B}}$ est issu des résidus de q affectés par une faute théorique.

La première conversion fait intervenir le calcul des deux quantités suivantes :

$$\begin{aligned}
\overline{\xi}_{1,q,\mathcal{B}} M_1 + \sum_{i=2}^{n} \xi_{i,q,\mathcal{B}} M_i &= \sum_{i=1}^{n} \xi_{i,q,\mathcal{B}} M_i + e_1 M_1 \\
&= \mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}\right) + e_1 M_1 \\
&= \mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}\right) + (m_1 + f_1 - \xi_{1,q,\mathcal{B}}) M_1 \\
&= \mathrm{sum}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) + M, \qquad (2.35) \\
\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) &= \lfloor \mathrm{eval}_h\left(\overline{\xi}_{1,q,\mathcal{B}}\right) + \sum_{i=2}^{n} \mathrm{eval}_h\left(\xi_{i,q,\mathcal{B}}\right) \rfloor. \qquad (2.36)
\end{aligned}$$

Comme $\mathrm{eval}_h(\overline{\xi}_{1,q,\mathcal{B}}) = \mathrm{eval}_h(f_1) + \mathrm{eval}_h(m_1) + \dfrac{\delta}{2^h}$, $\delta \in \{0, 1\}$, nous pouvons alors écrire les inégalités suivantes :

$$\begin{cases}
\mathrm{eval}_h\left(\overline{\xi}_{1,q,\mathcal{B}}\right) + \displaystyle\sum_{i=2}^{n} \mathrm{eval}_h\left(\xi_{i,q,\mathcal{B}}\right) \leqslant \dfrac{\overline{q}}{M} + \kappa_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) + \mathrm{eval}_h(m_1) + \dfrac{1}{2^h} \\
\qquad\qquad\qquad\qquad\qquad\qquad < \kappa_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) + 2, \\
\mathrm{eval}_h\left(\overline{\xi}_{1,q,\mathcal{B}}\right) + \displaystyle\sum_{i=2}^{n} \mathrm{eval}_h\left(\xi_{i,q,\mathcal{B}}\right) \geqslant \dfrac{\overline{q}}{M} + \kappa_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) + \mathrm{eval}_h(m_1) - \Delta_{kw} \\
\qquad\qquad\qquad\qquad\qquad\qquad \geqslant \kappa_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) - 1.
\end{cases} \tag{2.37}$$

Ces inégalités montrent que le résultat du calcul du Cox donné par l'Équation (2.36) donne $\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) = \kappa_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) + \mu$, avec $\mu \in \{-1, 0, 1\}$. Le cas $\mu = 1$ permet de corriger le terme M supplémentaire apparaissant dans l'Équation (2.35). De plus, il est clair d'après la seconde ligne du système (2.37) que $\mu = -1$ implique $\overline{q} < (\Delta_{kw} - \mathrm{eval}_h(m_1))\, M$.

Finalement, la conversion donne la valeur suivante :

$$\widehat{\overline{q}} = \mathrm{sum}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) + M - \tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) M = \overline{q} + (1-\mu) M \text{ avec } \mu \in \{-1, 0, 1\}. \tag{2.38}$$

De plus, $\widehat{\overline{q}}$ vérifie aussi :

$$\widehat{\overline{q}} < (2 + \Delta_{kw} - \mathrm{eval}_h(m_1)) M. \tag{2.39}$$

Il est clair que si $f_1 = \xi_{1,q,\mathcal{B}}$ dans l'Équation (2.34), la perturbation ne modifie pas le résultat final qui reste correct. De plus, sous les hypothèses $\mathcal{H}_1$,

$$s = \frac{xy + \widehat{\overline{q}}p}{M} < 2p < (1 - \alpha_{kw}) M'.$$

De la même manière, sous les hypothèses $\mathcal{H}_2$, l'Inéquation (2.39) implique :

$$s = \frac{xy + \widehat{\overline{q}}p}{M} < 3p < (1 - \alpha_{kw}) M'.$$

Pour résumer, si $f_1 \in \{0, \xi_{1,q,\mathcal{B}}\}$, alors la faute est corrigée. Autrement, le modèle théorique est utilisable, et la faute sera détectée puisque $f_1 \in [\![0, 2^r - m_1[\![\subset [\![0, m_1[\![$.

Catégorie 2 Comme pour la catégorie 1, nous étudions le cas :

$$\overline{\xi}_{1,s,\mathcal{B}'} = \xi_{1,s,\mathcal{B}'} + e_1,\ e_1 \in [\![m'_1 - \xi_{1,s,\mathcal{B}'}, 2^r - \xi_{1,s,\mathcal{B}'}[\![, \tag{2.40}$$

puisque le cas $e_1 \in [\![-\xi_{1,s,\mathcal{B}'}, m'_1 - \xi_{1,s,\mathcal{B}'}[\![$ correspond au modèle théorique. Ainsi, il vient :

$$\begin{aligned} \overline{\xi}_{1,s,\mathcal{B}'} M'_1 + \sum_{i=2}^{\ell} \xi_{i,s,\mathcal{B}'} M'_i &= \sum_{i=1}^{\ell} \xi_{i,s,\mathcal{B}'} M_i + e_1 M'_1 \\ &= \mathrm{sum}_{\mathcal{B}}(\boldsymbol{s}_{\mathcal{B}}) + e_1 M_1 \qquad (2.41) \\ & \qquad (2.42) \end{aligned}$$

Par hypothèse,

$$\begin{cases} s < (1 - \alpha_{kw}) M', \\ \mathrm{eval}_h(e_1) \leqslant \mathrm{eval}_h(2^r - 1) = 1 - \dfrac{1}{2^h}. \end{cases} \tag{2.43}$$

Nous avons ainsi les inégalités suivantes :

$$\begin{cases} \mathrm{eval}_h\left(\overline{\xi}_{1,s,\mathcal{B}'}\right) + \displaystyle\sum_{i=2}^{\ell} \mathrm{eval}_h(\xi_{i,s,\mathcal{B}'}) + \alpha_{kw} \leqslant \dfrac{s}{M'} + \kappa_{\mathcal{B}'}(\boldsymbol{s}_{\mathcal{B}'}) + \mathrm{eval}_h(e_1) + \dfrac{1}{2^r} + \alpha_{kw} \\ \qquad\qquad < \kappa_{\mathcal{B}'}(\boldsymbol{s}_{\mathcal{B}'}) + 2, \\ \mathrm{eval}_h\left(\overline{\xi}_{1,s,\mathcal{B}'}\right) + \displaystyle\sum_{i=2}^{\ell} \mathrm{eval}_h(\xi_{i,s,\mathcal{B}'}) + \alpha_{kw} \geqslant \dfrac{s}{M'} + \kappa_{\mathcal{B}'}(\boldsymbol{s}_{\mathcal{B}'}) + \mathrm{eval}_h(e_1) - \Delta_{kw} + \alpha_{kw} \\ \qquad\qquad \geqslant \kappa_{\mathcal{B}'}(\boldsymbol{s}_{\mathcal{B}'}) - \Delta_{kw} + \alpha_{kw} \\ \qquad\qquad \geqslant \kappa_{\mathcal{B}'}(\boldsymbol{s}_{\mathcal{B}'}). \end{cases} \tag{2.44}$$

Vu les Inégalités (2.44), le calcul du Cox donne donc $\tilde{\kappa}_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) = \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + \mu$, avec $\mu \in \{0,1\}$. Par conséquent, reprenant l'Équation (2.41), il vient :

$$\begin{aligned} \mathrm{Bex}_{kwc}\left(\mathcal{B}', \bar{\boldsymbol{s}}_{\mathcal{B}'}, \alpha_{kw}\right) &= \mathrm{sum}_{\mathcal{B}}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + e_1 M_1' - \tilde{\kappa}_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) M' \\ &= s + \left(e_1 - \mu m_1'\right) M_1'. \end{aligned} \tag{2.45}$$

Or, vu (2.44), $\mu = 0$ implique que :

$$\frac{s}{M'} + \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + \mathrm{eval}_h\left(e_1\right) - \Delta_{kw} + \alpha_{kw} < \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + 1.$$

Et donc :

$$\mu = 0 \Rightarrow \mathrm{eval}_h\left(e_1\right) < 1 + \Delta_{kw} - \alpha_{kw}.$$

Par conséquent, en notant $A = \max\{a \in [\![0, 2^r[\![\mid \mathrm{eval}_h\left(a\right) \geqslant 1 + \Delta_{kw} - \alpha_{kw}\}$, nous avons :

$$\left|\frac{\mathrm{Bex}_{kwc}\left(\mathcal{B}', \bar{\boldsymbol{s}}_{\mathcal{B}'}, \alpha_{kw}\right) - s}{M_1'}\right| < \max\left(m \in \mathcal{B} \cup \mathcal{B}', A\right).$$

Ces inégalités montrent que la condition (2.31) du présent théorème suffit pour assurer la détection.

Afin de montrer maintenant la suffisance de la condition $m_R > m$ pour tout $m \in \mathcal{B} \cup \mathcal{B}'$ sous l'Hypothèse 2.32, il faut voir que si $e_1 \geqslant m_1'$ alors :

$$\mathrm{eval}_h\left(e_1\right) \geqslant \mathrm{eval}_h\left(m_1'\right) \geqslant \frac{2^r - 2^c - \left|2^r - 2^c\right|_{2^{r-h}}}{2^r} = 1 - \frac{1}{2^{r-c}} - \frac{1}{2^h} + \frac{1}{2^r}.$$

Ainsi, l'Hypothèse (2.32) implique que $\mathrm{eval}_h\left(e_1\right) - \Delta_{kw} + \alpha_{kw} \geqslant 1$ et donc, vu la seconde inéquation du système (2.44) que $\tilde{\kappa}_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) = \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + 1$, c'est-à-dire $\mu = 1$. Ainsi, si $e_1 \geqslant m_1'$ alors $\mu = 1$ et par suite :

$$\left|e_1 - m_1'\right| < 2^r - m_1' < m_1'.$$

Ainsi, dans tous les cas, $\left|e_1 - \mu m_1'\right| < m_1'$, ce qui achève la preuve.

□

Exemple 2.1 *Reprenant les paramètres de l'Exemple 1.4 calibrés pour le calcul d'un RSA de 1024 bits, $n = 33$ et $r = 32$, avec des moduli pseudo-Mersenne, la condition $\Delta_{kw} < \min_{i=1,..,n} \mathrm{eval}_h\left(m_i\right)$ est vérifiée dès que $h \geqslant 7$. Dans ce cas, il vient alors notamment que $\min_{i=1,..,n} \mathrm{eval}_h\left(m_i\right) - \Delta_{kw} > \frac{1}{2}$. Il suffit donc de choisir un coefficient correcteur $\alpha_{kw} = \frac{1}{2}$ et des tailles de bases RNS vérifiant $M > 2\sigma p$ et $M' > 4p$.*

Ces choix de paramètres satisfont également la Condition (2.32). Ainsi, m_R doit simplement être choisi premier à MM' et vérifier la Condition (2.33), à savoir m_R plus grand que tout modulus de $\mathcal{B} \cup \mathcal{B}'$.

Architecture adaptée

L'architecture proposée Figure 2.8 est adaptée de celle de Nozaki et al. (2001).

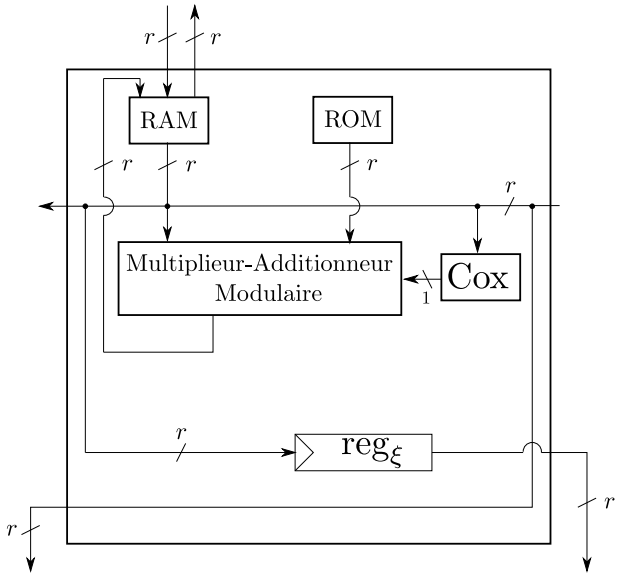


FIGURE 2.6 – *Unité Rower modifiée.*

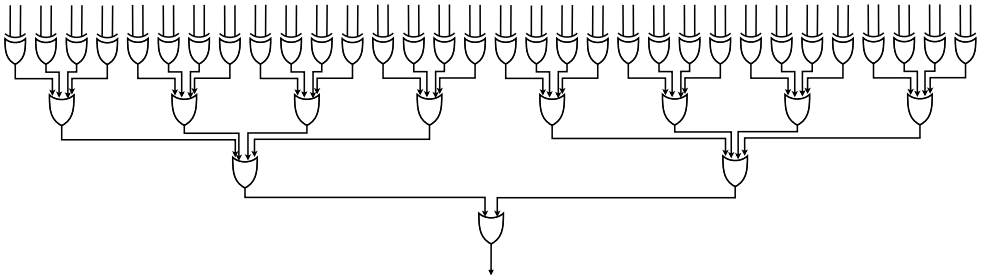


FIGURE 2.7 – *Unité de détection de faute (FDU) avec* $r = 32$.

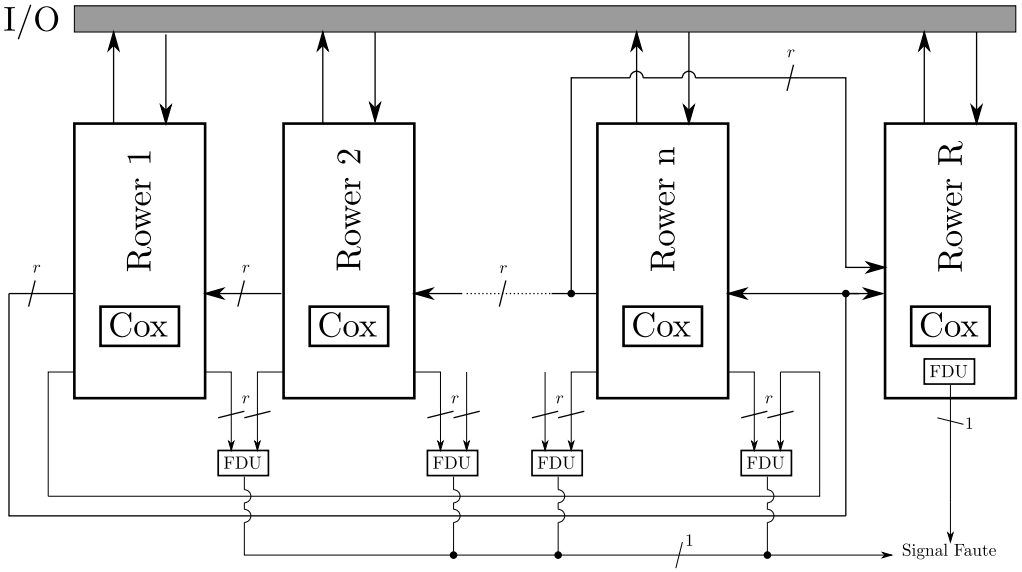


FIGURE 2.8 – *Principe d'une architecture de type Cox-Rower avec capacité de détection de faute.*

Au vu des conditions du Théorème 2.5 sur la redondance, il est toujours possible de choisir $m_R = 2^r$. Dans ce cas, un Rower dédié au canal redon-

dant possède une structure plus simple qu'un Rower classique présenté Figure 1.2, puisque la réduction modulo 2^r est alors une simple troncature. Comme chaque Rower est généralement dédié au calcul dans au moins un canal de $\mathcal{B}$ et un canal de $\mathcal{B}'$, si le modèle d'attaquant contre lequel il convient de se protéger prend en compte la possibilité de perturbations localisées sur une mémoire d'un Rower via un outil assez précis comme un laser, il peut être nécessaire d'envisager de stocker les valeurs précalculées afférentes aux différents canaux dans différentes ROM espacées. Plus généralement, le nombre u de Rowers est avantageusement choisi comme divisant n, taille des bases $\mathcal{B}$ et $\mathcal{B}'$ en terme de nombre de moduli que nous supposons identiques par soucis de simplicité. En primant l'optimisation du temps de calcul sur la surface utilisée, n Rowers sont utilisés, ce qui permet de plus de diminuer les risques de perturbations locales affectant des valeurs concernant plusieurs canaux RNS. Une autre contre-mesure serait de rajouter de la redondance pour fournir une capacité de détection de fautes multiples, ce qui sera étudié par la suite (cf. Section 3.1).

Si l'Hypothèse 2.2 (p. 62) ne peut être posée, alors il faut vérifier la bonne propagation des valeurs de type $\xi_{i,q,\mathcal{B}}$ et $\xi_{j,s,\mathcal{B}'}$ durant les deux conversions de base. Pour ce faire, l'architecture de la Figure 2.8 intègre dans chaque Rower une unité de détection de faute. Lors d'une conversion de base, les n canaux doivent envoyer leur valeur de sortie ξ_i vers les $n-1$ autres canaux et le canal redondant. La propagation se fait de manière circulaire en $n-1$ étapes. Afin de vérifier que les propagations n'ont pas été perturbées, chaque Rower i stocke sa propre valeur ξ_i dans un registre reg_ξ. Au terme de la $n-1$-ième étape, celle-ci est comparée avec la dernière valeur reçue par le Rower numéro $(i+1) \bmod n$ via une unité FDU. Cette étape de comparaison est indépendante du flux de calcul principal. Cette solution de redondance des registres de sortie sera forcément plus coûteuse lorsque nous étendrons le modèle aux fautes multiples.

Un seul Cox pourrait être également utilisé (cf. Prop. 2.8, p. 72), mais s'agissant d'une unité peu coûteuse en surface, chaque Rower peut posséder son propre Cox. Cela permet de conserver la propriété d'avoir un résultat final correct dans $[\![0, 2p[\![$ (contrairement au cas d'un Cox unique selon la stratégie suggérée par la Prop. 2.8). De plus, concernant la Proposition 2.8, le modèle de faute adopté a été celui d'une faute transitoire sur le registre d'accumulation du Cox. Il est clair qu'une faute permanente pourrait provoquer l'apparition de plusieurs retenues sortantes supplémentaires (jusqu'à n), ou la disparition de tout ou partie des retenues qui auraient du apparaître. Il est alors toujours possible d'adapter l'énoncé de cette proposition à ce cas particulier.

Enfin le Rower dédié au canal redondant possède sa propre unité FDU pour l'exécution du test de cohérence.

Dans Bajard, Eynard, et Gandino (2013b) une étude comparative avec des paramètres similaires à ceux de l'Exemple 1.4 est fournie. Celle-ci montre que la surface supplémentaire totale consacrée à un Rower redondant dédié au modulus $m_R = 2^r$ et aux unités FDU reste inférieure à celle d'un Rower standard.

2.4.3 Comparaison avec la technique de détection de Guillermin (2011)

Détection des fautes de catégorie 1 et 2 via un Cox modifié

Guillermin (2011) propose une technique de détection utilisant un Cox mo-

difié qui permet d'obtenir une meilleure approximation des fractions $\frac{\xi_i}{m_i}$ dans l'Équation (1.19). Pour l'obtenir, il introduit une nouvelle fonction eval. Plus précisément, là où Kawamura et al. (2000) proposent d'approcher $\frac{1}{m_i}$ par $\frac{1}{2^r}$, Guillermin utilise une somme partielle d'ordre t de la série géométrique de somme $\frac{1}{m_i}$:

$$\mathrm{eval}_t\left(\frac{\xi_i}{m_i}\right) = \frac{\xi_i}{2^r}\sum_{j=0}^{t}\left(\frac{2^r - m_i}{2^r}\right)^j.$$

La détection est effectuée lors de la conversion Bex_2. Le principe de cette technique repose sur les constatations suivantes, où $\alpha_{kw} \in [\Delta_{kw}, 1[$ est toujours un coefficient corrigeant l'erreur totale induite par les approximations :

$$\begin{cases} \displaystyle\sum_{i=1}^{\ell} \mathrm{eval}_t\left(\xi_{i,s,\mathcal{B}'}\right) + \alpha_{kw} \leqslant \sum_{i=1}^{\ell} \frac{\xi_{i,s,\mathcal{B}'}}{m'_i} + \alpha_{kw} = \frac{s}{M'} + \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + \alpha_{kw} \\ \qquad\qquad\qquad\qquad < \dfrac{3p}{M'} + \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + \alpha_{kw} < \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + 1 \\ \displaystyle\sum_{i=1}^{\ell} \mathrm{eval}_t\left(\xi_{i,s,\mathcal{B}'}\right) + \alpha_{kw} \geqslant \frac{s}{M'} + \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) + \alpha_{kw} - \Delta_{kw} \geqslant \kappa_{\mathcal{B}'}\left(\boldsymbol{s}_{\mathcal{B}'}\right) \end{cases}$$

$$\Rightarrow \frac{s}{M'} \leqslant \mathrm{reg}_{acc} = \sum_{i=1}^{\ell} \mathrm{eval}_t\left(\xi_{i,s,\mathcal{B}'}\right) - \lfloor \sum_{i=1}^{\ell} \mathrm{eval}_t\left(\xi_{i,s,\mathcal{B}'}\right) \rfloor \leqslant \frac{s}{M'} + \alpha_{kw} < \frac{3p}{M'} + \alpha_{kw}. \tag{2.46}$$

Or, la quantité reg_{acc} est la valeur finale présente dans l'accumulateur du Cox à la fin de la conversion. Ainsi, en choisissant une base $\mathcal{B}'$ assez large pour que :

$$\frac{3p}{M'} < \frac{1}{2^{\sigma+1}} \tag{2.47}$$

et si l'ordre t de la somme définissant la fonction eval_t est assez grand pour pouvoir choisir $\alpha_{kw} = \frac{1}{2^{\sigma+1}}$, alors des Équations (2.46) et (2.47) découle que $\mathrm{reg}_{acc} \in [0, \frac{1}{2^{\sigma}}[$. Autrement dit, reg_{acc} a ses σ bits de poids fort nuls. Fort de cette remarque, Guillermin choisit donc le coefficient σ et la précision t de la fonction eval_t, de manière à ce que pour toute faute $e_i \in [\![-\xi_{1,s,\mathcal{B}'}, 2^r - \xi_{1,s,\mathcal{B}'}]\!]$ commise sur $\xi_{i,s,\mathcal{B}'}$ les inégalités suivantes soient toujours vérifiées :

$$\frac{1}{2^{\sigma}} < \left|\mathrm{eval}\left(e'_i\right)\right| < 1 - \frac{1}{2^{\sigma}} \Rightarrow \frac{1}{2^{\sigma}} < \left|\mathrm{reg}_{acc} + \mathrm{eval}\left(e'_i\right)\right| < 1.$$

En remarquant que :

$$\frac{1}{2^r} < \left|\mathrm{eval}\left(e'_i\right)\right| < 1 - \frac{1}{m_i} < 1 - \frac{1}{2^r},$$

il vient donc la condition suffisante $\sigma \geqslant r + 1$. De cette manière, la présence d'une faute matérielle sur un coefficient $\xi_{1,s,\mathcal{B}'}$ est détectée par la présence d'un bit non nul parmi les σ bits de poids fort de reg_{acc}.

La détection des fautes de catégorie 1 reste plus problématique. Néanmoins, en notant

$$\widehat{\overline{q}} = \mathrm{Bex}_{kw}\left(\mathcal{B}, \mathcal{B}', \overline{\boldsymbol{q}}_{\mathcal{B}}\right) = q + e_i M_i + \delta M$$

avec $0 < |e_i| < m_i$ et $\delta \in \{0, \pm 1\}$, et $\overline{s} = \left|\left(xy + \widehat{\widehat{q}}p\right) M^{-1}\right|_{M'}$, alors $\overline{s} = s + |e_i m_i p + \delta p|_{M'}$ ou $\overline{s} = s - |-e_i m_i p - \delta p|_{M'}$. Par conséquent, en notant $E = \overline{s} - s$, et en reprenant les Inéquations (2.46), il vient :

$$\frac{s+E}{M'} \leqslant \mathrm{reg}_{acc} \leqslant \frac{s+E}{M'} + \alpha_{kw} < \frac{E}{M'} + \frac{1}{2^\sigma}.$$

De ce fait, si

$$\frac{1}{2^\sigma} < \frac{|E|}{M'} < 1 - \frac{1}{2^\sigma}, \tag{2.48}$$

alors l'erreur sera détectée par le Cox modifié. Guillermin exhibe un majorant de la probabilité qu'une erreur E de la forme $|\pm e_i m_i p + \delta p|_{M'}$ ne vérifie pas les Inéquations (2.48) en se basant sur un dénombrement des fautes ayant la forme de E.

Pour se protéger contre les fautes de catégorie 4 sur le Cox, Guillermin suggère d'utiliser une redondance du Cox.

Analyse comparative

La technique de Guillermin repose sur l'utilisation des conversions de type Kawamura et al. (2000), et est donc conditionnée par l'utilisation d'une architecture de type Cox-Rower. *A contrario*, la technique proposée dans ce mémoire ne dépend pas des conversions de base choisies, puisqu'il suffit d'utiliser une redondance adaptée à l'algorithme de réduction modulaire choisi.

Vu la Condition (2.47) sur M' donnée par Guillermin, à savoir $M' > 2^{\sigma+1}3p$ avec la condition supplémentaire $\sigma \geqslant r + 1$, soit donc $M' > 2^{r+2}3p$, et par comparaison avec les Conditions (2.30) de notre étude, à savoir $(1 - \alpha_{kw})M' > 3p$, la technique du Cox modifié nécessite au moins un modulus supplémentaire dans la base $\mathcal{B}'$. D'après la Remarque 1.10 (p. 30), cela impacte nécessairement le temps de calcul de la seconde extension de base, et donc le temps total de calcul d'une réduction modulaire. Les résultats de mesure d'une implantation sur FPGA de sa technique appliquée pour le calcul d'un RSA-CRT de 1024 bits via 2 exponentiations modulaires par échelle de Montgomery montrent une latence de 5% dû à l'utilisation de cette technique de détection. *A contrario* pour la présente technique, le coût supplémentaire en terme de surface est aussi celui d'un Rower pour le canal redondant. De plus, celui-ci ne fait jamais partie de la base de départ d'une conversion. Ainsi, le temps global n'est pas affecté par les détections des fautes de catégorie 1, 2, 4 et 5. Seul la détection des fautes de catégorie 3 induit une augmentation du temps de traitement équivalent à la moitié d'une réduction modulaire, puisque le coût d'une telle réduction est essentiellement celui de ses deux conversions de base. Pour le même RSA-CRT, la latence est donc estimée à $\frac{1}{2\times 1024} \sim 0,05\%$. Mais celle-ci n'est due qu'à une conversion supplémentaire pour la détection d'une faute finale dans $\mathcal{B}$, conversion également nécessaire pour une telle détection dans le cas de la méthode du Cox modifié.

Enfin, la présente technique a l'avantage d'offrir une détection certaine des fautes de catégorie 1, qui constituent un point délicat pour la méthode du Cox modifié, et a le grand avantage de se généraliser aisément à la détection de fautes multiples via l'ajout de canaux redondants supplémentaires, comme il sera vu par la suite.

Conclusion

Un nouvel algorithme de multiplication modulaire RNS résistant aux fautes uniques (Alg. 13, p. 67) a été présenté dans un premier temps. La capacité de détection s'appuie sur les RNS redondants. Ceux-ci permettent une détection de faute dont le principe est basé sur un test de cohérence entre résidus principaux et résidus redondants. Si cette technique est connue de longue date, nous avons précisé des conditions suffisantes sur la redondance qui permettent d'effectuer le test de cohérence à partir de différentes opérations de conversion de base (cf. Th. 2.3, p. 60). Ce processus de détection a été intégré de manière efficace à l'algorithme de multiplication modulaire RNS, en s'appuyant complètement sur la structure même de cet algorithme, et en restant entièrement adaptable à une arithmétique résistante aux fuites (cf. 2.3.4). En conclusion, nous pouvons affirmer avoir satisfait les Objectifs 2.1 et 2.2 (p. 61). Le Théorème 2.4 (p. 66) démontre formellement la validité de l'approche suggérée.

Au final, il résulte qu'au cours d'une exponentiation modulaire exécutée avec échelle de Montgomery, le surcoût en temps pour s'assurer de l'intégrité de la totalité des résidus du résultat final contre la présence d'une faute unique est de l'ordre de $\log_2 \left(exposant^2\right)^{-1}$. Il s'agit du test de cohérence dédié à la vérification de la présence d'une faute de catégorie 3 dans la base RNS principale $\mathcal{B}$ du résultat final.

Une étude supplémentaire incluant un modèle de faute plus fin adapté aux contraintes de représentation binaire des résidus pour une implantation matérielle dans une architecture de type Cox-Rower a permis d'établir des conditions sur la redondance pour la détection de fautes matérielles sur un canal RNS (cf. Th. 2.5, p. 78). Un modèle général d'architecture adaptée au nouvel algorithme de multiplication modulaire en RNS redondant a été suggéré.

3 Vers une arithmétique RNS dans $\mathbb{F}_p$ et $\mathbb{F}_{p^s}$ résistante aux fautes multiples

Sommaire

Nous généralisons le principe de multiplication modulaire en RNS redondant présenté dans le chapitre précédent pour l'adapter à la détection de fautes multiples. Ce modèle permet d'intégrer le principe de tolérance aux pannes. En pratique, les unités physiques (*e.g.* les Rowers) d'une architecture parallèle spécifique permettant l'implantation d'algorithmes RNS pour des calculs sur un corps fini peuvent être dédiées aux calculs dans plusieurs canaux RNS des bases principales et auxiliaires $\mathcal{B}$ et $\mathcal{B}'$. Une perturbation affectant de manière permanente ou transitoire une telle unité physique peut alors être modélisée par une faute multiple, qui par définition modifie plusieurs résidus mis en jeu dans une multiplication modulaire.

Une première partie sera consacrée à l'adaptation de l'Algorithme 13 de multiplication modulaire RNS dotée d'une capacité de détection des fautes uniques au modèle de faute multiple. Après avoir décrit le nouveau modèle de faute multiple, nous fournirons des conditions suffisantes sur la redondance garantissant la détection de ces fautes. Ensuite, nous affinerons le modèle pour

intégrer les contraintes spécifiquement liées à la représentation binaires des résidus erronés. Enfin, nous nous attacherons dans une dernière section à adapter cette approche à l'arithmétique en RNS redondant dans le cas des corps finis non premiers.

3.1 De la détection des fautes multiples

La capacité de détection d'erreur offerte par les RNS redondants dans le cadre d'une arithmétique basique sur les entiers $\mathbb{Z}$ est naturellement généralisable à la détection d'erreurs multiples affectant simultanément plusieurs résidus. Elle a été étudiée par les mêmes auteurs cités dans la section concernant les fautes uniques. Le niveau de protection souhaité est relié au nombre de moduli redondants utilisés et dépend donc de la quantité de ressources que l'utilisateur est disposé à allouer aux canaux redondants. Nous allons voir que les principes de cette généralisation s'appliquent encore une fois très bien aux calculs en RNS redondant en présence de réductions modulaires.

3.1.1 RNS redondants et fautes multiples

Tout comme pour le cas des fautes uniques, nous présentons tout d'abord le modèle de faute, puis nous rappelons les conditions suffisantes pour la détection de ces fautes. Ces conditions dans le cas d'une base RNS redondante $\mathcal{B}_R$ première à la base principale $\mathcal{B}$ se retrouvent dans la même littérature évoquée dans le chapitre des fautes uniques. Nous fournissons en annexe les résultats et preuves dans le cas le plus général où les moduli redondants ne sont ni supposés premiers entre eux deux à deux, ni premiers avec les moduli principaux. Ces résultats très généraux justifient la pertinence du choix de la simplification adoptée dans les parties qui suivent, où $\mathcal{B}_R$ est automatiquement considérée être une base RNS première avec les autres bases RNS utilisées.

Modèle de faute et procédure de détection

Le contexte est celui d'un RNS $\mathcal{B} = \{m_1, \ldots, m_n\}$ muni d'une base RNS de moduli redondants $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$ première avec $\mathcal{B}$. Tout ensemble intègre de résidus $(\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_{\mathcal{B}_R})$ représente un unique entier x de l'intervalle légitime $[\![0, M[\![$. La Définition 2.4 se généralise de la manière suivante :

Définition 3.1 *Soit un entier $d \in [\![1, n]\!]$. Une d-faute affectant les résidus $\boldsymbol{x}_\mathcal{B} = \varphi_\mathcal{B}(x)$ ($x \in [\![0, M[\![$) est la donnée de d indices $\mathcal{I}_d = \{i_1, \ldots, i_d\}$ ($1 \leqslant i_1 < \ldots < i_d \leqslant n$) et de d nombres $e_{i_j} \in [\![1, m_{i_j}[\![$. L'ensemble des résidus erronés est noté $\overline{\boldsymbol{x}}_\mathcal{B}$, où $\overline{x}_j = x_j$ pour tout $j \in [\![1, n]\!] \backslash \mathcal{I}_d$, et $\overline{x}_i = |x_i + e_i|_{m_i}$ pour tout $i \in \mathcal{I}_d$.*

La Proposition 2.1 (p. 53) est adaptée aux fautes multiples afin de déterminer les quantités qui devront être détectées par un test de cohérence.

Proposition 3.1 *Soit $\mathcal{I}_d \subset [\![1, n]\!]$ et $\overline{\boldsymbol{x}}_\mathcal{B}$ les résidus de $x \in [\![0, M[\![$ et une d-erreur $\left(e_{i_1}, \ldots, e_{i_d}\right)$ affectant les résidus d'indices $\mathcal{I}_d$ de x. Notant $\overline{x} = \varphi_\mathcal{B}^{-1}(\overline{\boldsymbol{x}}_\mathcal{B})$ l'entier de l'intervalle $[\![0, M[\![$ représenté par les résidus $\overline{\boldsymbol{x}}_\mathcal{B}$, et $M_{\mathcal{I}_d} = \prod_{j \notin \mathcal{I}_d} m_j$, alors :*

$$\exists a_{\mathcal{I}_d} \in]\!] - \prod_{i \in \mathcal{I}_d} m_i, \prod_{i \in \mathcal{I}_d} m_i [\![\backslash \{0\},\ \overline{x} = x + a_{\mathcal{I}_d} M_{\mathcal{I}_d}. \tag{3.1}$$

Démonstration. Intuitivement, ajouter une faute affectant les résidus d'indices $\mathcal{I}_d$ revient à ajouter à x un multiple de $M_{\mathcal{I}_d}$ (modulo M). Nous montrons que ce multiple est en fait donné par l'entier $b = \sum_{i \in \mathcal{I}_d} \left| e_i M_i^{-1} \right|_{m_i} M_i \bmod M$. En effet, pour tout $i \in \mathcal{I}_d$, $\overline{x}_i = |x_i + b_i|_{m_i} = |x_i + e_i|_{m_i}$ et pour tout $i \notin \mathcal{I}_d$, $\overline{x}_i = x_i$, donc

$\overline{x} = (x + b) \bmod M$. Plus précisément, comme $\sum_{i \in \mathcal{I}_d} \left| e_i M_i^{-1} \right|_{m_i} M_i < dM$, il existe $\beta \in [\![0, d[\![$ tel que :

$$b = \sum_{i \in \mathcal{I}_d} \left| e_i M_i^{-1} \right|_{m_i} M_i - \beta M \in [\![0, M[\![.$$

Soit de plus $\delta \in \{0, 1\}$ tel que $|x + b|_M = x + b - \delta M$. Ainsi, comme $\beta + \delta \leqslant d$, nous pouvons écrire :

$$\begin{aligned}
\overline{x} &= x + b - \delta M \\
&= x + \sum_{i \in \mathcal{I}_d} \left| e_i M_i^{-1} \right|_{m_i} M_i - (\beta + \delta) M \\
&= x + \sum_{j=1}^{\beta+\delta} \left(\left| e_{i_j} M_{i_j}^{-1} \right|_{m_{i_j}} - m_{i_j} \right) M_{i_j} + \sum_{j=\beta+\delta+1}^{t} \left| e_{i_j} M_{i_j}^{-1} \right|_{m_{i_j}} M_{i_j} \\
&= x + \left[\sum_{j=1}^{\beta+\delta} \left(\left| e_{i_j} M_{i_j}^{-1} \right|_{m_{i_j}} - m_{i_j} \right) \prod_{i \in \mathcal{I}_d \setminus \{i_j\}} m_i + \sum_{j=\beta+\delta+1}^{d} \left| e_{i_j} M_{i_j}^{-1} \right|_{m_{i_j}} \prod_{i \in \mathcal{I}_d \setminus \{i_j\}} m_i \right] M_{\mathcal{I}_d} \\
&= x + a_{\mathcal{I}_d} M_{\mathcal{I}_d}.
\end{aligned}$$

Comme $|\overline{x} - x| < M$, alors $\left| a_{\mathcal{I}_d} \right| < \prod_{i \in \mathcal{I}_d} m_i$. De plus $\left| a_{\mathcal{I}_d} \right|_{m_{i_1}} = e_1 \neq 0$ implique que $a_{\mathcal{I}_d} \not\equiv 0 \bmod M$ et donc finalement que $a_{\mathcal{I}_d} \in]\!] - \prod_{i \in \mathcal{I}_d} m_i, \prod_{i \in \mathcal{I}_d} m_i [\![\setminus \{0\}$. □

La Proposition 3.1 précédente précise la forme d'une faute multiple. La proposition suivante va détailler, à partir d'un entier x de l'intervalle légitime, l'ensemble des entiers de cet intervalle qui diffèrent de x sur au plus d résidus. Ceci permettra de déterminer une condition nécessaire sur la taille de la redondance pour détecter les d-fautes.

Proposition 3.2 *Soit $d \in [\![1, n]\!]$, $\mathcal{I}_d \subset [\![1, n]\!]$ et $c \in]\!]0, \prod_{i \in \mathcal{I}_d} m_i [\![$. Alors il existe un entier $x \in [\![0, M[\!]$ et une d-faute d'indices $\mathcal{I}_d$ tels que $\left| \varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) - x \right| = cM_{\mathcal{I}_d}$.*

Démonstration. Considérons la d-faute de résidus $e_i = |cM_i|_{m_i}$ si $i \in \mathcal{I}_d$, et $e_i = 0$ sinon. Il est clair que $\varphi_{\mathcal{B}}^{-1}(\boldsymbol{e}_{\mathcal{B}}) = cM_{\mathcal{I}_d}$. Il suffit alors de considérer que cette faute affecte $x = 0$. □

Les propositions précédentes permettent d'établir le théorème suivant.

Théorème 3.1 *Soit une base RNS $\mathcal{B} = \{m_1, \ldots, m_n\}$, M_R un entier premier à M et d un entier dans $[\![1, n]\!]$. Pour toute d-faute sur des résidus $\boldsymbol{x}_{\mathcal{B}}$, $\varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) \neq x \bmod M_R$, si, et seulement si,*

$$\forall (i_1, \ldots, i_d) \subset [\![1, n]\!], \quad M_R > \prod_{j=1}^{d} m_{i_j}. \tag{3.2}$$

Démonstration. Nous prouvons la suffisance par contraposition. Supposons donc qu'il existe x un entier de l'intervalle dynamique $[\![0, M[\![$ de $\mathcal{B}$, de résidus $\boldsymbol{x}_{\mathcal{B}}$, et une d-faute d'indices $\mathcal{I}_d = \{i_1, \ldots, i_d\}$ affectant $\boldsymbol{x}_{\mathcal{B}}$, et pour laquelle $\varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = x \bmod M_R$. Par la Proposition 3.1, il existe un entier $a_{\mathcal{I}_d}$ vérifiant

$0 < \left|a_{\mathcal{I}_d}\right| < \prod_{i\in\mathcal{I}_d} m_i$ et tel que $\varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = x + a_{\mathcal{I}_d} M_{\mathcal{I}_d} \in [\![0, M[\![$. L'hypothèse $\varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = x \bmod M_R$ implique donc que M_R divise $a_{\mathcal{I}_d} M_{\mathcal{I}_d}$. Comme M_R est premier avec M, et donc avec $M_{\mathcal{I}_d}$, il vient alors que M_R divise $a_{\mathcal{I}_d}$. Ainsi, $M_R < \prod_{i\in\mathcal{I}_d} m_i$, ce qui achève la preuve de la suffisance.

Montrons la nécessité par contraposition. Soit $\mathcal{I}_d$ un ensemble de d indices tels que $M_R < \prod_{i\in\mathcal{I}_d} m_i$. Alors, par la Proposition 3.2, il existe un entier x et une d-faute d'indices $\mathcal{I}_d$ telle que $\left|\varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) - x\right| = M_R M_{\mathcal{I}_d}$. Ainsi, $\varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = x \bmod M_R$. □

La procédure de détection DetectMultErr reste identique à celle utilisée pour la détection des fautes uniques, et se base sur un test de cohérence. Le Théorème 3.1 donne une condition nécessaire sur la taille de la redondance pour la détection des d-fautes. Néanmoins, ce résultat doit être affiné en prenant en compte le type de conversion utilisé pour la procédure de détection, et également en considérant que les résidus redondants peuvent aussi être affectés par des fautes. C'est l'objet du Théorème 3.2.

Procédure DetectMultErr(Bex,$\mathcal{B}$,$\mathcal{B}_R$,$\boldsymbol{x}_{\mathcal{B}}$,$\boldsymbol{x}_R$)

```
1 début
2 |  x̂_R ← Bex(B, B_R, x_B);
3 |  retourner x̂_R == x_R
```

Théorème fondamental de détection pratique d'erreurs multiples

Théorème 3.2 *Soit $\mathcal{B} = \{m_1, \ldots, m_n\}$ une base RNS et $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$ k entiers avec $k \in [\![2, n]\!]$. Soit $x \in [\![0, M[\![$ donné par ses résidus $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R})$.*

Quelle que soit la conversion utilisée dans la procédure de détection DetectMultErr, pour tout $d \in [\![k+1, n+k]\!]$ il existe une d-faute non détectable.

1. *Toute d-faute affectant $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R})$, pour tout $d \in [\![1, k]\!]$, est détectable par la Procédure TestCoherence ou la Procédure DetectMultErr couplée avec la conversion* Bex_{mrs} *si, et seulement si,*

$$\forall m_{R,z} \in \mathcal{B}_R,\ m_{R,z} > \max\{m \mid m \in \mathcal{B}\}. \tag{3.3}$$

2. *Si un modulus supplémentaire m_{sk} est adjoint à la base $\mathcal{B}$ et* Bex_{sk} *est utilisé par la Procédure DetectMultErr, une condition suffisante est donnée par :*

$$\forall m_{R,z} \in \mathcal{B}_R,\ m_{R,z} > \max\{m \mid m \in \mathcal{B} \cup \{m_{sk}\}\}. \tag{3.4}$$

3. *Si toute valeur x à résidus intègres est supposée être dans l'intervalle $[\![0, (1-\alpha_{kw})\, M[\![$ et si la conversion* Bex_{kwc} *est utilisée dans la Procédure DetectMultErr, alors toute d-faute est détectable pour tout $d \in [\![1, k]\!]$ si les Conditions (3.3) du cas* Bex_{mrs} *sont vérifiées.*

Démonstration. Soit $(\overline{x}_{\mathcal{B}}, \overline{x}_{\mathcal{B}_R})$ les résidus issus de ceux de x affectés par une d-faute, et soit $\overline{x} = \varphi_{\mathcal{B}}^{-1}(\overline{x}_{\mathcal{B}}) \in [\![0, M[\![$. Nous rappelons que le test de cohérence est la vérification de l'égalité $\text{Bex}(\overline{x}_{\mathcal{B}}) \stackrel{?}{=} \overline{x}_{\mathcal{B}_R}$, où Bex est une conversion de la base $\mathcal{B}$ vers la base $\mathcal{B}_R$.

Pour tout $d \in [\![k+1, n+k]\!]$, il est possible de construire une d-faute non détectable. Les résidus $(x_{\mathcal{B}}, x_{\mathcal{B}_R})$ sont par hypothèse ceux d'un entier x quelconque de l'intervalle dynamique de $\mathcal{B}$. Soit une faute affectant $d-k$ des résidus principaux $x_{\mathcal{B}}$. En notant $\overline{x} = \text{Bex}(\mathcal{B}, \mathcal{B}_R, \overline{x}_{\mathcal{B}}) \in [\![0, M[\![$, il suffit de considérer la faute modifiant les résidus redondants de la manière suivante : $\overline{x}_{R,i} = |\overline{x}|_{m_{R,i}}$ pour tout $i \in [\![1, k]\!]$. La capacité de détection du RNS redondant est donc inférieure ou égale à k.

1. Cas Bex_{mrs}. C'est le cas le plus direct puisque quels que soient les résidus erronés $\overline{x}_{\mathcal{B}}$ considérés, Bex_{mrs} est toujours une conversion avec réduction complète modulo M. Ainsi, $\text{Bex}_{mrs}(\mathcal{B}, \overline{x}_{\mathcal{B}}) = \varphi_{\mathcal{B}}^{-1}(\overline{x}_{\mathcal{B}})$.

   - Preuve de la suffisance.
     La suffisance peut être montrée par récurrence sur k. Le cas $k = 1$ est prouvé par le Théorème 2.3. Soit $k \in [\![2, n]\!]$ un entier. L'hypothèse de récurrence est la détection de toute d-faute pour $d \leqslant k-1$ sous la Condition (3.3).
     Soit alors une k-faute affectant $k_{\mathcal{B}}$ résidus principaux et k_R résidus redondants avec $k_{\mathcal{B}} + k_R = d+1$, $k_{\mathcal{B}} \leqslant n$ et $k_R \leqslant k$. Il est clair que si $k_{\mathcal{B}} \leqslant k-1$, alors l'hypothèse de récurrence appliquée avec l'ensemble des résidus redondants privé d'un résidu redondant erroné permet de conclure à la détection.
     Il reste donc à considérer le cas où $k_{\mathcal{B}} = k$. Soit une k-faute d'indices $\mathcal{I}_k$. La Condition (3.3) implique notamment que $M_R > \prod_{i \in \mathcal{I}_k} m_i$. Ainsi, le Théorème 3.1 garantit la détection de la faute. La détection est assurée par le test de cohérence.
   - Prouvons la nécessité par contraposition, avec construction d'un contre-exemple.
     Supposons qu'il existe $(z, i) \in [\![1, k]\!] \times [\![1, n]\!]$ tel que $m_{R,z} < m_i$, et soit $x = 0$. Alors, il est possible de construire un ensemble de résidus erronés $(\overline{x}_{\mathcal{B}}, \overline{x}_{\mathcal{B}_R})$ contenant au plus k fautes et qui passera avec succès le test de cohérence. Soit $\overline{x}_t = 0$ pour tout $t \in [\![1, n]\!] \backslash \{i\}$, et $\overline{x}_i = |m_{R,z} M_i|_{m_i} = m_{R,z}$. Alors, en notant une nouvelle fois $\overline{x}$ l'entier $\overline{x} = \varphi_{\mathcal{B}}^{-1}(\overline{x}_{\mathcal{B}})$ de $[\![0, M[\![$ et en notant $a_i = \left|\overline{x}_i M_i^{-1}\right|_{m_i}$, nous avons :

     $$\begin{aligned} \overline{x} &= a_i M_i \\ &= \left|m_{R,z} M_i \times M_i^{-1}\right|_{m_i} \times M_i \\ &= m_{R,z} M_i. \end{aligned}$$

     Nous supposons alors que les résidus redondants erronés sont $\overline{x}_{R,z} = 0$ et $\overline{x}_{R,t} = |m_{R,z} M_i|_{m_{R,t}}$ pour tout $t \neq z$. Par conséquent,

     $$\varphi_{\mathcal{B}_R} \circ \varphi_{\mathcal{B}}^{-1}(\overline{x}_{\mathcal{B}}) = \text{Bex}_{mrs}(\mathcal{B}, \mathcal{B}_R, \overline{x}_{\mathcal{B}}) = \overline{x}_{\mathcal{B}_R}.$$

     Ceci implique que le test de cohérence ne détecte pas la faute multiple donnée.

2. Cas Bex_{sk}. Ici, il est n'est plus garanti que $\text{Bex}_{sk}(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}) = \varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}})$.

   Nous prouvons la suffisance de la condition par récurrence sur k. Le cas $k = 1$ est donné par le Théorème 2.3. Soit donc $k \in [\![2, n]\!]$. Pour les mêmes raisons que précédemment, seul le cas où k fautes affectant les résidus principaux reste à vérifier en détails.

   Dans un premier temps, nous supposons le résidu $\overline{x}_{sk}$ utilisé pour calculer $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ intègre. Nous avons donc :

$$\overline{x} = x + a_{\mathcal{I}_k} M_{\mathcal{I}_k} \in [\![0, M[\![$$

   pour des indices $\mathcal{I}_k \subset [\![1, n]\!]$, et un entier $0 < \left|a_{\mathcal{I}_k}\right| < \prod_{i \in \mathcal{I}_k} m_i$ (cf. Prop. 3.1). Alors la conversion donne :

$$\text{Bex}_{sk}(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}, x_{sk}) = \overline{x} + bM$$

   où b est un entier tel que $|b| < m_{sk}$ et vérifiant également :

$$b \equiv -a_{\mathcal{I}_k} M_{\mathcal{I}_k} \bmod m_{sk}.$$

   En particulier, ceci implique que $\left|\text{Bex}_{sk}(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}, x_{sk}) - x\right|_{m_{sk}} = 0$. Il faut souligner que $\overline{x} + bM \neq x$. En effet, il suffit de voir que, par hypothèse, $\overline{x} + bM \not\equiv x \bmod \prod_{i \in \mathcal{I}_k} m_i$. Finalement, nous avons :

$$0 < \left|\text{Bex}_{sk}(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}, x_{sk}) - x\right| = \left|a_{\mathcal{I}_k} M_{\mathcal{I}_k} + bM\right| = c m_{sk} M_{\mathcal{I}_k}$$

   avec c un entier tel que $0 < c < \prod_{i \in \mathcal{I}_k} m_i < M_R$. Comme M_R est premier avec $m_{sk} M_{\mathcal{I}_k}$, nous avons $c m_{sk} M_{\mathcal{I}_k} \not\equiv 0 \bmod M_R$. La détection est donc assurée.

   Dans un deuxième temps, nous supposons le résidu x_{sk} erroné. Soit donc $\overline{x}_{sk} \neq x_{sk}$, et $\mathcal{I}_{k-1}$ les autres indices des résidus principaux erronés. Nous pouvons alors écrire :

$$\overline{x} = \text{Bex}_{\text{sk}}(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}, \overline{x}_{sk}) = x + a_{\mathcal{I}_{k-1}} M_{\mathcal{I}_{k-1}} - bM$$

   avec $0 < \left|a_{\mathcal{I}_{k-1}}\right| < \prod_{i \in \mathcal{I}_{k-1}} m_i$ et $0 \leqslant b \leqslant m_{sk} - 1$. De plus, nous avons alors :

$$\left| a_{\mathcal{I}_{k-1}} - b \times \prod_{i \in \mathcal{I}_{k-1}} m_i \right| < m_{sk} \times \prod_{i \in \mathcal{I}_{k-1}} m_i < M_R.$$

   Mais nous avons aussi $a_{\mathcal{I}_{k-1}} - b \times \prod_{i \in \mathcal{I}_{k-1}} m_i \neq 0$. En effet, comme par exemple

$$x_{sk} \not\equiv \overline{x}_{sk} \equiv \left(x_{sk} + a_{\mathcal{I}_{k-1}} - b \times \prod_{i \in \mathcal{I}_{k-1}} m_i \right) \bmod m_{sk}$$

   alors $(a_{\mathcal{I}_{k-1}} - b \times \prod_{i \in \mathcal{I}_{k-1}} m_i) \not\equiv 0 \bmod m_{sk}$. Par conséquent,

$$\left|a_{\mathcal{I}_{k-1}} M_{\mathcal{I}_{k-1}} - bM\right|_{M_R} \neq 0$$

   d'où la détection de la faute multiple.

3. Cas Bex_{kwc}. De même, la conversion appliquée à des résidus erronés peut ne pas être complètement réduite.

   Le principe de la preuve est identique au cas Bex_{mrs}. Ainsi, nous avons $\overline{x} = x + a_{\mathcal{I}_k} M_{\mathcal{I}_k}$ avec $0 < \left|a_{\mathcal{I}_k}\right| < \prod_{i \in \mathcal{I}_k} m_i$. Mais dans ce cas nous avons désormais :

$$\text{Bex}_{kwc}(\mathcal{B}, \overline{\boldsymbol{x}}_{\mathcal{B}}, \alpha_{kw}) - x = \overline{x} - \delta M - x = a_{\mathcal{I}_k} M_{\mathcal{I}_k} - \delta M$$

   avec $\delta \in \{0, 1\}$. De plus, vu la Remarque 1.9 (p. 29), et comme par hypothèse $x \in [\![0, (1 - \alpha_{kw})\, M[\![$, les implications suivantes sont vérifiées :

$$\delta = 1 \Rightarrow \overline{x} = \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{x}}_{\mathcal{B}}\right) \in [\![(1 - \alpha_{kw})\, M, M[\![\Rightarrow \overline{x} > x \Rightarrow a_{\mathcal{I}_k} > 0.$$

   Ainsi, nous en déduisons que

$$0 < \left| a_{\mathcal{I}_k} - \delta \prod_{i \in \mathcal{I}_k} m_i \right| < \prod_{i \in \mathcal{I}_k} m_i < M_R$$

   d'où la détection de la faute.

□

Remarque 3.1 *Le Théorème 3.2, tout comme le Théorème 2.3 , montre que l'utilisation d'une conversion* Bex_{sk} *pour la détection des fautes multiples reste foncièrement similaire à l'utilisation d'une conversion* Bex_{mrs}*. En effet, les Hypothèses (3.4) s'écrivent de manière identique aux Hypothèses (3.3) lorsque le modulus* m_{sk} *est considéré de la même manière que tout autre modulus* m_i *de* $\mathcal{B}$*.*

Comme pour la détection des fautes uniques, les opérations arithmétiques RNS basiques dans un corps fini $\mathbb{F}_p$ sont protégées contre l'injection de fautes multiples aussi longtemps qu'aucun dépassement de capacité n'intervient, c'est-à-dire tant que les données traitées et intègres appartiennent à l'intervalle légitime $[\![0, M[\![$. Dès qu'une réduction modulo p est effectuée, la cohérence entre résidus principaux et redondants peut être brisée à cause de la difficulté à obtenir un résidu redondant de $\mathbf{q}_{\mathcal{B}}$ dans l'Algorithme 8 de réduction modulaire RNS. L'Algorithme 13 de multiplication modulaire en RNS redondant fournit une solution efficace pour conserver la capacité de détection des fautes uniques des RNS redondants malgré la présence de la réduction modulaire.

Dans le contexte des fautes multiples, l'apparition conjointe d'erreurs de catégories 1, 2, 4 et 5 peut surgir. Cependant, nous allons prouver que le schéma général de l'Algorithme 13 permet toujours de détecter ce genre de fautes multiples.

3.1.2 Application à la multiplication modulaire RNS

L'intégration et l'utilisation de la redondance pour la détection des fautes multiples se font de manière identique à celles concernant la détection des fautes uniques. Les conditions sur les bases RNS $\mathcal{B}$, $\mathcal{B}'$ et $\mathcal{B}_R$ ainsi que sur les tailles des entrées restent identiques à celles de l'Algorithme 8 de multiplication modulaire RNS standard, et sont résumées dans la Table 1.1 (p. 34).

Localisation des fautes

Dans le contexte de la multiplication modulaire, le modèle de faute multiple est construit sur les catégories introduites dans la Partie 2.3.2. La discussion menée en prélude à la Proposition 2.8 (p. 72) et concernant les fautes de catégorie 4 affectant le Cox reste valable. Chaque Rower sera supposé posséder son propre Cox. Une discussion concernant le cas d'un seul Cox sera cependant menée. La détection des fautes qui modifient en partie les valeurs calculées par le(s) Cox sera étudiée par la suite.

La capacité de détection choisie, fixée à un entier k, donne la taille de la base redondante $\mathcal{B}_R$. Pour tout $d \leqslant k$, la structure d'une d-faute injectée durant une multiplication modulaire est décrite par la définition suivante.

Définition 3.2 *Une d-faute affectant l'opération de multiplication modulaire RNS est la donnée de $d = u + v + w$ indices $\mathcal{I}_u = \{i_1, \ldots, i_u\} \subset [\![1, n]\!]$, $\mathcal{J}_v = \{j_1, \ldots, j_v\} \subset [\![1, \ell(+1)]\!]$ (si m_{sk} est adjoint à $\mathcal{B}'$, il pourra être noté $m_{\ell+1}$) et $\mathcal{Z}_w = \{z_1, \ldots, z_w\} \subset [\![1, k]\!]$, et de d nombres $e_i \in \mathbb{Z}/m_i\mathbb{Z}$ pour tout $i \in \mathcal{I}_u$, $e'_j \in \mathbb{Z}/m'_j\mathbb{Z}$ pour tout $j \in \mathcal{J}_v$, et $e_{R,z} \in \mathbb{Z}/m_{R,z}\mathbb{Z}$ pour tout $z \in \mathcal{Z}_w$. Si u (resp. v, w) est nul, alors par définition $\mathcal{I}_u = \varnothing$ (resp. $\mathcal{J}_v = \varnothing$, $\mathcal{Z}_w = \varnothing$). Une telle d-faute modifie les résidus $\boldsymbol{q}_{\mathcal{B}}$, $\boldsymbol{s}_{\mathcal{B}'}$ et $\boldsymbol{s}_{\mathcal{B}_R}$ calculés lors de l'exécution de l'Algorithme 13 de la manière suivante :*

$$\begin{cases} \forall i \in \mathcal{I}_u,\ \overline{q}_i = |q_i + e_i|_{m_i}, \\ \forall j \in \mathcal{J}_v,\ \overline{s}'_j = \left|s'_j + e'_j\right|_{m'_j}, \\ \forall z \in \mathcal{Z}_w,\ \overline{s}_{R,z} = |s_{R,z} + e_{R,z}|_{m_{R,z}}. \end{cases} \tag{3.5}$$

Une telle faute contient donc u fautes de catégorie 1, v fautes de catégorie 2 (et éventuellement de catégorie 4 s'il y a présence d'un modulus m_{sk}) et w fautes de catégorie 5.

L'Hypothèse 2.2 (p. 62) reste valable afin d'appuyer la pertinence de ce modèle de faute malgré la présence des opérations de conversion de base.

Algorithme proposé et preuve de correction

Théorème 3.3 *Soit $\mathcal{H}$ un ensemble d'hypothèses de la Table 1.1 (p. 34). Soit $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$ une base redondante première à $\mathcal{B}$ et $\mathcal{B}'$ vérifiant les conditions suivantes :*

$$\forall \left(m, m', m_R\right) \in \mathcal{B} \times \mathcal{B}' \times \mathcal{B}_R, m_R > \begin{cases} m, m' \text{ si } \mathcal{H} \in \{\mathcal{H}_{mrs}, \mathcal{H}_{kw}\}, \\ m_{sk}m, m' \text{ si } \mathcal{H} = \mathcal{H}_{sk}. \end{cases} \tag{3.6}$$

Alors pour tout $d \in [\![1, k]\!]$, l'Algorithme 14 détecte toute d-faute telle que décrite dans la Définition 3.2.

Démonstration. Soit une d-faute d'indices $\mathcal{I}_u = \{i_1, \ldots, i_u\}$, $\mathcal{J}_v = \{j_1, \ldots, j_v\}$ et $\mathcal{Z}_w = \{z_1, \ldots, z_w\}$, avec $d = u + v + w$, et définie par les Équations (3.5). En prélude à la preuve du théorème, nous énumérons quelques notations et constatations préliminaires.

- Effet des fautes dans $\mathcal{B}$ affectant $\boldsymbol{q}_{\mathcal{B}}$: par la Proposition 3.1,

$$\overline{q} = \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) = q + e_{\mathcal{I}_u} M_{\mathcal{I}_u} \in [\![0, M[\![,$$

Algorithme 14 : MulModRRNS $(\mathcal{B}, \mathcal{B}', \mathcal{B}_R, x, y, p)$

Données :

- trois bases copremières $\mathcal{B} = \{m_1, \ldots, m_n\}$, $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$ et $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$
- deux entiers x, y représentés par leurs résidus $(\boldsymbol{x}_\mathcal{B}, \boldsymbol{x}_{\mathcal{B}'}, \boldsymbol{x}_{\mathcal{B}_R})$ et $(\boldsymbol{y}_\mathcal{B}, \boldsymbol{y}_{\mathcal{B}'}, \boldsymbol{y}_{\mathcal{B}_R})$ dans les trois bases
- un modulus p représenté par ses résidus précalculés $\boldsymbol{p}_{\mathcal{B}' \cup \mathcal{B}_R}$ dans la base $\mathcal{B}' \cup \mathcal{B}_R$
- les résidus précalculés de $\left|-p^{-1}\right|_M$ dans $\mathcal{B}$ et de $\left|M^{-1}\right|_{M' M_R}$ dans $\mathcal{B}' \cup \mathcal{B}_R$
- deux procédures de conversion de base Bex_1 et Bex_2
- toutes ces données vérifient également les hypothèses de la Table 1.1

Résultat : les résidus de $s \equiv xyM^{-1} \bmod p$ dans $\mathcal{B} \cup \mathcal{B}' \cup \mathcal{B}_R$

1 **début**
2 **pour** $i \leftarrow 1$ **à** n **faire**
/* en ∥ dans les canaux de $\mathcal{B}$ */
3 $q_i \leftarrow x_i \times y_i \times \left|-p^{-1}\right|_{m_i} \bmod m_i$
4 $\left(\widehat{q}'_1, \ldots, \widehat{q}'_\ell, \widehat{q}_{R,1}, \ldots, \widehat{q}_{R,k}\right) \leftarrow \text{Bex}_1\left(\mathcal{B}, \mathcal{B}' \cup \mathcal{B}_R, (q_1, \ldots, q_n)\right)$
/* conversion de $\mathcal{B}$ vers $\mathcal{B}' \cup \mathcal{B}_R$ */
5 **pour** $j \leftarrow 1$ **à** ℓ **faire**
6 $t'_j \leftarrow x'_j \times y'_j + \widehat{q}'_j \times p'_j \bmod m'_j$
7 **pour** $z \leftarrow 1$ **à** k **faire**
8 $t_{R,z} \leftarrow x_{R,z} \times y_{R,z} + \widehat{q}_{R,z} \times p_{R,z} \bmod m_{R,z}$
/* en ∥ dans les canaux de $\mathcal{B}' \cup \mathcal{B}_R$ */
9 **pour** $j \leftarrow 1$ **à** ℓ **faire**
10 $s'_j \leftarrow t'_j \times \left|M^{-1}\right|_{m'_j} \bmod m'_j$
11 **pour** $z \leftarrow 1$ **à** k **faire**
12 $s_{R,z} \leftarrow t_{R,z} \times \left|M^{-1}\right|_{m_{R,z}} \bmod m_{R,z}$
/* en ∥ dans les canaux de $\mathcal{B}' \cup \mathcal{B}_R$ */
13 $(s_1, \ldots, s_n, \widehat{s}_{R,1}, \ldots, \widehat{s}_{R,k}) \leftarrow \text{Bex}_2\left(\mathcal{B}', \mathcal{B} \cup \mathcal{B}, \left(s'_1, \ldots, s'_\ell\right)\right)$
/* conversion de $\mathcal{B}'$ vers $\mathcal{B} \cup \mathcal{B}_R$ */
14 **si** $\boldsymbol{s}_{\mathcal{B}_R} \neq \widehat{\boldsymbol{s}}_{\mathcal{B}_R}$ **alors**
/* procédure de détection */
15 **retourner** `Faute détectée.`
16 **sinon**
17 **retourner** $\left(s_1, \ldots, s_n, s'_1, \ldots, s'_\ell, s_{R,1}, \ldots, s_{R,k}\right)$

où $|e_{\mathcal{I}_u}| < \prod_{i\in\mathcal{I}_u} m_i$. La valeur retournée par la conversion est :

$$\widehat{q} = \mathrm{Bex}_1\left(\mathcal{B}, \overline{\boldsymbol{q}}_{\mathcal{B}}\right) = \overline{q} + \delta M$$

où la valeur de δ dépend de la conversion utilisée.

Soit $t = xy + \widehat{q}p$. Alors les conditions sur les tailles de $\mathcal{B}$ et $\mathcal{B}'$ garantissent que $0 \leqslant t < MM'$.

Précisons cela dans le cas des hypothèses $\mathcal{H}_{kw}$. Nous rappelons que ces hypothèses donnent $xy < (1-\Delta_{kw})\,Mp$, et que toute valeur convertie par Bex_{kw} est inférieure à $(1+\Delta_{kw})\,M$ (cf. Remarque 1.8, p. 29). Ainsi, $\widehat{q} = \overline{q} + \delta M < (1+\Delta_{kw})\,M$ où $\overline{q}$ est par définition dans $[\![0, M[\![$, et $\delta \in \{0,1\}$. Par conséquent,

$$0 \leqslant t = xy + \widehat{q}p < (1-\Delta_{kw})\,Mp + (1+\Delta_{kw})\,Mp = 2Mp.$$

Ainsi, t est complètement déterminé par ses résidus $(\boldsymbol{t}_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'})$ dans la base étendue $\mathcal{B}\cup\mathcal{B}'$. Lorsque $u = 0$, alors $|t|_M = 0$. Dans le cas contraire, $|t|_M = a_{\mathcal{I}_u} M_{\mathcal{I}_u}$ avec $0 < a_{\mathcal{I}_u} = \left|e_{\mathcal{I}_u} M_{\mathcal{I}_u} p\right|_{\prod_{i\in\mathcal{I}_u} m_i} < \prod_{i\in\mathcal{I}_u} m_i$, et $a_{\mathcal{I}_u} = 0$ si, seulement si, $u = 0$. Plus précisément,

$$\begin{aligned} |t|_M &= a_{\mathcal{I}_u} M_{\mathcal{I}_u} \\ &= \sum_{i\in\mathcal{I}_u} \left|e_i p M_i^{-1}\right|_{m_i} M_i \bmod M \\ &= \varphi_{\mathcal{B}}^{-1}\left(\left(0,\ldots,0,\left|e_{i_1}p\right|_{m_{i_1}},0,\ldots,0,\left|e_{i_u}p\right|_{m_{i_u}}0,\ldots,0\right)_{\mathcal{B}}\right). \end{aligned} \tag{3.7}$$

- Effet des fautes dans $\mathcal{B}'$ affectant $\boldsymbol{s}_{\mathcal{B}'}$: comme précédemment, les notations suivantes sont introduites :

$$\begin{cases} \overline{s} = \left|tM^{-1}\right|_{M'} + b_{\mathcal{J}_v} M'_{\mathcal{J}_v} \in [\![0, M'[\![, \\ b_{\mathcal{J}_v} \equiv \sum_{j\in\mathcal{J}_v} \left|e'_j M_j^{-1}\right|_{m'_j} M'_j \bmod M, \\ 0 < |b_{\mathcal{J}_v}| < \prod_{j\in\mathcal{J}_v} m'_j, \text{ et } b_{\mathcal{J}_v} = 0 \Leftrightarrow v = 0. \end{cases}$$

- Effet des fautes dans $\mathcal{B}_R$ affectant $\boldsymbol{s}_{\mathcal{B}_R}$:

$$\begin{cases} \overline{s}_R = \left|tM^{-1}\right|_{M_R} + c_{\mathcal{Z}_w} M_{R,\mathcal{Z}_w} \in [\![0, M_R[\![, \\ c_{\mathcal{Z}_w} \equiv \sum_{z\in\mathcal{Z}_w} \left|e_{R,z} M_{R,z}^{-1}\right|_{m_{R,z}} M_{R,z} \bmod M_R \\ 0 < |c_{\mathcal{Z}_w}| < \prod_{z\in\mathcal{Z}_w} m_{R,z}, \text{ et } c_{\mathcal{Z}_w} = 0 \Leftrightarrow w = 0. \end{cases}$$

Le test de cohérence vérifie la nullité de la quantité $\left|\mathrm{Bex}_2\left(\mathcal{B}', \overline{\boldsymbol{s}}_{\mathcal{B}'}\right) - \overline{s}_R\right|_{M_R}$. Par conséquent, il suffit de montrer l'implication suivante pour prouver que la détection réussit :

$$d = u + v + w \neq 0 \Rightarrow \left|\mathrm{Bex}_2\left(\mathcal{B}', \overline{\boldsymbol{s}}_{\mathcal{B}'}\right) - \overline{s}_R\right|_{M_R} \neq 0.$$

Cette quantité est nulle sous les hypothèses données par la Table 1.1 lorsque $d = 0$. Comme M est premier avec M_R, il revient au même de montrer l'implication suivante :

$$d = u + v + w \neq 0 \Rightarrow \left|M\left(\text{Bex}_2\left(\mathcal{B}', \overline{\boldsymbol{s}}_{\mathcal{B}'}\right) - \overline{s}_R\right)\right|_{M_R} \neq 0.$$

La conversion Bex_2 est complètement réduite dès que $d = 0$. De plus, c'est aussi le cas lorsque $d \neq 0$ et $\text{Bex}_2 = \text{Bex}_{mrs}$. Par conséquent, $\text{Bex}_{mrs}\left(\mathcal{B}', \overline{\boldsymbol{s}}_{\mathcal{B}'}\right) = \overline{s}$. Si Bex_{kwc} ou Bex_{sk} sont utilisées, alors $\text{Bex}_2\left(\mathcal{B}', \overline{\boldsymbol{s}}_{\mathcal{B}'}\right) = \overline{s} + \mu M'$. Les valeurs possibles de μ seront détaillées par la suite.

Pour récapituler, nous avons d'une part :

$$\begin{aligned} |M \times \overline{s}_R|_{M_R} &= \left|M \times \left|tM^{-1}\right|_{M_R} + c_{\mathcal{Z}_w} M_{R,\mathcal{Z}_w} M\right|_{M_R} \\ &= \left|t + c_{\mathcal{Z}_w} M_{R,\mathcal{Z}_w} M\right|_{M_R} \end{aligned} \tag{3.8}$$

et d'autre part :

$$M \times \text{Bex}_2\left(\mathcal{B}', \overline{\boldsymbol{s}}_{\mathcal{B}'}\right) = M \times \left|tM^{-1}\right|_{M'} + b_{\mathcal{J}_v} M'_{\mathcal{J}_v} M + \mu M M'. \tag{3.9}$$

Nous rappelons qu'il s'agit de montrer que si $d \neq 0$ alors

$$\left|M \times \overline{s}_R - M \times \text{Bex}_2\left(\mathcal{B}', \overline{\boldsymbol{s}}_{\mathcal{B}'}\right)\right|_{M_R} \neq 0.$$

Étant donné que $M \times \left|tM^{-1}\right|_{M'} < MM'$, cette quantité est définie par ses résidus dans $\mathcal{B} \cup \mathcal{B}'$. Plus précisément :

$$M \times \left|tM^{-1}\right|_{M'} = \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'}\right).$$

Par conséquent il suffit de vérifier que si $d \neq 0$ alors :

$$\left|t + c_{\mathcal{Z}_w} M_{R,\mathcal{Z}_w} M - \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'}\right) - b_{\mathcal{J}_v} M'_{\mathcal{J}_v} M - \mu M M'\right|_{M_R} \neq 0.$$

En utilisant l'Équation (3.7) il vient :

$$\begin{aligned} \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{t}_{\mathcal{B}}, \mathbf{0}_{\mathcal{B}'}\right) &= M' \left|tM'^{-1}\right|_M \\ &= M' \left|a_{\mathcal{I}_u} M_{\mathcal{I}_u} M'^{-1}\right|_M \\ &= M' \varphi^{-1}_{\mathcal{B}}\left(\left(0, \ldots, 0, \left|e_{i_1} p M'^{-1}\right|_{m_{i_1}}, 0, \ldots, 0, \left|e_{i_u} p M'^{-1}\right|_{m_{i_u}} 0, \ldots, 0\right)_{\mathcal{B}}\right) \\ &= \tilde{a}_{\mathcal{I}_u} M_{\mathcal{I}_u} M' \end{aligned}$$

où $\tilde{a}_{\mathcal{I}_u} = \left|a_{\mathcal{I}_u} M^{-1}_{\mathcal{I}_u} M'^{-1}\right|_{\prod_{i\in\mathcal{I}_u} m_i} \neq 0 \Leftrightarrow a_{\mathcal{I}_u} \neq 0 \Leftrightarrow u \neq 0.$

Une nouvelle notation est introduite. Si $\beta \in \{0, 1\}$ est tel que

$$t = \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{t}_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'}\right) = \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{t}_{\mathcal{B}}, \mathbf{0}_{\mathcal{B}'}\right) + \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{t}_{\mathcal{B}'}\right) - \beta M M' \in [\![0, MM'[\![$$

alors l'entier χ est défini par :

$$\begin{aligned}\chi &= \frac{1}{M_{\mathcal{I}_u} M'_{\mathcal{J}_v}} \left(t - M \times \text{Bex}_2 \left(\mathcal{B}', \bar{\boldsymbol{s}}_{\mathcal{B}'}\right)\right) \\ &= \frac{1}{M_{\mathcal{I}_u} M'_{\mathcal{J}_v}} \left(\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{t}_{\mathcal{B}}, \boldsymbol{0}_{\mathcal{B}'}\right) - b_{\mathcal{J}_v} M'_{\mathcal{J}_v} M - (\mu + \beta)\, MM'\right) \\ &= \tilde{a}_{\mathcal{I}_u} \prod_{j\in\mathcal{J}_v} m'_j - b_{\mathcal{J}_v} \prod_{i\in\mathcal{I}_u} m_i - (\beta + \mu) \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j.\end{aligned} \tag{3.10}$$

En particulier, χ vérifie donc :

$$\chi + \mu \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j = \frac{1}{M_{\mathcal{I}_u} M'_{\mathcal{J}_v}} \left(t - M\bar{s}\right) \in \left]\!\right] - \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j, \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j \left[\!\right],$$

et comme $u + v \neq 0$ implique $\tilde{a}_{\mathcal{I}_u} \neq 0$ ou $b_{j\in\mathcal{J}_v} \neq 0$, alors $\chi \neq 0$.

Finalement, nous avons :

$$\chi \bmod \prod_{z\notin\mathcal{Z}_w} m_{R,z} \not\equiv 0 \Rightarrow \text{Faute détectée.} \tag{3.11}$$

Il nous suffit donc de prouver que $\chi \bmod \prod_{z\notin\mathcal{Z}_w} m_{R,z} \not\equiv 0$.

Les différents scenarii possibles sont examinés dès à présent.

- $\text{Bex}_2 = \text{Bex}_{mrs}$: dans ce cas $\mu = 0$. Donc $0 < |\chi| < \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j$. De plus, étant donné les conditions sur les moduli redondants et comme $k - w \geqslant u + v$,
$$\prod_{z\notin\mathcal{Z}_w} m_{R,z} > \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j.$$
Ceci garantit que $|\chi| \prod_{z\notin\mathcal{Z}_w} m_{R,z} \neq 0$ et donc la détection de la faute multiple.

- $\text{Bex}_2 = \text{Bex}_{sk}$: les fautes dans $\mathcal{B}' \cup \{m_{sk}\}$ peuvent impliquer que le résidu s_{sk} ne soit plus une information redondante des résidus $\boldsymbol{s}_{\mathcal{B}'}$. Autrement dit, les résidus erronés $\boldsymbol{s}_{\mathcal{B}'\cup\{m_{sk}\}}$ peuvent représenter n'importe quel entier de l'intervalle $[\![0, m_{sk}M'[\![$. Ainsi, μ peut prendre une valeur quelconque dans $\left]\!\right] - m_{sk}, m_{sk}\left[\!\right]$. Par conséquent,
$$0 < |\chi| < m_{sk} \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j.$$
Comme $u + v \leqslant k - w$, $\prod_{z\notin\mathcal{Z}_w} m_{R,z} > m_{sk} \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j$, ce qui implique l'inégalité attendue $|\chi| \prod_{z\notin\mathcal{Z}_w} m_{R,z} \neq 0$.

- $\text{Bex}_2 = \text{Bex}_{kwc}$: aucune faute n'est supposée affecter directement le Cox durant les deux conversions Bex_1 et Bex_2. $\mu = 0$ implique que $0 < |\chi| < \prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j$. Ensuite, les résidus erronés dans $\mathcal{B}'$ peuvent aussi impliquer $\mu = -1$. Ce cas n'apparaît que si $\bar{s} \in [\![(1 - \alpha_{kw})\, M', M'[\![$. Comme $0 \leqslant t < (1 - \alpha_{kw})\, MM'$, il vient immédiatement que :
$$-\prod_{i\in\mathcal{I}_u} m_i \prod_{j\in\mathcal{J}_v} m'_j < \chi < 0.$$

Ainsi, dans les deux cas $\mu = 0$ ou $\mu = -1$, les inégalités suivantes sont vérifiées :

$$0 < |\chi| < \prod_{i \in \mathcal{I}_u} m_i \prod_{j \in \mathcal{J}_v} m'_j.$$

Et donc $|\chi|_{\prod_{z \notin \mathcal{Z}_w} m_{R,z}} \neq 0$, ce qui achève la preuve.

□

Stratégies concernant les fautes affectant le(s) Cox

Cas d'un Cox général Si une ou plusieurs des fautes de la faute multiple considérée modifient de manière transitoire le registre d'accumulation du Cox durant la seconde conversion de base Bex_{kwc}, l'effet peut être modélisé par une modification de la valeur $\kappa_{\mathcal{B}'}(\overline{s}_{\mathcal{B}'})$ en ajoutant ou soustrayant un entier μ. μ est borné en valeur absolue par un entier τ qui décrit le nombre maximal de fois où le registre d'accumulation du Cox peut être modifié durant une attaque. Ainsi μ est une valeur de $[\![-\tau - 1, \tau]\!]$. L'effet sur l'entier χ introduit dans la preuve du Théorème 3.3 est résumé par les inégalités suivantes :

$$|\chi| < (\tau + 1) \prod_{i \in \mathcal{I}_u} m_i \prod_{j \in \mathcal{J}_v} m'_j.$$

Par hypothèse sur la faute multiple, $u + v + \tau \leqslant k - w$. Ainsi, tant que les moduli redondants sont supérieurs à $\tau \times \max(m \in \mathcal{B} \cup \mathcal{B}')$, les fautes affectant le Cox ne modifient pas les hypothèses $\mathcal{H}_{kw}$.

De telles fautes intervenant aussi durant la première conversion de base Bex_{kw} impliquent l'obtention de la valeur $\widehat{q} = \overline{q} + \delta M$ avec $\delta \in [\![-\tau, \tau + 1]\!]$. Une manière de contrer ce type de faute est d'ajouter systématiquement $+\tau M$ aux résidus calculés par Bex_1 dans $\mathcal{B}' \cup \mathcal{B}_R$. Afin d'éviter tout dépassement de capacité de la valeur $t = xy + \widehat{q}p$ dans la base étendue $\mathcal{B} \cup \mathcal{B}'$, il convient d'augmenter la taille de M' en imposant la condition $(\tau + 2)\, p < (1 - \alpha_{kw})M'$. Cependant, si aucune autre faute que celles-ci n'est injectée, le test de cohérence ne détectera rien. Ceci n'est pas un problème puisque le résultat final s reste bien congru modulo p à $\left|xyM^{-1}\right|_p$. Mais alors la sortie s de la multiplication modulaire est cette fois dans l'intervalle $[\![0, (\tau + 2)\, p[\![$. Dans le but de conserver une cohérence entre les tailles d'entrées et sorties de l'algorithme, il faut donc augmenter la taille de M de manière à avoir $(\tau + 2)^2\, p < (1 - \Delta_{kw})M$.

Cas d'un Cox par Rower De la même manière que pour le cas d'une faute unique, les fautes affectant les Cox durant la première conversion ont pour effet de modifier certains résidus de $\boldsymbol{s}_{\mathcal{B}'}$. Or, ceux-ci étant réduits modulo les m'_j avant la seconde conversion, l'effet est identique à celui de fautes de catégorie 2. Dans le cas des perturbations apparaissant durant la seconde conversion, l'effet se réduit de manière similaire à celui de fautes de catégorie 3 et/ou 5 qui seront donc détectées.

Surcoût dû à la procédure de détection

Le surcout engendré par l'introduction de k canaux redondants est simplement k fois le surcoût présenté dans la Table 2.1 (p. 74). Néanmoins, la parallé-

lisation permet une intégration optimale des canaux redondants de manière à éviter toute majoration du temps d'exécution de la multiplication modulaire.

Le problème de la gestion des fautes multiples de catégorie 3 est identique au cas de la détection des fautes uniques de catégorie 3. Il s'agit donc d'effectuer une conversion de base $\mathrm{Bex}(\mathcal{B}, \mathcal{B}_R, \boldsymbol{s}_{\mathcal{B}})$, lorsque la sortie s n'est plus supposée être réutilisée comme entrée de la multiplication modulaire redondante par exemple. Le coût en termes d'opérations élémentaires est résumé dans la Table 3.1, où les moduli de $\mathcal{B}$ et $\mathcal{B}_R$ sont supposés de même taille.

hypothèses	$\mathcal{H}_{mrs}$	$\mathcal{H}_{sk}$	$\mathcal{H}_{kw}$
Mult.	$\frac{(2k+n)(n-1)}{2}$ MMEl	$(kn+k+n)$ MMEl $+(n+1)$ $\mathtt{Mul}_{\log_2(n)}$	$(k+1)n$ MMEl
Add.	$\frac{(2k+n)(n-1)}{2}$ AMEl	kn AMEl $+n$ $\mathtt{Add}_{\log_2(n)}$ $+n$ $\mathtt{Add}_h$	$k(2n-2)$ AMEl

TABLE 3.1 – *Surcoût de la procédure de détection des fautes multiples de catégorie 3 en termes d'opérations élémentaires.*

3.1.3 Adaptation à l'architecture Cox-Rower

Le contexte est celui de la Partie 2.4. L'architecture possède un Cox général, qui peut être protégé par redondance, et l'effet des fautes matérielles comme décrites par la Définition 2.5 est étudié. Deux bases principales $\mathcal{B} = \{m_1, \ldots, m_n\}$ et $\mathcal{B}' = \{m'_1, \ldots, m'_\ell\}$ et une base redondante $\mathcal{B}_R$ sont données. Elles sont supposées vérifier initialement les conditions $\mathcal{H}_{kw}$ de la Table 1.1, mais ces conditions vont être affinées au modèle de faute matérielle.

Une d-faute multiple, pour un certain $d \in [\![1, k]\!]$, se décomposant notamment en u fautes de catégorie 1 et v fautes de catégorie 2, est considérée. Comme souligné dans la discussion précédente, les fautes sur les Cox se réduisent à celles de catégorie 1 et 2. Nous allons suivre l'exécution de la réduction modulaire, en relevant au fur et à mesure les effets des fautes matérielles.

Effet des fautes de catégorie 1

Les fautes de catégorie 1 sont étudiées via les valeurs perturbées $\overline{\xi}_{i_j,q,\mathcal{B}} = \xi_{i_j,q,\mathcal{B}} + e_{i_j}$, où pour tout $j \in [\![1, u]\!]$, $e_{i_j} \in [\![-\xi_{i_j,q,\mathcal{B}}, 2^r - \xi_{i_j,q,\mathcal{B}}[\![$. Si certaines valeurs perturbées vérifient $\overline{\xi}_{i_j,q,\mathcal{B}} \geqslant m_{i_j}$, ces erreurs peuvent provoquer l'apparition d'un multiple de M dans la somme $\mathrm{sum}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) = \sum_{i=1}^{n} \overline{\xi}_{i,q,\mathcal{B}} M_i$. Pour contrer ce phénomène, il est toujours possible d'augmenter la taille de M' pour avoir $M'(1-\alpha_{kw}) > (k+2)p$ afin d'empêcher tout dépassement de capacité. Ensuite, si une valeur $\overline{\xi}_{i_j,q,\mathcal{B}}$ vérifie $\overline{\xi}_{i_j,q,\mathcal{B}} \not\equiv \xi_{i_j,q,\mathcal{B}} \bmod m_i$, alors le modèle théorique est applicable et garantit la détection.

Cependant, nous allons voir que l'effet de ces dépassements de capacité multiples reste aisément gérable.

Nous considérons donc u fautes sur les coefficients $\xi_{i,q,\mathcal{B}}$ indexés par $\mathcal{I}_u$. Parmi ces coefficients erronés, un certain nombre peut, nous l'avons dit, être plus grand que le modulus associé. Nous notons alors $\tilde{\xi}_{i_j,q,\mathcal{B}} = \overline{\xi}_{i_j,q,\mathcal{B}} - m_{i_j}$ si $\overline{\xi}_{i_j,q,\mathcal{B}} \geqslant m_i$, et $\tilde{\xi}_{i_j,q,\mathcal{B}} = \overline{\xi}_{i_j,q,\mathcal{B}}$ sinon. Par conséquent, chaque $\tilde{\xi}_{i_j,q,\mathcal{B}}$ est dans $[\![0, m_i[\![$, et l'ensemble de ces n coefficients correspond à ceux, $\xi_{i,\tilde{q},\mathcal{B}}$, d'un entier

$\tilde{q}$ de $[\![0, M[\![$. Les résidus $\tilde{\boldsymbol{q}}_{\mathcal{B}}$ de $\tilde{q}$ dans $\mathcal{B}$ diffèrent de ceux de q pour tout ou partie des indices $\mathcal{I}_u$.

Par suite, nous obtenons :

$$\sum_{i=1}^{n} \overline{\xi}_{i,q,\mathcal{B}} M_i = \sum_{i=1}^{n} \tilde{\xi}_{i,q,\mathcal{B}} M_i + aM = \sum_{i=1}^{n} \xi_{i,\tilde{q},\mathcal{B}} M_i + aM = \mathrm{sum}_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + aM. \quad (3.12)$$

L'entier a correspond au nombre de dépassements de capacité. En particulier, $a \leqslant u \leqslant k$. Nous notons $i_1, \ldots, i_a$ les indices de ces dépassements de capacité ($\{i_1, \ldots, i_a\} \subseteq \mathcal{I}_u$).

Nous allons trouver des conditions pour lesquelles la valeur $\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right)$ calculée par le Cox corrige au mieux le terme $+aM$ apparaissant dans l'Équation (3.12). Concrètement, nous allons chercher dans quelle mesure nous pouvons obtenir :

$$\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) = \kappa_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + a - \delta,\ \delta \in \{-1, 0, 1\}. \quad (3.13)$$

Les faits et notations suivants vont être utilisés :

$$\begin{cases} m_{i_j} = 2^r - c_{i_j}, \\ \tau_{i_j,\mathcal{B}} = \dfrac{c_{i_j} + \left|2^r - c_{i_j}\right|_{2^{r-h}}}{2^r}, \\ \mathrm{eval}_h\left(m_{i_j}\right) = \dfrac{2^r - c_{i_j} - \left|2^r - c_{i_j}\right|_{2^{r-h}}}{2^r} = 1 - \tau_{i_j,\mathcal{B}}, \\ \mathrm{eval}_h\left(\xi_{i_j,q,\mathcal{B}} + m_{i_j}\right) = \mathrm{eval}_h\left(\xi_{i_j,q,\mathcal{B}}\right) + \mathrm{eval}_h\left(m_{i_j}\right) + \dfrac{\delta_{i_j}}{2^h},\ \delta_{i_j} \in \{0, 1\}. \end{cases}$$

Nous avons alors les inégalités suivantes :

$$\sum_{i=1}^{n} \mathrm{eval}_h\left(\xi_{i,\tilde{q},\mathcal{B}}\right) + a - \sum_{j=1}^{a} \tau_{i_j,\mathcal{B}} \leqslant \sum_{i=1}^{n} \mathrm{eval}_h\left(\overline{\xi}_{i,q,\mathcal{B}}\right) < \sum_{i=1}^{n} \mathrm{eval}_h\left(\xi_{i,\tilde{q},\mathcal{B}}\right) + a - \sum_{j=1}^{a} \tau_{i_j,\mathcal{B}} + \frac{a}{2^h}.$$

De plus, nous rappelons que :

$$\begin{cases} \displaystyle\sum_{i=1}^{n} \mathrm{eval}_h\left(\xi_{i,\tilde{q},\mathcal{B}}\right) \geqslant \kappa_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + \frac{\tilde{q}}{M} - \Delta_{kw}^{(1)} \\ \displaystyle\sum_{i=1}^{n} \mathrm{eval}_h\left(\xi_{i,\tilde{q},\mathcal{B}}\right) \leqslant \kappa_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + \frac{\tilde{q}}{M} \end{cases}$$

où $\Delta_{kw}^{(1)}$ est un majorant de l'erreur totale commise par les approximations de la première conversion.

Par conséquent, nous obtenons finalement les inégalités suivantes sur la somme calculée dans le Cox :

$$\begin{cases} \displaystyle\sum_{i=1}^{n} \mathrm{eval}_h\left(\overline{\xi}_{i,q,\mathcal{B}}\right) \geqslant \kappa_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + \frac{\tilde{q}}{M} - \Delta_{kw}^{(1)} + a - \sum_{j=1}^{a} \tau_{i_j,\mathcal{B}} \\ \displaystyle\sum_{i=1}^{n} \mathrm{eval}_h\left(\overline{\xi}_{i,q,\mathcal{B}}\right) \leqslant \kappa_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + \frac{\tilde{q}}{M} + a - \sum_{j=1}^{a} \tau_{i_j,\mathcal{B}} + \frac{a}{2^h} \end{cases} \quad (3.14)$$

Soit $\tau_{\mathcal{B}} = \max\{\sum_{i\in\mathcal{I}_k} \tau_{i,\mathcal{B}} \mid \mathcal{I}_k \subset [\![1,n]\!],\ |\mathcal{I}_k| = k\}$. Si $\tau_{\mathcal{B}} + \Delta_{kw}^{(1)} < 1$ et $\frac{k}{2^h} \leqslant 1$, alors les Inégalités (3.14) impliquent :

$$\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) = \lfloor \sum_{i=1}^{n} \mathrm{eval}_h\left(\overline{\xi}_{i,q,\mathcal{B}}\right) \rfloor \in \{\kappa_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + a - 1, \kappa_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + a, \kappa_{\mathcal{B}}\left(\tilde{\boldsymbol{q}}_{\mathcal{B}}\right) + a + 1\}.$$

De ce fait, l'Égalité (3.13) est bien vérifiée.

Nous modifions alors l'algorithme de réduction de manière à ce que la valeur renvoyée par le Cox soit $\tilde{\kappa}_{\mathcal{B}}\left(\overline{\boldsymbol{q}}_{\mathcal{B}}\right) - 1$. Par suite, en notant $\widehat{q}$ l'entier donné par $\mathrm{Bex}_{kw,h}(\mathcal{B}, \mathcal{B}', \overline{\boldsymbol{q}}_{\mathcal{B}}, -1)$, nous avons alors $\widehat{q} \in \{\tilde{q}, \tilde{q} + M, \tilde{q} + 2M\}$.

Un point clef dans la démonstration du Théorème 3.3 est de s'assurer que $t = x + \widehat{q}p$ appartient à l'intervalle $[\![0, (1 - \alpha_{kw})MM'[\![$. Par suite, nous imposons une nouvelle contrainte à M, à savoir :

$$M > \frac{\sigma p}{1 - \Delta_{kw}^{(1)} - \tau_{\mathcal{B}}} \text{ et } M' \geqslant \frac{3p}{1 - \alpha_{kw}}$$

où nous rappelons que $\alpha_{kw} \in [\Delta_{kw}^{(2)}, 1[$ permet de corriger la seconde conversion de base dans le cas de l'absence de fautes. Ainsi, nous obtenons effectivement :

$$\begin{aligned} 0 \leqslant t = xy + \widehat{q}p &< \left(1 - \Delta_{kw}^{(1)} - \tau_{\mathcal{B}}\right) pM + (2 + \Delta_{kw}^{(1)} + \sum_{j=1}^{u} \tau_{i_j,\mathcal{B}})Mp \\ &< 3Mp \\ &\leqslant (1 - \alpha_{kw})\, MM'. \end{aligned} \tag{3.15}$$

Effet des fautes de catégorie 2

Le terme correcteur α_{kw} introduit précédemment et utilisé lors de la seconde conversion Bex_2 garantit que toute valeur $s < (1 - \alpha_{kw})\, M'$ est convertie complètement.

À cette étape, v coefficients $\xi_{j,s,\mathcal{B}'}$ indexés par $\mathcal{J}_v = \{j_1, \ldots, j_v\}$ sont supposés être affectés par une faute matérielle. Si $\overline{\xi}_{j,s,\mathcal{B}'} < m'_j$ pour tout $j \in \mathcal{J}_v$, le modèle théorique est directement utilisable. Il reste donc à considérer les cas $\overline{\xi}_{j_i} = m'_{j_i} + e_{j_i}$ avec $e_{j_i} \in [\![0, 2^r - m'_{j_i}[\![$. Soit $\overline{s}$ l'unique entier dans $[\![0, M'[\![$ défini par les résidus $\overline{s}_j = s_j$ si $j \notin \mathcal{J}_v$, $\overline{s}_j = \left|e_j M'_j\right|_{m'_j}$ si $j \in \mathcal{J}_v$, où s est par définition $\left|tM^{-1}\right|_{M'}$ et $t = xy + \widehat{q}p$. Ainsi, $\overline{s}$ s'écrit $\overline{s} = s + b_{\mathcal{J}_v} M'_{\mathcal{J}}$ avec $0 < |b_{\mathcal{J}_v}| < \prod_{i=1}^{v} m'_{j_i}$ et

$$\sum_{i=1}^{\ell} \overline{\xi}_{i,s,\mathcal{B}'} M'_i = \sum_{j=1}^{\ell} \xi_{j,\overline{s},\mathcal{B}'} M'_j + vM = \mathrm{sum}_{\mathcal{B}'}\left(\overline{\boldsymbol{s}}_{\mathcal{B}'}\right) + vM'.$$

Il s'agit désormais d'exhiber des conditions garantissant que :

$$\tilde{\kappa}_{\mathcal{B}'}\left(\overline{\boldsymbol{s}}_{\mathcal{B}'}\right) = \kappa_{\mathcal{B}'}\left(\overline{\boldsymbol{s}}_{\mathcal{B}'}\right) + v + \delta, \quad \delta \in \{0, 1\}.$$

Reprenant les Inégalités (3.14) de la discussion concernant les fautes de catégorie 1 et en les adaptant à la situation présente avec les résidus de $\overline{s}$, nous obtenons les inégalités suivantes :

$$\begin{cases} \sum_{i=1}^{\ell} \mathrm{eval}_h\left(\overline{\xi}_{i,\overline{s},\mathcal{B}'}\right) + \alpha_{kw} \geqslant \kappa_{\mathcal{B}'}\left(\overline{\mathbf{s}}_{\mathcal{B}'}\right) + \dfrac{\overline{s}}{M'} - \Delta_{kw}^{(2)} + v - \sum_{j=1}^{v} \tau_{i_j,\mathcal{B}'} + \alpha_{kw}, \\ \sum_{i=1}^{\ell} \mathrm{eval}_h\left(\overline{\xi}_{i,\overline{s},\mathcal{B}'}\right) + \alpha_{kw} \leqslant \kappa_{\mathcal{B}'}\left(\overline{\mathbf{s}}_{\mathcal{B}'}\right) + \dfrac{\overline{s}}{M'} + v - \sum_{j=1}^{v} \tau_{i_j,\mathcal{B}'} + \dfrac{v}{2^h} + \alpha_{kw}. \end{cases} \tag{3.16}$$

Soit $\eta_{\mathcal{B}'} = \max\{\frac{v}{2^h} - \sum_{j\in\mathcal{J}_v} \tau_{j,\mathcal{B}'} \mid v \in [\![1,k]\!], \mathcal{J}_v \subseteq [\![1,\ell]\!], |\mathcal{J}_v| = v\} < \frac{k}{2^h}$. Si α_{kw} vérifie les inégalités suivantes :

$$\Delta_{kw}^{(2)} + \tau_{\mathcal{B}'} \leqslant \alpha_{kw} < 1 - \eta_{\mathcal{B}'} \tag{3.17}$$

alors la valeur calculée par le Cox est :

$$\tilde{\kappa}_{\mathcal{B}'}\left(\overline{\mathbf{s}}_{\mathcal{B}'}\right) = \lfloor \sum_{i=1}^{\ell} \mathrm{eval}_h\left(\overline{\xi}_{i,\overline{s},\mathcal{B}'}\right) + \alpha_{kw} \rfloor = \kappa_{\mathcal{B}'}\left(\overline{\mathbf{s}}_{\mathcal{B}'}\right) + v + \delta,\ \delta \in \{0,1\}.$$

Finalement,

$$\begin{aligned} \mathrm{Bex}_{kw,h}\left(\mathcal{B}', \overline{\mathbf{s}}_{\mathcal{B}'}, \alpha_{kw}\right) &= \sum_{i=1}^{\ell} \overline{\xi}_{i,\overline{s},\mathcal{B}'} M_i' - \tilde{\kappa}_{\mathcal{B}'}\left(\overline{\mathbf{s}}_{\mathcal{B}'}\right) M' \\ &= \sum_{i=1}^{\ell} \xi_{i,\overline{s},\mathcal{B}'} + vM' - \left(\kappa_{\mathcal{B}'}\left(\overline{\mathbf{s}}_{\mathcal{B}'}\right) + v + \delta\right) M' \\ &= \overline{s} - \delta M' \\ &= s + b_{\mathcal{J}_v} M'_{\mathcal{J}} - \delta M',\ \delta \in \{0,1\}. \end{aligned} \tag{3.18}$$

Vu la seconde inégalité de (3.16), nous avons de plus :

$$\delta = 1 \Rightarrow \frac{\overline{s}}{M'} - \sum_{j=1}^{v} \tau_{i_j,\mathcal{B}'} + \frac{v}{2^h} + \alpha_{kw} \geqslant 1 \Rightarrow \overline{s} \geqslant \left(1 + \sum_{j=1}^{v} \tau_{i_j,\mathcal{B}'} - \frac{v}{2^h} - \alpha_{kw}\right) M'. \tag{3.19}$$

Les Inégalités (3.19) font sens sous l'hypothèse que α_{kw} vérifie les bornes (3.17), puisque nous avons alors les inégalités suivantes :

$$\begin{cases} 1 + \sum_{j=1}^{v} \tau_{i_j,\mathcal{B}'} - \dfrac{v}{2^h} - \alpha_{kw} > 1 + \sum_{j=1}^{v} \tau_{i_j,\mathcal{B}'} - \dfrac{v}{2^h} - 1 + \eta_{\mathcal{B}'} \geqslant 0, \\ 1 + \sum_{j=1}^{v} \tau_{i_j,\mathcal{B}'} - \dfrac{v}{2^h} - \alpha_{kw} \leqslant 1 + \tau_{\mathcal{B}'} - \alpha_{kw} \leqslant 1 - \Delta_{kw}^{(2)} < 1. \end{cases}$$

Si $\overline{s}$ vérifie l'inégalité de droite de (3.19), alors il est clair que pour tout réel $\alpha'_{kw} \in [\alpha_{kw} + \eta_{\mathcal{B}'}, 1[$, nous avons également dans ce cas $\overline{s} \geqslant \left(1 - \alpha'_{kw}\right) M'$. En effet, pour un tel $\alpha'_{kw} \in [\alpha_{kw} + \eta_{\mathcal{B}'}, 1[$ et par définition de $\eta_{\mathcal{B}'}$, nous avons :

$$\begin{aligned} 1 + \sum_{j=1}^{v} \tau_{i_j,\mathcal{B}'} - \frac{v}{2^h} - \alpha_{kw} &\geqslant 1 - \eta_{\mathcal{B}'} - \alpha_{kw} \\ &\geqslant 1 - \alpha'_{kw}. \end{aligned} \tag{3.20}$$

Ainsi, si la base $\mathcal{B}'$ vérifie $3p \leqslant (1-\alpha'_{kw})M'$ pour un certain $\alpha'_{kw} \in [\alpha_{kw}+\eta_{\mathcal{B}'}, 1[$, alors des implications 3.19 et des Inégalités (3.20) nous déduisons que :

$$\delta = 1 \Rightarrow \bar{s} \geqslant \left(1-\alpha'_{kw}\right) M'. \tag{3.21}$$

Finalement, nous obtenons :

$$-\alpha'_{kw}M' \leqslant \widehat{s} = \mathrm{Bex}_{kw,h}(\mathcal{B}', \bar{s}_{\mathcal{B}'}, \alpha_{kw}) < (1-\alpha'_{kw})M'. \tag{3.22}$$

La nouvelle condition sur la base RNS $\mathcal{B}'$ garantit alors que la quantité $t = xy + \widehat{q}p$ vérifie toujours (cf. (3.15)) :

$$0 \leqslant t < 3pM \leqslant \left(1-\alpha'_{kw}\right) MM'. \tag{3.23}$$

La fin de la preuve de la détection reste alors identique au cas théorique. Le premier point clef était de montrer que $|t - M \times \widehat{s}| < MM'$, ce qui découle dans notre cas de (3.22) et (3.23). Le second point clef était de voir que les seuls résidus non nuls de cet entier dans la base $\mathcal{B} \cup \mathcal{B}'$ ont leur indice dans $\mathcal{I}_u \cup \mathcal{J}_v$. Ceci reste évidemment vrai dans le contexte présent. En effet, les seuls résidus non nuls de $|t - M \times \widehat{s}|$ dans $\mathcal{B}$, indexés par $\mathcal{I} \subseteq \mathcal{I}_u$, sont ceux pour lesquels $\tilde{q}_i \neq q_i$. De même dans $\mathcal{B}'$, les résidus non nuls, indexés par $\mathcal{J} \subseteq \mathcal{J}_v$, sont ceux pour lesquels $\bar{s}_j \neq s_j$. Ainsi, $|t - M \times \widehat{s}|$ est un multiple non nul de $M_{\mathcal{I}}M'_{\mathcal{J}}$ et

$$0 < \frac{|t - M \times \widehat{s}|}{M_{\mathcal{I}}M'_{\mathcal{J}}} < \prod_{i\in\mathcal{I}} m_i \prod_{j\in\mathcal{J}} m'_j.$$

Finalement, l'hypothèse que les moduli redondants vérifient $m_R > m$ pour tout $m_R \in \mathcal{B}$ et tout $m \in \mathcal{B} \cup \mathcal{B}'$ suffit alors à assurer la détection.

Nouvelles conditions sur les bases et la qualité de l'approximation

Dans les discussions précédentes, des conditions suffisantes sur la qualité de l'approximation et sur les tailles des bases $\mathcal{B}$ et $\mathcal{B}'$ ont été exhibées de manière à prouver le théorème suivant :

Théorème 3.4 *Soit des bases copremières $\mathcal{B}$, $\mathcal{B}'$ et $\mathcal{B}_R$ avec* $\mathrm{Card}(\mathcal{B}) = n$, $\mathrm{Card}(\mathcal{B}') = \ell$ *et* $\mathrm{Card}(\mathcal{B}_R) = k$, *et les paramètres suivants :*

- $\tau_{\mathcal{B}} = \max\{\sum_{i\in\mathcal{I}_k} (1 - \mathrm{eval}_h(m_i)) \mid \mathcal{I}_k \subset [\![1,n]\!],\ |\mathcal{I}_k| = k\}$,
- $\tau_{\mathcal{B}'} = \max\{\sum_{j\in\mathcal{J}_k} (1 - \mathrm{eval}_h(m'_j)) \mid \mathcal{J}_k \subset [\![1,\ell]\!],\ |\mathcal{J}_k| = k\}$,
- $\eta_{\mathcal{B}'} = \max\{\frac{v}{2^h} - \sum_{j\in\mathcal{J}_v} \tau_{j,\mathcal{B}'} \mid v \in [\![1,k]\!], \mathcal{J}_v \subseteq [\![1,\ell]\!], |\mathcal{J}_v| = v\}$,
- $\Delta^{(1)}_{kw} \in [0,1[$ *un majorant de l'erreur* $\sum_{i=1}^{n} \left(\frac{\xi_{i,q,\mathcal{B}}}{m_i} - \mathrm{eval}_h\left(\xi_{i,q,\mathcal{B}}\right)\right)$,
- $\Delta^{(2)}_{kw} \in [0,1[$ *un majorant de l'erreur* $\sum_{j=1}^{\ell} \left(\frac{\xi_{j,s,\mathcal{B}'}}{m'_j} - \mathrm{eval}_h\left(\xi_{j,s,\mathcal{B}'}\right)\right)$.

Soit alors les hypothèses suivantes :

$$\mathcal{H} : \begin{cases} \sigma \geqslant 9,\ \dfrac{k}{2^h} \leqslant 1, \\ \Delta_{kw}^{(1)} + \tau_{\mathcal{B}}, < 1\ \Delta_{kw}^{(2)} + \tau_{\mathcal{B}'} + \eta_{\mathcal{B}'} < 1, \\ \Delta_{kw}^{(2)} + \tau_{\mathcal{B}'} \leqslant \alpha_{kw} < 1 - \eta_{\mathcal{B}'},\ \alpha_{kw} + \eta_{\mathcal{B}'} \leqslant \alpha'_{kw} < 1, \\ \sigma p < (1 - \Delta_{kw}^{(1)} - \tau_{\mathcal{B}})M,\ 3p \leqslant (1 - \alpha'_{kw})M' \end{cases} \tag{3.24}$$

Si chaque modulus redondant $m_R \in \mathcal{B}_R$ vérifie

$$m_R \wedge m = 1 \text{ et } m_R > m \text{ et tout } m \in \mathcal{B} \cup \mathcal{B}'$$

alors l'Algorithme 15 détecte voire corrige automatiquement toutes les t-fautes matérielles multiples, pour tout $t \in [\![1, k]\!]$. De plus, si aucune faute n'est injectée ou si une correction automatique a lieu, la sortie s de la multiplication modulaire est dans $[\![0, 3p[\![$.

Proposition 3.3 *Soit un entier c tel que tout modulus $m \in \mathcal{B} \cup \mathcal{B}'$ vérifie $2^r - m < 2^c$. Soit h un entier tel que $h \leqslant r - c$, $\frac{n+k}{2^{h-1}} < 1$ et $\frac{2\ell+3k}{2^h} < 1$. Si de plus nous avons $\alpha_{kw} = \frac{\ell+k}{2^{h-1}}$, $\alpha'_{kw} = \frac{2\ell+3k}{2^h}$, et $\sigma p < (1 - \frac{n+k}{2^{h-1}})M$, $3p \leqslant (1 - \alpha'_{kw})M'$, alors les hypothèses (3.24) du Théorème 3.4 sont satisfaites.*

Démonstration. Nous pouvons remarquer que α_{kw} est bien représentable dans le registre de h bits du Cox.

La preuve de la proposition repose simplement sur l'hypothèse $h \leqslant r - c$ et sur les inégalités suivantes, constatées précédemment :

$$\begin{cases} \tau_{\mathcal{B}}, \tau_{\mathcal{B}'} \leqslant \dfrac{k}{2^{r-c}} + \dfrac{k}{2^h} \leqslant \dfrac{k}{2^{h-1}} \\ \eta_{\mathcal{B}'} < \dfrac{k}{2^h} \\ \Delta_{kw}^{(1)} \leqslant \dfrac{n}{2^{r-c}} + \dfrac{n}{2^h} \leqslant \dfrac{n}{2^{h-1}} \\ \Delta_{kw}^{(2)} \leqslant \dfrac{\ell}{2^{r-c}} + \dfrac{\ell}{2^h} \leqslant \dfrac{\ell}{2^{h-1}} \end{cases}$$

Ainsi, il vient $\Delta_{kw}^{(1)} + \tau_{\mathcal{B}} \leqslant \frac{n+k}{2^{h-1}} < 1$ et $\Delta_{kw}^{(2)} + \tau_{\mathcal{B}'} + \eta_{\mathcal{B}'} \leqslant \frac{2\ell+3k}{2^h} < 1$. De plus, nous vérifions aisément que $\Delta_{kw}^{(2)} + \tau_{\mathcal{B}'} \leqslant \alpha_{kw} = \frac{\ell+k}{2^{h-1}} \leqslant 1 - \eta_{\mathcal{B}'}$, et $\alpha_{kw} + \eta_{\mathcal{B}'} \leqslant \frac{2\ell+3k}{2^h} = \alpha'_{kw} < 1$.

Enfin, nous avons finalement $\sigma p < (1 - \frac{n+k}{2^{h-1}})M \leqslant (1 - \Delta_{kw}^{(1)} - \tau_{\mathcal{B}})M$. La condition sur M' est identique à celle du théorème.

Pour conclure, les conditions du Théorème 3.4 sont vérifiées. □

Exemple 3.1 *Afin de montrer que les hypothèses de la Proposition 3.3 concernant le paramètre h sont réalistes, nous considérons une nouvelle fois les paramètres de l'Exemple 1.4 pour le calcul d'un RSA de 1024 bits avec $n = 33$ et $r = 32$. Nous fixons par exemple la capacité de détection à $k = 6$. L'ensemble $[\![2^{32} - 2^{10}, 2^{32}[\![$ contient 72 nombres premiers, ce qui assure de disposer d'assez de moduli pour les trois bases $\mathcal{B}$, $\mathcal{B}'$ et $\mathcal{B}_R$. Ainsi, nous fixons le paramètre c à 10.*

Par conséquent, h doit vérifier $h \leqslant r - c = 23$, ainsi que $\frac{2n+3k}{2^h} < 1$. Cette dernière inégalité est vérifiée dès que h entier est supérieur ou égal à 7. Par suite, en posant

Algorithme 15 : MulModRRNS_HW $(\mathcal{B}, \mathcal{B}', \mathcal{B}_R, x, y, p, h, \alpha_{kw})$

Données :

- $\mathcal{B}$, $\mathcal{B}'$, $\mathcal{B}_R$, h et α_{kw} vérifient les hypothèse du Théorème 3.4 ;
- un modulus p représenté par ses résidus précalculés $\boldsymbol{p}_{\mathcal{B}' \cup \mathcal{B}_R}$ dans la base $\mathcal{B}' \cup \mathcal{B}_R$, et premier avec M ;
- deux entiers x, y représentés par leurs résidus $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}'}, \boldsymbol{x}_{\mathcal{B}_R})$ et $(\boldsymbol{y}_{\mathcal{B}}, \boldsymbol{y}_{\mathcal{B}'}, \boldsymbol{y}_{\mathcal{B}_R})$ tels que $xy < \sigma p$;
- les résidus précalculés de $\left|-p^{-1}\right|_M$ dans $\mathcal{B}$ et de $\left|M^{-1}\right|_{M'M_R}$ dans $\mathcal{B}' \cup \mathcal{B}_R$

Résultat : les résidus de $s \equiv xyM^{-1} \bmod p$ dans $\mathcal{B} \cup \mathcal{B}' \cup \mathcal{B}_R$, $s \in [\![0, 3p[\![$

début
- **pour** $i \leftarrow 1$ **à** n **faire**
  - $q_i \leftarrow x_i \times y_i \times \left|-p^{-1}\right|_{m_i} \bmod m_i$
- $\left(\widehat{q}'_1, \ldots, \widehat{q}'_\ell, \widehat{q}_{R,1}, \ldots, \widehat{q}_{R,k}\right) \leftarrow \mathrm{Bex}_{kw,h}\left(\mathcal{B}, \mathcal{B}' \cup \mathcal{B}_R, (q_1, \ldots, q_n), -1\right)$
- **pour** $j \leftarrow 1$ **à** ℓ **faire**
  - $t'_j \leftarrow x'_j \times y'_j + \widehat{q}'_j \times p'_j \bmod m'_j$
- **pour** $z \leftarrow 1$ **à** k **faire**
  - $t_{R,z} \leftarrow x_{R,z} \times y_{R,z} + \widehat{q}_{R,z} \times p_{R,z} \bmod m_{R,z}$
- **pour** $j \leftarrow 1$ **à** ℓ **faire**
  - $s'_j \leftarrow t'_j \times \left|M^{-1}\right|_{m'_j} \bmod m'_j$
- **pour** $z \leftarrow 1$ **à** k **faire**
  - $s_{R,z} \leftarrow t_{R,z} \times \left|M^{-1}\right|_{m_{R,z}} \bmod m_{R,z}$
- $(s_1, \ldots, s_n, \widehat{s}_{R,1}, \ldots, \widehat{s}_{R,k}) \leftarrow \mathrm{Bex}_{kw,h}\left(\mathcal{B}', \mathcal{B} \cup \mathcal{B}, \left(s'_1, \ldots, s'_\ell\right), \alpha_{kw}\right)$
- **si** $\boldsymbol{s}_{\mathcal{B}_R} \neq \widehat{\boldsymbol{s}}_{\mathcal{B}_R}$ **alors**
  - **retourner** `Faute détectée.`
- **sinon**
  - **retourner** $\left(s_1, \ldots, s_n, s'_1, \ldots, s'_\ell, s_{R,1}, \ldots, s_{R,k}\right)$

$h = 7$, le coefficient de correction pour la seconde conversion est $\alpha_{kw} = \frac{84}{128}$. De plus, les conditions sur les entiers M et M' sont parfaitement vérifiées, et ceci avec une grande liberté de choix du paramètre σ (il suffit de voir que, comme $M > (2^r - 2^c)^n$, au moins tout σ tel que $\log_2(\sigma) \leqslant n \log_2(2^r - 2^c) - 1023 \sim 33$ convient). Ainsi, pour $\sigma = 9$ et $\alpha'_{kw} = \frac{2n+3k}{2^h} = \frac{66+18}{2^7}$, nous vérifions que :

$$\begin{cases} \log_2(M), \log_2(M') > \log_2\left((2^{32} - 2^{10})^n\right) \sim 1056 \\ \log_2\left(\dfrac{9p}{1 - \frac{33+6}{2^6}}\right) \sim 1028.5,\ \log_2\left(\dfrac{3p}{1 - \frac{66+18}{2^7}}\right) \sim 1027.1 \end{cases}$$

3.2 Arithmétique protégée pour les calculs dans $\mathbb{F}_{p^s}$

La construction des représentations RNS des éléments d'un corps fini $\mathbb{F}_{p^s}$, explicitée dans la Partie 1.4, peut faire l'objet d'une adjonction de redondance. Le principe est similaire à celui des RNS redondants pour les entiers introduits en 2.2.

Après l'introduction du modèle de faute, les conditions nécessaires et suffisantes sur la forme de la redondance garantissant la détection d'un nombre maximal de fautes fixé seront fournies. Puis une version redondante de l'Algorithme 12 (p. 43) sera proposée, permettant la détection d'erreurs dites multiples dans un sens identique à celui donné par la Définition 3.1 (p. 89).

Le contexte est fixé par le choix d'un polynôme unitaire $N(X)$ de degré s et irréductible sur $\mathbb{F}_p$. Le corps $\mathbb{F}_{p^s}$ est identifié à $\mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$. $\mathcal{B}$ désigne un ensemble de polynômes $\{m_1(X), \ldots, m_n(X)\}$ premiers entre eux deux à deux. C'est donc une base de l'espace vectoriel $\mathbb{F}_p[X]_d$ des polynômes sur $\mathbb{F}_p$ de degré strictement inférieur à $d = \sum_{i=1}^{n} d_i$, où $d_i = \deg(m_i)$. L'isomorphisme $\varphi_{\mathcal{B}}$, définissant la représentation RNS associée à la base $\mathcal{B}$, est donné par le théorème des restes chinois :

$$\begin{array}{rccl} \varphi_{\mathcal{B}} : & \mathbb{F}_p[X]_d & \to & \mathbb{F}_p[X]/m_1(X)\mathbb{F}_p[X] \times \ldots \times \mathbb{F}_p[X]/m_n(X)\mathbb{F}_p[X] \\ & A(X) & \mapsto & \left(|A(X)|_{m_1(X)}, \ldots, |A(X)|_{m_n(X)}\right) \\ & & & = (A_1(X), \ldots, A_n(X)) = \boldsymbol{A}_{\mathcal{B}}. \end{array}$$

L'application inverse est utilisée pour définir les procédures de conversion de base. Celles-ci se basent sur les interpolations newtonienne (Algorithme 9) et lagrangienne (Algorithme 10).

3.2.1 Modèle de faute

Le modèle de faute multiple est directement introduit, sans passer par le cas particulier des fautes uniques. L'effet de telles fautes est ensuite explicité.

Définition 3.3 *Soit k un entier dans $[\![1, n]\!]$. Une k-faute d'indices $\mathcal{I}_k \subseteq [\![1, n]\!]$ sur des résidus $\boldsymbol{A}_{\mathcal{B}} = (A_1(X), \ldots, A_n(X))$ est la donnée d'un ensemble de k indices $\mathcal{I}_k \subseteq [\![1, n]\!]$ et de k polynômes non nuls $(E_i(X))_{\mathcal{I}k}$ vérifiant $\deg(E_i) < d_i = \deg(m_i)$. Les résidus erronés sont définis par $\overline{A}_i(X) = (A(X) + E_i(X)) \bmod m_i(X) = A_i(X) + E_i(X)$.*

Proposition 3.4 *Soit $\boldsymbol{A}_{\mathcal{B}}(X)$ des résidus d'un polynôme $A(X) \in \mathbb{F}_p[X]_d$, et une k-faute d'indices $\mathcal{I}_k$ et de valeurs $(E_i(X))_{\mathcal{I}k}$. Si $\overline{\boldsymbol{A}}_{\mathcal{B}}$ dénote les résidus $\boldsymbol{A}_{\mathcal{B}}$ affectés par la faute considérée,*

alors il existe un polynôme non nul $E_{\mathcal{I}_k}(X)$ de degré au plus $\sum_{i\in\mathcal{I}_k} d_i - 1$ tel que :

$$\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) = A(X) + E_{\mathcal{I}_k}(X) \times \prod_{j\in[\![1,n]\!]\setminus\mathcal{I}_k} m_j(X). \tag{3.25}$$

Plus précisément, notant $\mathcal{B}_{\mathcal{I}_k} = \{m_i(X)\}_{i\in\mathcal{I}_k}$ et $\boldsymbol{E}_{\mathcal{B}_{\mathcal{I}_k}}$ les résidus $(E_i(X))_{\mathcal{I}k}$ de la k-faute considérée, alors nous avons :

$$\begin{aligned} E_{\mathcal{I}_k}(X) &= \varphi_{\mathcal{B}_{\mathcal{I}_k}}^{-1}\left(\boldsymbol{E}_{\mathcal{B}_{\mathcal{I}_k}}\right) \\ &= \sum_{i\in\mathcal{I}_k} \left| E_i(X) \prod_{j\in\mathcal{I}_k\setminus\{i\}} m_j(X)^{-1} \right|_{m_i(X)} \times \prod_{j\in\mathcal{I}_k\setminus\{i\}} m_j(X). \end{aligned} \tag{3.26}$$

Démonstration. Il est clair que pour tout $j \notin \mathcal{I}_k$, le résidu modulo $m_j(X)$ du membre de droite de l'Équation (3.25) est égal à $A_j(X)$ et donc à $\overline{A}_j(X)$. De plus, pour tout $i \in \mathcal{I}_k$, la définition de $E_{\mathcal{I}_k}$ donnée par l'Équation (3.26) implique $E_{\mathcal{I}_k}(X) \bmod m_i(X) = E_i(X)$, ce qui implique que pour tout $i \in \mathcal{I}_k$, le résidu modulo $m_i(X)$ du membre de droite de l'Équation (3.25) vaut $A_i(X) + E_i(X) = \overline{A}_i(X)$. Ensuite, $\deg(E_{\mathcal{I}_k}) \leqslant \sum_{i\in\mathcal{I}_k} d_i - 1$. Ainsi,

$$\begin{aligned} \deg\left(A(X) + E_{\mathcal{I}_k}(X) \times \prod_{j\in[\![1,n]\!]\setminus\mathcal{I}_k} m_j(X)\right) &\leqslant \max\left(\deg(A), \sum_{i\in\mathcal{I}_k} d_i - 1 + \sum_{j\in[\![1,n]\!]\setminus\mathcal{I}_k} d_j\right) \\ &= d - 1. \end{aligned}$$

La bijectivité de l'application $\varphi_{\mathcal{B}}$ permet alors de conclure à l'égalité et justifie notamment le fait que $E_{\mathcal{I}_k}$ est non nul. □

La proposition précédente précise la manière dont une faute multiple modifie un polynôme dont les résidus sont perturbés par ladite faute. La proposition suivante quant à elle exhibe l'ensemble de toutes les valeurs atteignables par l'injection d'une faute de taille maximale k.

Proposition 3.5 *Pour tout entier $k \in [\![1, n]\!]$, tout ensemble de k indices $\mathcal{I}_k$, tout polynôme $E(X) \in \mathbb{F}_p[X]$ de degré au plus égal à $\sum_{i\in\mathcal{I}_k} d_i - 1$ et tout polynôme $A \in \mathbb{F}_p[X]_d$ défini par ses résidus $\boldsymbol{A}_{\mathcal{B}}$ dans la base $\mathcal{B}$, il existe un entier $t \in [\![1, k]\!]$ et une t-faute d'indices $\mathcal{I}_t \subseteq \mathcal{I}_k$ affectant $\boldsymbol{A}_{\mathcal{B}}$ telle que :*

$$\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}(X)\right) - A(X) = E(X) \times \prod_{i\notin\mathcal{I}_k} m_i(X). \tag{3.27}$$

Démonstration. L'ensemble de polynômes $\mathcal{B}_{\mathcal{I}_k} = \{m_i(X)\}_{i\in\mathcal{I}_k}$ est une base de l'espace vectoriel $\mathbb{F}_p[X]_{d_{\mathcal{I}_k}}$ où $d_{\mathcal{I}_k} = \sum_{i\in\mathcal{I}_k} d_i$. Soit $E \in \mathbb{F}_p[X]$ un polynôme de degré au plus égal à $\sum_{i\in\mathcal{I}_k} d_i - 1$. Alors le polynôme $\widehat{E}(X) = E(X) \times \prod_{i\notin\mathcal{I}_k} m_i(X)$ est de degré strictement inférieur à $d = \sum_{i=1}^{n} d_i$. Les indices des résidus non nuls de $\varphi_{\mathcal{B}}\left(\widehat{E}(X)\right)$ constituent un sous-ensemble $\mathcal{I}_t$ de $\mathcal{I}_k$ pour un entier $t \in [\![1, k]\!]$.

Soit donc la t-faute donnée par ces résidus et indices, et $\overline{\boldsymbol{A}}_{\mathcal{B}}$ les résidus d'un polynôme $A \in \mathbb{F}_p[X]_d$ modifiés par cette faute. Comme $\overline{\boldsymbol{A}}_{\mathcal{B}} = \boldsymbol{A}_{\mathcal{B}} + \widehat{\boldsymbol{E}}_{\mathcal{B}}$, la propriété d'isomorphisme de $\varphi_{\mathcal{B}}^{-1}$ implique que :

$$\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) = A(X) + \widehat{E}(X).$$

Ceci achève la preuve de la proposition.

□

3.2.2 Détection des fautes multiples

La détection des fautes introduites par la Définition 3.3 peut être réalisée par l'adjonction d'un polynôme redondant $M_R(X)$ à la base $\mathcal{B}$. Le principe du test de cohérence de la Définition 2.2 s'applique de la même manière au contexte présent. Si $\left(\overline{\boldsymbol{A}}_{\mathcal{B}}, A_R\right)$ est un ensemble de résidus issus des résidus $(\boldsymbol{A}_{\mathcal{B}}, A_R)$ d'un polynôme $A \in \mathbb{F}_p[X]_d$ et potentiellement affecté par une faute multiple, alors la procédure de détection teste l'égalité suivante dans $\mathbb{F}_p[X]/M_R(X)\mathbb{F}_p[X]$:

$$A_R(X) \stackrel{?}{\equiv} \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) \bmod M_R(X). \tag{3.28}$$

L'Égalité (3.28) est bien vérifiée pour tout élément de $\mathbb{F}_p[X]_d$, puisque dans ce cas $A(X) = \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right)$ et par définition $A_R(X) = A(X) \bmod M_R(X)$.

Pour une taille maximale de faute k, la Proposition 3.4 implique une condition nécessaire sur $M_R(X)$ pour que l'égalité (3.28) soit non vérifiée dès qu'une t-faute est présente, pour tout entier $t \in [\![1, k]\!]$. En particulier, le degré de $M_R(X)$ ne peut être strictement inférieur à toute somme du type $\sum_{i\in\mathcal{I}_k} d_i$. Le théorème suivant va plus loin en exhibant une condition nécessaire et suffisante sur le degré de $M_R(X)$ pour assurer la détection de toute faute de taille maximale k.

Par la suite, nous supposons que $M_R(X)$ est premier avec $M(X)$. Le cas général d'une base de polynômes redondants ni supposés premiers à la base $\mathcal{B}$ et ni supposés premiers entre eux est détaillé en annexe.

Théorème 3.5 *Soit $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ un ensemble de n polynômes de $\mathbb{F}_p[X]$ premiers entre eux deux à deux, $d_i = \deg(m_i)$ pour tout $i \in [\![1, n]\!]$ et $d = \sum_{i=1}^{n} d_i$. Soit $M_R(X) \in \mathbb{F}_p[X]$ un polynôme de degré d_R premier avec $M(X)$, et k un entier dans $[\![1, n]\!]$. Alors pour tout $A \in \mathbb{F}_p[X]_d$, tout entier $t \in [\![1, k]\!]$ et toute t-faute affectant $\varphi_{\mathcal{B}}(A) = \boldsymbol{A}_{\mathcal{B}}$,*

$$A_R \neq \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) \bmod M_R(X)$$

si, et seulement si,

$$d_R \geqslant \max\{\sum_{i\in\mathcal{I}_k} d_i \mid \mathcal{I}_k \subseteq [\![1, n]\!],\ |\mathcal{I}_k| = k\}.$$

Démonstration. • Nécessité : la preuve est faite par contraposition. Soit un ensemble de k indices $\mathcal{I}_k$ tel que $d_R < \sum_{i\in\mathcal{I}_k} d_i$. Soit $A(X) \in \mathbb{F}_p[X]_d$. Par la Proposition 3.4, il existe un entier $t \in [\![1, k]\!]$ et une t-faute d'indices $\mathcal{I}_t \subseteq \mathcal{I}_k$ telle que, si elle affecte les résidus de A,

$$\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) - A(X) = M_R(X) \times \prod_{i\notin\mathcal{I}_k} m_i(X). \tag{3.29}$$

Par conséquent, l'Équation (3.29) est nulle modulo $M_R(X)$, ce qui achève la preuve de la nécessité.

- Suffisance : soit une t-faute non nulle d'indices $\mathcal{I}_t$, avec $t \in [\![1, k]\!]$, affectant $A(X) \in \mathbb{F}_p[X]_d$. Par la Proposition 3.5,

$$\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) - A(X) = E(X) \prod_{i \notin \mathcal{I}_t} m_i(X)$$

avec $\deg(E) < \sum_{i \in \mathcal{I}_t} d_i$. Ainsi, nous pouvons écrire :

$$\left|\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) - A(X)\right|_{M_R(X)} = 0 \Leftrightarrow \left|E(X) \prod_{i \notin \mathcal{I}_t} m_i(X)\right|_{M_R(X)} = 0.$$

Or, la deuxième équation de l'équivalence précédente implique que $M_R(X)$ divise $E(X)$ et donc :

$$d_R < \sum_{i \in \mathcal{I}_t} d_i.$$

La preuve est achevée.

□

Lorsque le modèle de faute est généralisé de manière à prendre en compte les fautes affectant les résidus redondants, il est nécessaire de remplacer le polynôme $M_R(X)$ par un ensemble de polynômes $\{m_{R,1}(X), \ldots, m_{R,k}(X)\}$. Dans ce contexte, le test de cohérence est défini sur l'ensemble des polynômes de $\mathbb{F}_p[X]_d$.

Vu la Remarque 1.16 (p. 40), les conversions Bex_{Lag} et Bex_{New} utilisées pour évaluer une fonction du type $\varphi_{\mathcal{B}_R} \circ \varphi_{\mathcal{B}}^{-1}$ où $\mathcal{B}$ est une base RNS sont toujours complètement réduites. Par conséquent, si $\overline{A}(X)$ est défini par $\overline{A}(X) = \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right)$ où $\overline{\boldsymbol{A}}_{\mathcal{B}}$ sont les résidus de $A \in \mathbb{F}_p[X]_d$ affectés par une faute multiple, alors $\overline{A}(X) = \text{Bex}\left(\mathcal{B}, \overline{\boldsymbol{A}}_{\mathcal{B}}\right)$. De ce fait, la procédure de détection est la suivante :

Procédure DetectMultErrExt(Bex,$\mathcal{B}$,$\mathcal{B}_R$,$\boldsymbol{A}_{\mathcal{B}}$,$\boldsymbol{A}_R$)

1 **début**
2 $\quad \tilde{\boldsymbol{A}}_R \leftarrow \text{Bex}\left(\mathcal{B}, \mathcal{B}_R, \boldsymbol{A}_{\mathcal{B}}\right) = \varphi_{\mathcal{B}_R}\left(\varphi_{\mathcal{B}}^{-1}\left(\boldsymbol{A}_{\mathcal{B}}\right)\right)$;
3 $\quad$ **retourner** $\tilde{\boldsymbol{A}}_R == \boldsymbol{A}_R$

Théorème 3.6 *Soit $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ n polynômes premiers entre eux deux à deux et $\mathcal{B}_R = \{m_{R,1}(X), \ldots, m_{R,k}(X)\}$ k polynômes de $\mathbb{F}_p[X]$ premiers entre eux deux à deux et premiers avec les éléments de $\mathcal{B}$, avec $k \leqslant n$.*

Pour tout $\ell \in [\![k+1, n+k]\!]$, il existe une ℓ-faute non détectable.

De plus, la Procédure `DetectMultErrExt` *détecte toutes les ℓ-fautes sur tout ensemble de résidus $(\boldsymbol{A}_{\mathcal{B}}, \boldsymbol{A}_R)$, pour $A \in \mathbb{F}_p[X]_d$, et ceci pour tout $\ell \leqslant k$ si, et seulement si,*

$$\min\{d_{R,z} \mid z \in [\![1, k]\!]\} \geqslant \max\{d_i \mid i \in [\![1, n]\!]\}. \tag{3.30}$$

Démonstration. Si $\ell \geqslant k+1$, il est possible de construire une ℓ-faute non détectée par la Procédure `DetectMultErrExt`. Soit $A(X) = 0$ et une ℓ-faute affectant $\ell - k$ résidus de A d'indices $\mathcal{I}$ ainsi que les k résidus redondants $(A_{R,z}(X))_{z\in[\![1,k]\!]}$. Soit $E_i(X) \in \mathbb{F}_p[X]_{d_i}$ l'erreur sur $A_i(X)$ et E défini par $E(X) = \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}\right) - A(X)$. Par la Proposition 3.4,

$$E(X) = E_{\mathcal{I}} \times \prod_{i\notin\mathcal{I}}^{n} m_i(X)^{-1}.$$

Les autres résidus de l'erreur sont alors définis par $E_{R,z}(X) = |E(X)|_{m_{R,z}(X)}$ pour tout $z \in [\![1,k]\!]$. Une telle erreur n'est ainsi pas détectée.

- Prouvons la nécessité par contraposition.

  Supposons l'existence d'un doublet $(z,i) \in [\![1,k]\!] \times [\![1,n]\!]$ tel que $d_{R,z}(X) < d_i$.

  Soit la k-faute sur $A(X) = 0$ donnée par les valeurs suivantes :

  $$\begin{cases} E_i(X) = |m_{R,z}(X)|_{m_i(X)} = m_{R,z}(X), \\ \forall j \in [\![1,k]\!]\backslash\{z\},\ E_{R,j}(X) = |m_{R,z}M_i(X)|_{m_{R,j}(X)}\,. \end{cases}$$

  En particulier, cela implique :

  $$\varphi_{\mathcal{B}}^{-1}(\boldsymbol{A}_{\mathcal{B}}) = m_{R,z}(X)M_i(X).$$

  Cette faute vérifie $\text{Bex}\left(\mathcal{B},\mathcal{B}_R,\overline{\boldsymbol{A}}_{\mathcal{B}}\right) = \overline{\boldsymbol{A}}_R$ et n'est donc pas détectée.

- Prouvons la suffisance par contraposition.

  Supposons qu'il existe un entier $\ell \leqslant k$ pour lequel il existe une ℓ-faute sur un polynôme $A \in \mathbb{F}_p[X]_d$ non détectée. Cette faute est présumée affecter ℓ_P résidus principaux d'indices $\mathcal{I}_{\ell_P}$ ainsi que ℓ_R résidus redondants. En particulier, $\ell_P + \ell_R = \ell \leqslant k$ et il y a donc au moins $\ell_P \leqslant k - \ell_R$ résidus redondants intègres. Soit ℓ_P tels résidus redondants intègres et d'indices $\mathcal{Z}_{\ell_P}$. Par la Proposition 3.4 qui précise la forme du polynôme donné par des résidus erronés, nous pouvons écrire :

  $$\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) - A(X) = E_{\mathcal{I}_{\ell_P}}(X) \times \prod_{i\notin\mathcal{I}_{\ell_P}} m_i(X)$$

  où $\deg\left(E_{\mathcal{I}_{\ell_P}}\right) < \sum_{i\in\mathcal{I}_{\ell_P}} d_i$. Par hypothèse, la faute n'est pas détectée, ce qui implique en particulier que $\prod_{z\in\mathcal{Z}_{\ell_P}} m_{R,z}$ divise $E_{\mathcal{I}_{\ell_P}}(X)$. Il vient alors :

  $$\sum_{z\in\mathcal{Z}_{\ell_P}} d_{R,z} < \sum_{i\in\mathcal{I}_{\ell_P}} d_i.$$

  Or, ceci contredit l'Hypothèse (3.30).

  La preuve est achevée.

□

Remarque 3.2 *L'utilisation de l'Algorithme 10 de conversion de base basée sur une interpolation de lagrange pour la procédure de détection* `DetectMultErrExt` *dans un corps fini $\mathbb{F}_{p^s}$ nécessite dans ce cas que le corps de base $\mathbb{F}_p$ possède au moins $n+k$ éléments distincts.*

3.2.3 Multiplication modulaire redondante dans $\mathbb{F}_{p^s}$

Les techniques de détection précédentes sont intégrées à l'Algorithme 12 en se basant sur les mêmes principes ayant conduits à l'élaboration de l'Algorithme 13 de multiplication modulaire en RNS redondant pour les entiers.

Le principe de catégorisation des fautes proposé dans la section 2.3.2 est utilisé une nouvelle fois pour le modèle de fautes dans $\mathbb{F}_{p^s}$. Contrairement au cas entier, les fautes de catégorie 4 sont ici inexistantes. D'une part, la conversion Bex_{New} est similaire à Bex_{mrs} et est donc toujours complète. D'autre part, vu la Remarque 1.16 (p. 40), la conversion Bex_{Lag}, construite sur le même principe que Bex_{sk}, Bex_{kw} et Bex_{crt}, n'a pas besoin de « matériel » supplémentaire, comme le sont le canal $\mathbb{Z}/m_{sk}\mathbb{Z}$ et le Cox, et est une conversion également complète.

Catégories de faute

Une faute multiple n'est, par définition, que l'agrégation de plusieurs fautes uniques. Le contexte est celui donné par l'Algorithme 12 (. 43). Les principes fondateurs de cet algorithme et des extrapolations polynomiales mises en jeu sont similaires avec ceux régissant l'Algorithme 13. Les opérations de conversion de base utilisées sont toujours supposées soumises à l'Hypothèse 2.2. Celle-ci concerne le besoin de garantir le fait que lorsqu'une valeur est envoyée d'un canal RNS de $\mathcal{B}$ vers tous ceux de $\mathcal{B}'$ lors d'une conversion de $\mathcal{B}$ vers $\mathcal{B}'$, alors la valeur reçue doit être la même dans tous les canaux de $\mathcal{B}'$. Les fautes uniques sont donc classées comme suit, où la nouvelle catégorie 4 est en fait la catégorie 5 du cas des entiers, à savoir la catégorie des fautes sur les résidus redondants.

Catégorie 1 Se réduit à toute faute $E_i(X) \in \mathbb{F}_p[X]_{d_i}$ sur le résidu $Q_i(X)$ dans le canal $\mathbb{F}_p[X]/m_i(X)\mathbb{F}_p[X]$ de la base $\mathcal{B}$ avant la première conversion de base.

Catégorie 2 Toute faute ayant pour effet d'ajouter un polynôme d'erreur $E'_j(X) \in \mathbb{F}_p[X]_{d'_j}$ sur le résidu $S'_j(X)$ dans le canal $\mathbb{F}_p[X]/m'_j(X)\mathbb{F}_p[X]$ de la base $\mathcal{B}'$ avant la seconde conversion de base.

Catégorie 3 Ce sont les fautes $E_i(X) \in \mathbb{F}_p[X]_{d_i}$ affectant un résidu $S_i(X)$ dans le canal $\mathbb{F}_p[X]/m_i(X)\mathbb{F}_p[X]$ de la base $\mathcal{B}$ après la seconde conversion de base.

Catégorie 4 Toute faute sur un résidu redondant $S_{R,z}(X)$ avant la seconde conversion, ou sur un résidu $\widehat{S}_{R,z}(X)$ après la seconde conversion, dans le canal $\mathbb{F}_p[X]/m_{R,z}(X)\mathbb{F}_p[X]$ de la base $\mathcal{B}_R$.

Définition 3.4 *Soit $\mathcal{B}$, $\mathcal{B}'$ et $\mathcal{B}_R$ les trois bases mises en jeu dans l'Algorithme 16, et un triplet d'entiers $(c_1, c_2, c_4) \in [\![1, n]\!] \times [\![1, \ell]\!] \times [\![1, k]\!]$. Une (c_1, c_2, c_4)-faute est la donnée de trois ensembles d'indices $\mathcal{I}_{c_1} \subseteq [\![1, n]\!]$, $\mathcal{J}_{c_2} \subseteq [\![1, \ell]\!]$ et $\mathcal{Z}_{c_4} \subseteq [\![1, k]\!]$ possédant respectivement c_1, c_2 et c_4 éléments (pouvant être vides), et la donnée de c_1 polynômes $E_i(X)$ de degré au plus $d_i - 1$ pour tout $i \in \mathcal{I}_{c_1}$ (fautes uniques de catégorie 1), de c_2 polynômes $E'_j(X)$ de degré au plus $d'_j - 1$ pour tout $j \in \mathcal{J}_{c_2}$ (fautes uniques de catégorie 2), et de c_4 polynômes $E_{R,z}(X)$ de degré au plus $d_{R,z} - 1$ pour tout $z \in \mathcal{Z}_{c_4}$ (fautes uniques de catégorie 4).*

La définition précédente fournit le modèle de faute multiple dans le cadre d'une réduction modulaire. La problématique des fautes de catégorie 3 reste strictement identique à celle du cas de la réduction modulaire en RNS redondant donnée par l'Algorithme 14. Pour leur détection, il s'agit de procéder à une conversion supplémentaire pour effectuer la Procédure DetectMultErrExt sur les résidus dans la base $\mathcal{B}$ du résultat final.

3.2.4 Algorithme proposé, preuve de correction

Théorème 3.7 *L'Algorithme 16 (p. 117) de multiplication modulaire dans le corps $\mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$ détecte toute (c_1, c_2, c_4)-faute vérifiant $c_1 + c_2 + c_4 \in [\![1, k]\!]$.*

Démonstration. Soit $t \leqslant k$ un entier et soit une t-faute d'indices $\mathcal{I}_{c_1} \cup \mathcal{J}_{c_2} \cup \mathcal{Z}_{c_4}$ et de valeurs $(E_i(X))_{i\in\mathcal{I}_{c_1}}$, $\left(E'_j(X)\right)_{j\in\mathcal{J}_{c_2}}$, $(E_{R,z}(X))_{z\in\mathcal{Z}_{c_4}}$.

Par la Proposition 3.4, les fautes d'indices $\mathcal{I}_{c_1}$ sur les résidus de Q dans la base $\mathcal{B}$ définissent un polynôme $E_{\mathcal{I}_{c_1}}(X) = \varphi_{\mathcal{B}_{\mathcal{I}_{c_1}}}\left((E_i(X))_{i\in\mathcal{I}_{c_1}}\right)$ de degré au plus $\sum_{i\in\mathcal{I}_{c_1}} d_i - 1$. Soit alors le polynôme suivant :

$$E(X) = E_{\mathcal{I}_{c_1}}(X) \times \prod_{i\in[\![1,n]\!]\setminus\mathcal{I}_{c_1}} m_i(X), \text{ où } \deg(E) \leqslant \sum_{i\in\mathcal{I}_{c_1}} d_i - 1 + \sum_{i\in[\![1,n]\!]\setminus\mathcal{I}_{c_1}} d_i = d - 1.$$

De cette notation, il vient que les résidus erronés dans la base $\mathcal{B}$ avant la première conversion sont ceux du polynôme suivant :

$$\overline{Q}(X) = Q(X) + E(X). \tag{3.31}$$

Le polynôme $T(X) = A(X)B(X) + \overline{Q}(X)N(X)$ calculé dans les bases $\mathcal{B}'$ et $\mathcal{B}_R$ entre les première et seconde conversions est de degré strictement inférieur à $d + d'$. En effet,

$$\begin{aligned}
\deg(T) &\leqslant \max(\deg(A) + \deg(B), \deg(\overline{Q}) + \deg(N)) \\
&\leqslant \max(2s + 2\sigma - 2, d - 1 + s) \\
&\leqslant \max(d + s - 1, d - 1 + s) \\
&\leqslant d + d' - 1.
\end{aligned}$$

De ce fait, $T(X)$ est complètement déterminé par ses rédidus dans $\mathcal{B} \cup \mathcal{B}'$, c'est-à-dire que nous pouvons écrire $T(X) = \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}(\boldsymbol{T}_{\mathcal{B}}, \boldsymbol{T}_{\mathcal{B}'}) = \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}(\boldsymbol{E}_{\mathcal{B}} \times \boldsymbol{N}_{\mathcal{B}}, \boldsymbol{T}_{\mathcal{B}'})$.

Les valeurs des c_2-fautes d'indices $\mathcal{J}_{c_2}$ et c_4-fautes d'indices $\mathcal{Z}_{c_4}$ définissent respectivement les polynômes

$$E_{\mathcal{J}_{c_2}}(X) = \varphi_{\mathcal{B}'_{\mathcal{J}_{c_2}}}\left(\left(E'_j(X)\right)_{j\in\mathcal{J}_{c_2}}\right), \text{ de degré au plus } \sum_{j\in\mathcal{J}_{c_2}} d'_j - 1$$

et

$$E_{\mathcal{Z}_{c_4}}(X) = \varphi_{\mathcal{B}_{R,\mathcal{Z}_{c_4}}}\left((E_{R,z}(X))_{z\in\mathcal{Z}_{c_4}}\right), \text{ de degré au plus } \sum_{z\in\mathcal{Z}_{c_4}} d_{R,z} - 1.$$

Avant la seconde conversion, les résidus dans la base $\mathcal{B}'$ sont donc ceux du polynôme suivant :

$$\begin{aligned}\overline{S'}(X) &= \left|T(X)M(X)^{-1}\right|_{M'(X)} + E_{\mathcal{J}_{c_2}}(X) \times \prod_{j\in[\![1,\ell]\!]\setminus\mathcal{J}_{c_2}} m'_j(X)\\ &= \left|T(X)M(X)^{-1}\right|_{M'(X)} + E'(X).\end{aligned}$$

De la même manière dans la base $\mathcal{B}_R$:

$$\begin{aligned}\overline{S}_R(X) &= \left|T(X)M(X)^{-1}\right|_{M_R(X)} + E_{\mathcal{Z}_{c_4}}(X) \times \prod_{z\in[\![1,k]\!]\setminus\mathcal{Z}_{c_4}} m_{R,z}(X)\\ &= \left|T(X)M(X)^{-1}\right|_{M_R(X)} + E_R(X).\end{aligned}$$

La seconde conversion de base calculant les résidus dans $\mathcal{B}\cup\mathcal{B}_R$ de $\overline{S'}(X)$, la détection est assurée si, et seulement si, $\overline{S'}(X) \bmod M_R(X) \neq \overline{S}_R(X)$ ou, de manière équivalente puisque $M(X)\wedge M_R(X) = 1$, si, et seulement si,

$$\left|T(X)M(X)^{-1}\right|_{M'(X)} M(X) + E'(X)M(X) \not\equiv T(X) + E_R(X)M(X) \bmod M_R(X). \tag{3.32}$$

Le polynôme $\left|T(X)M(X)^{-1}\right|_{M'(X)} M(X) + E'_S(X)M(X)$ étant de degré $< d + d'$, il est défini par ses résidus dans la base $\mathcal{B}\cup\mathcal{B}'$. Autrement dit,

$$\left|T(X)M(X)^{-1}\right|_{M'(X)} M(X) + E'(X)M(X) = \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{T}_{\mathcal{B}'} + \boldsymbol{E}'_{\mathcal{B}'} \times \boldsymbol{M}_{\mathcal{B}'}\right).$$

L'Inéquation (3.32) se réécrit alors :

$$\begin{aligned}&\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{T}_{\mathcal{B}'} + \boldsymbol{E}'_{\mathcal{B}'} \times \boldsymbol{M}_{\mathcal{B}'}\right) \not\equiv \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{T}_{\mathcal{B}}, \boldsymbol{T}_{\mathcal{B}'}\right) + E_R(X)M(X) \bmod M_R(X)\\ &\Leftrightarrow \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{E}'_{\mathcal{B}'} \times \boldsymbol{M}_{\mathcal{B}'}\right) \not\equiv \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{E}_{\mathcal{B}} \times \boldsymbol{N}_{\mathcal{B}}, \mathbf{0}_{\mathcal{B}'}\right) + E_R(X)M(X) \bmod M_R(X).\end{aligned}$$

Comme $E_R(X)M(X) \equiv 0 \bmod m_{R,z}(X)$ pour tout $z \notin \mathcal{Z}_{c_4}$, pour garantir la détection de la faute multiple il suffit de vérifier que :

$$\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{E}'_{\mathcal{B}'} \times \boldsymbol{M}_{\mathcal{B}'}\right) \not\equiv \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{E}_{\mathcal{B}} \times \boldsymbol{N}_{\mathcal{B}}, \mathbf{0}_{\mathcal{B}'}\right) \bmod \prod_{z\notin\mathcal{Z}_{c_4}} m_{R,z}(X). \tag{3.33}$$

D'une part, comme les résidus non nuls de $\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{E}_{\mathcal{B}} \times \boldsymbol{N}_{\mathcal{B}}, \mathbf{0}_{\mathcal{B}'}\right)$ sont ceux d'indices $\mathcal{I}_{c_1}$, la Proposition 3.4 donne l'existence d'un polynôme $P_{\mathcal{I}_{c_1}}(X)$ de degré au plus $\sum_{i\in\mathcal{I}_{c_1}} d_i - 1$ tel que :

$$\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{E}_{\mathcal{B}} \times \boldsymbol{N}_{\mathcal{B}}, \mathbf{0}_{\mathcal{B}'}\right) = P_{\mathcal{I}_{c_1}}(X) \times \prod_{i\in[\![1,n]\!]\setminus\mathcal{I}_{c_1}} m_i(X) \times \prod_{j=1}^{\ell} m'_j(X).$$

D'autre part, il vient de la même manière qu'il existe un polynôme $P_{\mathcal{J}_{c_2}}(X)$ de degré au plus $\sum_{j\in\mathcal{J}_{c_2}} d'_j - 1$ tel que :

$$\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{E}'_{\mathcal{B}'} \times \boldsymbol{M}_{\mathcal{B}'}\right) = P_{\mathcal{J}_{c_2}}(X) \times \prod_{j\in[\![1,\ell]\!]\setminus\mathcal{J}_{c_2}} m'_j(X) \times \prod_{i=1}^{n} m_i(X).$$

Par suite,

$$\begin{aligned}
&\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\mathbf{0}_{\mathcal{B}}, \boldsymbol{E}'_{\mathcal{B}'} \times \boldsymbol{M}_{\mathcal{B}'}\right) - \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}'}\left(\boldsymbol{E}_{\mathcal{B}} \times \boldsymbol{N}_{\mathcal{B}}, \mathbf{0}_{\mathcal{B}'}\right) \\
&= \left(P_{\mathcal{J}_{c_2}}(X) \times \prod_{i\in\mathcal{I}_{c_1}} m_i(X) - P_{\mathcal{I}_{c_1}}(X) \times \prod_{j\in\mathcal{J}_{c_2}} m'_j(X) \right) \times \prod_{i\notin\mathcal{I}_{c_1}} m_i(X) \prod_{j\notin\mathcal{J}_{c_2}} m'_j(X) \\
&= F(X) \times \prod_{i\notin\mathcal{I}_{c_1}} m_i(X) \prod_{j\notin\mathcal{J}_{c_2}} m'_j(X).
\end{aligned}$$

Le polynôme $F(X)$ est donc de degré au plus $\sum_{i\in\mathcal{I}_{c_1}} d_i + \sum_{i\in\mathcal{J}_{c_2}} d'_j - 1$, et est non nul de l'instant où $c_1 + c_2 \neq 0$. De plus, l'Équation (3.33) est équivalente à la vérification suivante :

$$F(X) \not\equiv 0 \bmod \prod_{z\notin\mathcal{Z}_{c_4}} m_{R,z}(X).$$

Mais si cette inégalité n'est pas vérifiée, alors cela signifie que $\sum_{z\notin\mathcal{Z}_{c_4}} d_{R,z} \leqslant \sum_{i\in\mathcal{I}_{c_1}} d_i + \sum_{i\in\mathcal{J}_{c_2}} d'_j - 1$. Or par hypothèse $k - c_4 \geqslant c_1 + c_2$, et comme $\deg(m_{R,z}) \geqslant \deg(m)$ pour tout $z \in [\![1,k]\!]$ et tout $m \in \mathcal{B}\cup\mathcal{B}'$, ceci implique en particulier $\sum_{z\notin\mathcal{Z}_{c_4}} d_{R,z} \geqslant \sum_{i\in\mathcal{I}_{c_1}} d_i + \sum_{i\in\mathcal{J}_{c_2}} d'_j$. Ainsi, l'Inégalité (3.33) est bien vérifiée, et la détection assurée.

□

3.2.5 Comparaison avec la multiplication modulaire redondante de Medoš et Boztaş (2008)

Medoš et Boztaş (2008) utilisent les RNS redondants pour proposer un algorithme de multiplication modulaire de Montgomery dans $\mathbb{F}_{2^s}$, doté notamment d'une capacité de détection. Le principe de cet algorithme est aisément généralisable aux corps $\mathbb{F}_{p^s}$ pour p premier quelconque, et il est basé sur l'Algorithme 12 de réduction modulaire RNS de Montgomery dans $\mathbb{F}_{p^s}$.

L'utilisation des moduli redondants pour la détection des fautes est différente de l'approche suggérée dans ce mémoire. L'ajout de ces moduli est justifié par le fait qu'une faute multiple sur les résidus représente alors un polynôme dont le degré est attendu être supérieur ou égal à s sous certaines hypothèses.

Contexte et principe du processus de détection

Le contexte est toujours celui d'un corps $\mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$ où N est un polynôme irréductible sur $\mathbb{F}_p$ de degré s. Les éléments du corps sont représentés dans une base RNS principale $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ vérifiant $d = \sum_{i=1}^{n} \deg(m_i) = \sum_{i=1}^{n} d_i \geqslant s$ et dans une base redondante, première à $\mathcal{B}$, $\mathcal{B}_R = \{m_{R,1}(X), \ldots, m_{R,k}(X)\}$ de polynômes premiers entre eux deux à deux.

La proposition fondamentale permettant la création d'un processus de détection de fautes sur les résidus est la suivante.

Algorithme 16 : Multiplication modulaire redondante dans $\mathbb{F}_{p^s} \simeq \mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$

Données : σ un paramètre entier, $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$, $\mathcal{B}' = \{m'_1(X), \ldots, m'_\ell(X)\}$ et $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$ trois bases copremières avec $d = \deg(M) \geqslant s + 2\sigma - 1$, $d' = \deg(M') \geqslant s$ et $d_{R,z} = \deg(m_{R,z}) \geqslant \deg(m)$ pour tout $m \in \mathcal{B} \cup \mathcal{B}'$ et tout $z \in [\![1, k]\!]$; $A(X)$ et $B(X)$ deux éléments de $\mathbb{F}_p[X]_{s+\sigma}$ représentés par leurs résidus dans $\mathcal{B} \cup \mathcal{B}' \cup \mathcal{B}_R$

Résultat : les résidus dans $\mathcal{B} \cup \mathcal{B}' \cup \mathcal{B}_R$ de $S(X) = A(X)B(X)M(X)^{-1} \bmod N(X)$

début
  pour $i \leftarrow 1$ **à** n **faire**
    /* en parallèle dans $\mathcal{B}$ */
    $Q_i(X) \leftarrow -A_i(X)B_i(X)N(X)^{-1} \bmod m_i(X)$
  $\left(Q'_1(X), \ldots, Q'_\ell(X)\right), (Q_{R,1}(X), \ldots, Q_{R,k}(X)) \leftarrow$ $\mathrm{Bex}_1\left(\mathcal{B}, \mathcal{B}' \cup \mathcal{B}_R, (Q_1(X), \ldots, Q_n(X))\right)$
    /* première interpolation */
  pour $i \leftarrow 1$ **à** ℓ **faire**
    /* en parallèle dans $\mathcal{B}'$ */
    $T'_i(X) \leftarrow \left(A'_i(X)B'_i(X) + Q'_i(X)N(X)\right) \bmod m'_i(X)$
    $S'_i(X) \leftarrow T'_i(X)M(X)^{-1} \bmod m'_i(X)$
  pour $z \leftarrow 1$ **à** k **faire**
    /* en parallèle dans $\mathcal{B}_R$ */
    $T_{R,z}(X) \leftarrow (A_{R,z}(X)B_{R,z}(X) + Q_{R,z}(X)N(X)) \bmod m_{R,z}(X)$
    $S_{R,z}(X) \leftarrow T_{R,z}(X)M(X)^{-1} \bmod m_{R,z}(X)$
  $(S_1(X), \ldots, S_n(X)), \left(\widehat{S}_{R,1}(X), \ldots, \widehat{S}_{R,k}(X)\right) \leftarrow$ $\mathrm{Bex}_2\left(\mathcal{B}', \mathcal{B} \cup \mathcal{B}_R, \left(S'_1(X), \ldots, S'_\ell(X)\right)\right)$
    /* seconde interpolation */
  si $\left(\widehat{S}_{R,1}(X), \ldots, \widehat{S}_{R,k}(X)\right) \neq (S_{R,1}(X), \ldots, S_{R,k}(X))$ **alors**
    retourner `Faute détectée.`
  sinon
    retourner $(S_1(X), \ldots, S_n(X)), \left(S'_1(X), \ldots, S'_\ell(X)\right), (S_{R,1}(X), \ldots, S_{R,k}(X))$

Proposition 3.6 *Soit $A(X) \in \mathbb{F}_p[X]_s$ et (u, v) un doublet d'entiers dans $[\![1, n]\!] \times [\![1, k]\!]$ avec $u + v \leqslant k$. Toute (u, v)-faute sur les résidus $(\boldsymbol{A}_{\mathcal{B}}, \boldsymbol{A}_{\mathcal{B}_R})$ vérifie $\deg\left(\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}_R}\left(\mathcal{B}\cup\mathcal{B}_R, \left(\overline{\boldsymbol{A}}_{\mathcal{B}}, \overline{\boldsymbol{A}}_{\mathcal{B}_R}\right)\right)\right) \geqslant s$ si, pour tout ensemble de $u + v$ indices $(\mathcal{I}_u, \mathcal{Z}_v) \subset [\![1, n]\!] \times [\![1, k]\!]$,*

$$\deg\left(\prod_{i\in[\![1,n]\!]\setminus\mathcal{I}_u} m_i(X) \times \prod_{z\in[\![1,k]\!]\setminus\mathcal{Z}_v} m_{R,z}(X)\right) \geqslant s.$$

Démonstration. Soit une (u, v)-faute d'indices $(\mathcal{I}_u, \mathcal{Z}_v)$ de résidus non nuls $(E_i(X))_{i\in\mathcal{I}_u}$ et $(E_{R,z}(X))_{z\in\mathcal{Z}_v}$ affectant les résidus de $A(X) \in \mathbb{F}_p[X]_s$. Par la Proposition 3.4, il existe un polynôme non nul $E_{\mathcal{I}_u,\mathcal{Z}_v}(X)$ tel que

$$\begin{aligned}
&\varphi^{-1}_{\mathcal{B}\cup\mathcal{B}_R}\left(\left(\overline{\boldsymbol{A}}_{\mathcal{B}}, \overline{\boldsymbol{A}}_{\mathcal{B}_R}\right)\right) - A(X) = \\
&\sum_{i\in\mathcal{I}_u} \left| E_i(X) \prod_{j\in[\![1,n]\!]\setminus\mathcal{I}_u} m_j(X)^{-1} \right|_{m_i(X)} \prod_{j\in[\![1,n]\!]\setminus\mathcal{I}_u} m_j(X) \prod_{z=1}^{k} m_{R,z}(X) \\
&+ \sum_{z\in\mathcal{Z}_v} \left| E_{R,z}(X) \prod_{w\in[\![1,k]\!]\setminus\mathcal{Z}_v} m_{R,w}(X)^{-1} \right|_{m_{R,z}(X)} \prod_{w\in[\![1,k]\!]\setminus\mathcal{Z}_v} m_{R,w}(X) \prod_{i=1}^{n} m_i(X) \\
&= E_{\mathcal{I}_u,\mathcal{Z}_v}(X) \prod_{i\in[\![1,n]\!]\setminus\mathcal{I}_u} m_i(X) \prod_{z\in[\![1,k]\!]\setminus\mathcal{Z}_v} m_{R,z}(X).
\end{aligned}$$

Le résultat vient attendu immédiatement. □

La procédure de détection de fautes sur des résidus $(\boldsymbol{A}_{\mathcal{B}}, \boldsymbol{A}_{\mathcal{B}_R})$ d'un élément de $\mathbb{F}_{p^s}$ nécessite une reconstruction complète du polynôme représenté par ces résidus.

Procédure DetectErrDegre($\mathcal{B}$,$\mathcal{B}_R$,$\boldsymbol{A}_{\mathcal{B}}$,$\boldsymbol{A}_{\mathcal{B}_R}$,$s$)

1 **début**
2 $\quad A(X) \leftarrow \varphi^{-1}_{\mathcal{B}\cup\mathcal{B}_R}((\boldsymbol{A}_{\mathcal{B}}, \boldsymbol{A}_{\mathcal{B}_R}))$;
3 $\quad$ **retourner** $\deg(A(X)) \geqslant s$

Remarque 3.3 *Afin d'assurer la détection de toute t-faute avec $t \leqslant k$ et d'après la Proposition 3.6, il suffit que les deux bases $\mathcal{B} \cup \mathcal{B}_R$ vérifient :*

$$\begin{aligned}
&\forall\, (\mathcal{I}_u, \mathcal{Z}_v) \subset [\![1, n]\!] \times [\![1, k]\!], \text{ tels que } |\mathcal{I}_u| + |\mathcal{Z}_v| = k, \\
&\deg\left(\prod_{i\in[\![1,n]\!]\setminus\mathcal{I}_u} m_i(X) \times \prod_{z\in[\![1,k]\!]\setminus\mathcal{Z}_v} m_{R,z}(X)\right) \geqslant s.
\end{aligned} \tag{3.34}$$

De plus, comme $d \geqslant s$, il suffit que $\deg(m_{R,z}) \geqslant \deg(m_i)$ pour tout $(i, z) \in [\![1, n]\!] \times [\![1, k]\!]$ pour que la Condition (3.34) soit vérifiée.

Multiplication modulaire redondante et analyse comparative avec l'Algorithme 12

L'Algorithme 17 applique la procédure de détection DetectErrDegre, définie ci-avant, sur les seuls résidus $(\boldsymbol{S}_{\mathcal{B}}, \boldsymbol{S}_{\mathcal{B}_R})$. Par la Proposition 3.6, la détection de toute t-faute sur ces résidus avec $t \leqslant k$ est garantie. De plus, pour toute

Algorithme 17 : Multiplication modulaire dans $\mathbb{F}_{p^s} \simeq \mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$ de Medoš et Boztaş (2008)

Données : $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$, $\mathcal{B}' = \{m'_1(X), \ldots, m'_n(X)\}$, $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$ et $\mathcal{B}'_R = \{m'_{R,1}, \ldots, m'_{R,k}\}$ quatre bases copremières avec $d = \sum_{i=1}^{n} d_i \geqslant s$, $d = \sum_{i=1}^{n} d'_i \geqslant s$; $\mathcal{B}$ et $\mathcal{B}_R$ (resp. $\mathcal{B}'$ et $\mathcal{B}'_R$) vérifient la condition de la Proposition 3.6 ; $A(X)$ et $B(X)$ deux éléments de $\mathbb{F}_p[X]_s$ représentés par leurs résidus dans $\mathcal{B} \cup \mathcal{B}_R \cup \mathcal{B}' \cup \mathcal{B}'_R$

Résultat : les résidus dans $\mathcal{B} \cup \mathcal{B}_R \cup \mathcal{B}' \cup \mathcal{B}'_R$ de $S(X) = A(X)B(X)M(X)^{-1} \bmod N(X)$

début
- $Q_{\mathcal{B}} \leftarrow -A(X)B(X)N(X)^{-1} \bmod M(X)$
- $Q_{\mathcal{B}_R} \leftarrow -A(X)B(X)N(X)^{-1} \bmod M_R(X)$
  `/* en parallèle sur les résidus dans` $\mathcal{B} \cup \mathcal{B}_R$ `*/`
- $\left(Q_{\mathcal{B}'}, Q_{\mathcal{B}'_R}\right) \leftarrow \text{Bex}_1\left(\mathcal{B} \cup \mathcal{B}_R, \mathcal{B}' \cup \mathcal{B}'_R, (Q_{\mathcal{B}}, Q_{\mathcal{B}_R})\right)$
  `/* première interpolation */`
- $S_{\mathcal{B}'} \leftarrow \left(A(X)B(X) + Q(X)N(X)\right) M(X)^{-1}M_R(X)^{-1} \bmod M'(X)$
- $S_{\mathcal{B}'_R} \leftarrow \left(A(X)B(X) + Q(X)N(X)\right) M(X)^{-1}M_R(X)^{-1} \bmod M'_R(X)$
  `/* en parallèle sur les résidus dans` $\mathcal{B}' \cup \mathcal{B}'_R$ `*/`
- $(S_{\mathcal{B}}, S_{\mathcal{B}_R}) \leftarrow \text{Bex}_2\left(\mathcal{B}' \cup \mathcal{B}'_R, \mathcal{B} \cup \mathcal{B}_R, \left(S_{\mathcal{B}'}, S_{\mathcal{B}'_R}\right)\right)$
- **si** *DetectErrDegre* $(\mathcal{B}, \mathcal{B}_R, S_{\mathcal{B}}, S_{\mathcal{B}_R}, s)$ **alors**
  - **retourner** `Faute détectée`
- **sinon**
  - **retourner** $\left(S_{\mathcal{B}}, S_{\mathcal{B}_R}, S_{\mathcal{B}'}, S_{\mathcal{B}'_R}\right)$

t-faute multiple affectant différents résidus dans les quatre bases au cours de l'exécution de l'algorithme et ayant pour effet de produire un polynôme d'erreur de degré au moins égal à s sur les résidus $(\boldsymbol{S}_{\mathcal{B}}, \boldsymbol{S}_{\mathcal{B}_R})$ provoque le déclenchement du signal de détection.

Le modèle de faute multiple donné par la Définition 3.3 (p. 108) est peu pertinent pour analyser la capacité de détection de l'Algorithme 17. Une u-faute multiple composée de u_1 fautes uniques affectant $\boldsymbol{Q}_{\mathcal{B}}$ et u_2 fautes uniques affectant $\boldsymbol{Q}_{\mathcal{B}_R}$ transforme Q de la manière suivante :

$$\begin{cases} \overline{Q}(X) = Q(X) + E_Q(X), \\ \displaystyle\sum_{i \in \mathcal{I}_{u_1} \subset [\![1,n]\!]} d_i + \sum_{z \in \mathcal{Z}_{u_2} \subset [\![1,k]\!]} d_{R,z} \leqslant \deg(E_Q) < d + d_R. \end{cases}$$

Continuant les calculs dans la base $\mathcal{B}' \cup \mathcal{B}'_R$ il vient :

$$\overline{S}(X) = S(X) + \left| E_Q(X) N(X) M(X)^{-1} M_R(X)^{-1} \right|_{M'(X) M'_R(X)}.$$

Et si de plus une v-faute multiple composée de v_1 fautes uniques affectant $\boldsymbol{S}_{\mathcal{B}'}$ et v_2 fautes uniques affectant $\boldsymbol{S}_{\mathcal{B}'_R}$ est introduite, les résidus finaux dans la base $\mathcal{B}' \cup \mathcal{B}'_R$ avant la seconde conversion de base représentent le polynôme suivant :

$$\begin{cases} \overline{S}(X) = S(X) + E_S(X) + \left| E_Q(X) N(X) M(X)^{-1} M_R(X)^{-1} \right|_{M'(X) M'_R(X)}, \\ \displaystyle\sum_{j \in \mathcal{J}_{v_1} \subset [\![1,n]\!]} d'_j + \sum_{z \in \mathcal{Z}_{v_2} \subset [\![1,k]\!]} d'_{R,z} \leqslant \deg(E_S) < d' + d'_R. \end{cases} \tag{3.35}$$

Si le modèle de faute considéré interdit toute perturbation durant la seconde conversion de base et donc convertit exactement la quantité (3.35), alors la procédure de détection de l'Algorithme 17 vérifie le degré du polynôme suivant :

$$\left(S(X) + E_S(X) + \left| E_Q(X) N(X) M(X)^{-1} M_R(X)^{-1} \right|_{M'(X) M'_R(X)} \right) \bmod M(X) M_R(X). \tag{3.36}$$

Comme $S(X)$ est le résultat final correct, $\deg(S) < s$. Comme souligné précédemment, si le polynôme

$$\left(E_S(X) + \left| E_Q(X) N(X) M(X)^{-1} M_R(X)^{-1} \right|_{M'(X) M'_R(X)} \right) \bmod M(X) M_R(X)$$

possède au plus k résidus non nuls dans la base $\mathcal{B} \cup \mathcal{B}_R$, ces fautes seront détectées, du fait de la Proposition 3.6. Autrement, trouver des conditions sur les bases RNS utilisées garantissant la détection des fautes multiples de catégorie 1 et 2 est délicat et potentiellement restrictif sur le choix des bases.

Finalement, la procédure de détection n'étant appliquée que sur les résidus dans $\mathcal{B} \cup \mathcal{B}_R$, seule l'intégrité des résidus $\left(\boldsymbol{S}_{\mathcal{B}}, \boldsymbol{S}_{\mathcal{B}_R}, \boldsymbol{S}_{\mathcal{B}'}, \boldsymbol{S}_{\mathcal{B}'_R} \right)$ contre les t-fautes de catégorie 2 avec $t \leqslant k$ est complètement garantie en l'absence de fautes de catégorie 3, et ce dès que $d' + d'_R \leqslant d + d_R$.

Un inconvénient majeur de l'Algorithme 17 est que la procédure de détection consiste en une reconstruction complète d'un polynôme donné dans

la base $\mathcal{B} \cup \mathcal{B}_R$. Le coût engendré a donc un impact négatif important si cette multiplication modulaire est mise en œuvre dans le cadre d'une exponentiation modulaire. De plus, contrairement à la méthode proposée dans ce mémoire, les deux bases redondantes $\mathcal{B}_R$ et $\mathcal{B}'_R$ présentent le désavantage de faire partie des bases RNS destinatrices lors des conversions de base, ce qui, vu la Remarque 1.10, augmente nécessairement le temps d'exécution de la multiplication modulaire.

Enfin, si la condition sur le degré des moduli redondants est identique pour les deux méthodes, le nombre de canaux redondants dans l'Algorithme 17 est le double de celui de l'Algorithme 16.

Conclusion

La flexibilité offerte par le principe d'intégration du processus de détection de faute à la multiplication modulaire présenté dans le chapitre précédent permet une généralisation aisée à la détection de fautes multiples. La théorie est fondée sur le Théorème 3.3 (p. 95).

De même que pour les fautes uniques, une étude intégrant un modèle de fautes multiples adapté pour une implantation matérielle a permis d'exhiber des conditions sur la qualité de l'approximation utilisée par le Cox et sur les bases RNS utilisées de manière à ce que, malgré la contrainte d'un modèle de faute matérielle autorisant l'apparition d'erreurs de valeur supérieure aux moduli des bases RNS, la condition sur les moduli redondants reste identique à celle du modèle de faute théorique (cf. Théorème. 3.4 (p.105)).

Enfin, le même principe a été adapté au cas d'une arithmétique pour les corps fini de type $\mathbb{F}_{p^s}$. Le Théorème 3.5 (p. 110) fournit les conditions nécessaires et suffisantes sur la redondance pour la détection de fautes multiples dans ce contexte.

4 Contribution à une optimisation arithmétique de la cryptographie asymétrique basée sur les réseaux

Sommaire

Dans ce chapitre nous nous intéressons à la cryptographie basée sur les réseaux euclidiens. Après de premières définitions et propriétés sur les réseaux, nous détaillons plus avant le principe des cryptosystèmes de type GGH. Ceux-ci sont construits sur la recherche d'un vecteur proche, en utilisant des algorithmes comme le Round-off de Babai qui constitue alors le cœur de la fonction de déchiffrement. Cependant, les cryptanalyses successives de tels systèmes (Nguyen 1999, Nguyen et Regev 2006) ont chaque fois durci leurs conditions d'utilisation en imposant une augmentation de la dimension des réseaux utilisés pour garantir la fiabilité du système en termes de sécurité, impliquant de fait une perte d'efficacité pratique face aux primitives cryptographiques basées sur les corps finis. Dans le but d'améliorer la compétitivité de ces cryptosystèmes, l'optimisation du round-off est une question de première importance.

Dans cette optique, nous proposons dans un premier temps une version RNS-MRS de cette opération, applicable à tout réseau répondant aux contraintes imposées par le cryptosystème originel. Puis, dans un second temps, nous exposons une méthode d'accélération spécifique à certaines classes de réseaux et présentant l'avantage d'utiliser exclusivement le RNS.

4.1 Réseaux et cryptosystèmes de type GGH

Outil efficace pour la cryptanalyse (Nguyen et Stern 2001), les réseaux prennent une importance croissante en cryptographie asymétrique. La sécurité des cryptosystèmes à base de réseaux s'est initialement reposée sur la difficulté de la résolution de problèmes comme la recherche de vecteurs courts ou de vecteurs proches. En 1996, Ajtai a montré que la résolution de tels problèmes pour des instances moyennes n'est pas plus simple que celle des instances les plus difficiles, renforçant de fait les notions de sécurité utilisées dans ce domaine.

Les réseaux tiennent une place de choix en tant que solution face à des problématiques cryptographiques actuelles et futures. Dans le paradigme de l'informatique quantique, alors que les problèmes de factorisation et de logarithme discret peuvent être résolus en temps polynomial (Shor 1994), la complexité des problèmes liés aux réseaux n'est pour le moment pas remise en cause. Les réseaux offrent ainsi un cadre de choix pour une cryptographie post-quantique. Enfin, la découverte des propriétés de chiffrement complètement homomorphe offertes par les réseaux (Gentry 2009) a marqué une avancée majeure. Permettant de transposer l'exécution d'additions et de multiplications sur les chiffrés, cette propriété place ainsi la cryptographie à base de réseaux au plus près des enjeux de sécurité à l'ère du « cloud computing ».

Le principe des cryptosystèmes de type GGH est introduit en 1997 par Goldreich et al.. Ils sont construits sur le problème de la recherche d'un proche vecteur. Au fil des années, les cryptanalyses successives ont amené à modifier les paramètres de ces systèmes, notamment la dimension des réseaux utilisés, les rendant alors chaque fois plus couteux à utiliser en pratique. Néanmoins, leur principe même de fonctionnement reste d'actualité. Nous décrivons celui-ci dans cette section, tout de suite après de premières considérations sur les réseaux.

4.1.1 Définitions de base et considérations générales

Convention 4.1 *Les vecteurs sont notés en ligne. Ainsi, le produit d'une matrice* $\mathbf{M} \in \mathbb{Z}^{\mathcal{N}\times\mathcal{N}}$ *par un vecteur* $\mathbf{v} \in \mathbb{Z}^{\mathcal{N}}$ *est noté* $\mathbf{vM}$*. De plus, la norme infinie de* $\mathbf{M}$ *est alors définie par :*

$$\left\|\mathbf{M}^{-1}\right\|_{\infty} = \max_{j\in[\![1,\mathcal{N}]\!]} \left(\sum_{i=1}^{\mathcal{N}} \left|\mathbf{M}_{i,j}\right|\right). \tag{4.1}$$

Définition 4.1 *Soit* $\mathcal{N}$ *un entier positif non nul. Un réseau* $\mathcal{R}$ *dans* $\mathbb{R}^{\mathcal{N}}$ *est un sous-groupe discret du groupe additif* $\left(\mathbb{R}^{\mathcal{N}}, +\right)$.

Un tel réseau $\mathcal{R}$ est engendré comme $\mathbb{Z}$-module par une famille libre de n vecteurs $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ de $\mathbb{R}^{\mathcal{N}}$ (en particulier, $n \leqslant \mathcal{N}$). Autrement dit, tout élément $\mathbf{v}$ de $\mathcal{R}$ s'écrit comme une combinaison linéaire des vecteurs $\mathbf{b}_1, \ldots, \mathbf{b}_n$. Ou encore, $\mathcal{R}$ est égal à la somme directe $\mathbf{b}_1\mathbb{Z} \oplus \ldots \oplus \mathbf{b}_n\mathbb{Z}$. $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ est appelée base du réseau $\mathcal{R}$.

Si la donnée d'une base permet donc de décrire entièrement le réseau associé, tout réseau possédant une base d'au moins deux éléments en possède en fait une infinité, ayant pour propriété commune de contenir le même nombre d'éléments.

Proposition 4.1 *Si $\mathcal{R} = \mathbf{b}_1\mathbb{Z} \oplus \ldots \oplus \mathbf{b}_n\mathbb{Z} \subseteq \mathbb{Z}^{\mathcal{N}}$ pour une famille libre de vecteurs $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$, alors toute autre base de $\mathcal{R}$ contient exactement n vecteurs. n est appelé rang de $\mathcal{R}$, et noté* $\mathrm{rg}\,(\mathcal{R})$.

Par convention, tout réseau $\mathcal{R}$ de $\mathbb{R}^{\mathcal{N}}$ considéré par la suite sera supposé être de rang plein : $\mathrm{rg}\,(\mathcal{R}) = \mathcal{N}$.

Définition 4.2 *Soit $\mathcal{R}$ un réseau de $\mathbb{R}^{\mathcal{N}}$ et $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_{\mathcal{N}}\}$ une base de $\mathcal{R}$. Le domaine fondamental de $\mathcal{R}$ par rapport à la base B est l'ensemble suivant :*

$$\mathcal{D}_{\mathbf{B}} = \{\sum_{i=1}^{\mathcal{N}} \beta_i \mathbf{b}_i \mid (\beta_1, \ldots, \beta_{\mathcal{N}}) \in [0,1[^{\mathcal{N}}\}.$$

Le volume fondamental associé à la base $\mathbf{B}$ est par définition $\mathcal{V}_{\mathbf{B}} = |\det(\mathbf{b}_1, \ldots, \mathbf{b}_{\mathcal{N}})|$.

Le volume fondamental ne dépend pas de la base choisie. Noté $\mathcal{V}_{\mathcal{R}}$, c'est un invariant du réseau considéré. Cela se traduit sur les bases par le fait que pour tout doublet de bases $(\mathbf{R}, \mathbf{B})$ de $\mathcal{R}$, $\mathbf{B}\mathbf{R}^{-1}$ est une matrice unimodulaire.

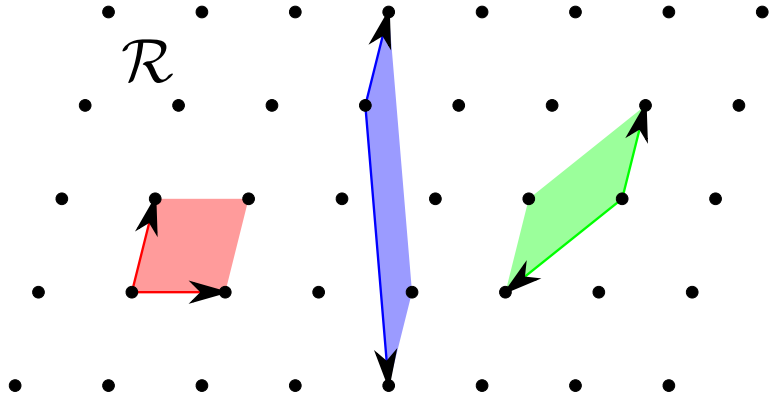


FIGURE 4.1 – *Trois bases d'un réseau $\mathcal{R}$ de $\mathbb{R}^2$ et les domaines fondamentaux associés.*

Nous introduisons la définition de la forme normale de Hermite (FNH) d'une matrice, à laquelle nous nous réfèrerons par la suite lorsque nous décrirons l'utilisation qui en est faite par Micciancio (2001) pour créer une variante du GGH.

Définition 4.3 *Soit $\mathbf{H} \in \mathbb{Z}^{\mathcal{N}}$ une matrice entière. Alors $\mathbf{H}$ est sous forme normale de Hermite si elle satisfait les conditions suivantes :*

- $\mathbf{H}_{i,i} \geqslant 0$ *pour tout* $i \in [\![1, \mathcal{N}]\!]$,
- $0 \leqslant \mathbf{H}_{i,j} < \mathbf{H}_{i,i}$ *pour tout* $j < i$,
- $\mathbf{H}_{i,j} = 0$ *pour tout* $j > i$.

Toute matrice $\mathbf{B}$ peut être transformée en une matrice $\mathbf{H} = FNH(\mathbf{B})$ sous forme normale de Hermite par des combinaisons linéaires sur ses lignes et celle-ci, telle que décrite par la définition précédente, est unique. Autrement dit, il existe une unique matrice unimodulaire $\mathbf{U} \in GL_{\mathcal{N}}(\mathbb{Z})$ pour laquelle $\mathbf{UB}$ est sous forme normale de Hermite. De ce fait, nous déduisons que si $\mathbf{R}$ et $\mathbf{B}$ sont deux bases de $\mathcal{R}$, alors il existe un unique doublet de matrices unimodulaires $\mathbf{U}_{\mathbf{R}}$ et $\mathbf{U}_{\mathbf{B}}$ tel que $FNH(\mathbf{R}) = \mathbf{U}_{\mathbf{R}}\mathbf{R}$ et $FNH(\mathbf{B}) = \mathbf{U}_{\mathbf{B}}\mathbf{B}$. De plus,

nous avons vu précédemment que $\mathbf{B}\mathbf{R}^{-1} = \mathbf{U}$ est aussi une matrice unimodulaire. Par suite, $FNH(\mathbf{B}) = \mathbf{U_B}\mathbf{B} = \mathbf{U_B}\mathbf{U}\mathbf{R}$, avec $\mathbf{U_B}\mathbf{U} \in GL_{\mathcal{N}}(\mathbb{Z})$. Ainsi, $FNH(\mathbf{R}) = \mathbf{U_R}\mathbf{U}^{-1}{\mathbf{U_B}}^{-1}FNH(\mathbf{B})$, ce qui implique donc que $\mathbf{U_R}\mathbf{U}^{-1}{\mathbf{U_B}}^{-1} = \mathbf{I}$ et $FNH(\mathbf{R}) = FNH(\mathbf{B})$. Par conséquent, chaque réseau possède une unique base sous forme normale de Hermite telle que définie précédemment.

Lorsque $\mathcal{R}$ est de rang plein, comme nous le supposerons systématiquement, alors à toute base $\mathbf{B}$ de $\mathcal{R}$ donnée, $\mathbb{R}^{\mathcal{N}}$ se partitionne de la manière suivante : $\mathbb{R}^{\mathcal{N}} = \bigcap_{\mathbf{v}\in\mathcal{R}} (\mathbf{v} + \mathcal{D}_{\mathbf{B}})$. Ainsi, tout vecteur $\mathbf{c} \in \mathbb{R}^{\mathcal{N}}$ s'écrit de manière unique comme une somme $\mathbf{v} + \mathbf{c_B}$ où $\mathbf{v} \in \mathcal{R}$ et $\mathbf{c_B} \in \mathcal{D}_{\mathbf{B}}$. $\mathbf{c_B}$ est la réduction de $\mathbf{c}$ modulo $\mathbf{B}$. Il en découle naturellement une relation d'équivalence, notée $\equiv_{\mathbf{B}}$. $\mathcal{D}_{\mathbf{B}}$ est par conséquent un ensemble complet de représentants pour le quotient $\mathbb{R}^{\mathcal{N}}_{\mathbf{B}} = \mathbb{R}^{\mathcal{N}}/\equiv_{\mathbf{B}}$. Cette réduction se calcule simplement lorsque $\mathbf{B}$ est sous forme normale de Hermite (cf. Algorithme 18), puisqu'il s'agit d'une simple application du principe du pivot de Gauss. Le vecteur $\mathbf{r}$ retourné par l'Algorithme 18 est un vecteur dont les coefficients vérifient alors $0 \leqslant \mathbf{r}_i < \mathbf{B}_{i,i}$.

Algorithme 18 : Réduction modulo une matrice FNH

Données : Une base $\mathbf{B}$ d'un réseau $\mathcal{R}$ de $\mathbb{R}^{\mathcal{N}}$ sous forme normale de Hermite (Déf. 4.3), et un vecteur $\mathbf{c} \in \mathbb{R}^{\mathcal{N}}$.
Résultat : $\mathbf{c_B} \in \mathcal{D}_{\mathbf{B}}$.

```
début
    r ← c
    pour i = N à 1 faire
        r ← r − ⌊r_i / B_{i,i}⌋ × B_{i,*}
    retourner r
```

Exemple 4.1 *Soit la matrice sous forme normale de Hermite telle que décrite par la Définition 4.3 suivante :*

$$\mathbf{B} = \begin{pmatrix} 3 & 0 & 0 \\ 1 & 7 & 0 \\ 2 & 2 & 5 \end{pmatrix}$$

et le vecteur $\mathbf{c} = \begin{pmatrix} 41 & 22 & -37 \end{pmatrix}$. *L'exécution de l'Algorithme 18 donne alors* $\mathbf{c_B} = \begin{pmatrix} 1 & 3 & 3 \end{pmatrix}$. *Ainsi,* $\mathbf{c}$ *se décompose comme suit :*

$$\mathbf{c} = \mathbf{c_B} + \begin{pmatrix} 40 & 19 & -40 \end{pmatrix} = \mathbf{c_B} + \begin{pmatrix} 17 & 5 & -8 \end{pmatrix} \times \mathbf{B}.$$

Parmi les problèmes classiques lié aux réseaux, la recherche d'un vecteur court ou celle d'un vecteur de $\mathcal{R}$ proche d'un vecteur $\mathbf{c}$ donné sont très utilisés en cryptographie. Nous donnons une définissons de ces classes de problèmes.

Définition 4.4 *Soit* $\mathcal{R}$ *un réseau de* $\mathbb{R}^{\mathcal{N}}$.

- *γ-Closest Vector Problem (γ-CVP) : à* $\mathbf{c} \in \mathbb{R}^{\mathcal{N}}$ *donné, trouver un vecteur* $\mathbf{v}$ *de* $\mathcal{R}$ *tel que pour tout* $\mathbf{w} \in \mathcal{R}$, $\|\mathbf{v} - \mathbf{c}\| \leqslant \gamma \|\mathbf{w} - \mathbf{c}\|$.
- *γ-Shortest Vector Problem (γ-SVP) : si* $\lambda_1(\mathcal{R})$ *désigne le minimum* $\min\{\|\mathbf{w}\| \mid \mathbf{w} \in \mathcal{R}\backslash\{0\}\}$, *trouver un vecteur non nul* $\mathbf{v}$ *de* $\mathcal{R}$ *tel que* $\|\mathbf{v}\| \leqslant \gamma \times \lambda_1(\mathcal{R})$.

Si ces deux problèmes semblent intuitivement très proches, le SVP ne présente pas la même homogénéité que le CVP puisque le cas du vecteur nul est écarté concernant la recherche d'un vecteur court. Goldreich et al. (1999) ont montré que résoudre γ-SVP n'est pas plus difficile que résoudre γ-CVP, quel que soit γ.

L'utilisation des réseaux en cryptographie asymétrique s'est intensifiée grâce à des résultats comme celui de Ajtai (1996), réduisant la difficulté calculatoire des instances les plus difficiles (« worst-case ») de problèmes classiques comme la recherche d'un vecteur court à celle d'instances choisies selon une certaine distribution de probabilité (« average-case »). Se donner un algorithme probabiliste polynomial en temps résolvant avec une probabilité $> \frac{1}{2}$ les instances randomisées permet une résolution probabiliste polynomiale en temps des pires cas. La complexité de ce genre de problème et cette connexion worst-case/average-case ont motivé le développement de la cryptographie basée sur les réseaux. Résoudre exactement γ-CVP a été par exemple prouvé NP-difficile pour $\gamma = 1$ par van Emde-Boas (1981), et plus généralement pour $\gamma = \mathcal{N}^{\frac{c}{\log\log\mathcal{N}}}$, $c > 0$ constant (Dinur et al. 2003).

Babai (1986) propose deux algorithmes, qui permettent de résoudre certaines classes du γ-CVP, dits « nearest plane » et « rounding-off ». Lorsque le réseau $\mathcal{R}$ considéré est décrit par une base $\mathbf{B}$, le round-off, décrit par l'Algorithme 19, consiste en un simple changement de base via une multiplication par $\mathbf{B}^{-1}$ qui envoie ainsi $\mathcal{R}$ sur le réseau $\mathbb{Z}^{\mathcal{N}}$. En écrivant le vecteur $\mathbf{c}$ comme $\mathbf{c_B} + \mathbf{v}$ où $\mathbf{v} \in \mathcal{R}$ et $\mathbf{c_B} \in \mathcal{D}_{\mathbf{B}}$, $\mathbf{cB}^{-1}$ est un vecteur appartenant au parallélotope $\mathbf{v} + [0,1[^{\mathcal{N}}$. La procédure de Babai retourne alors le sommet de ce parallélotope le plus proche de $\mathbf{cB}^{-1}$. L'efficacité de cette approche est liée au degré d'orthogonalité de la base $\mathbf{B}$, mesuré par le produit $(\det \mathbf{B})^{-1} \times \prod_{i=1}^{\mathcal{N}} \|\mathbf{b}_i\|$ appelé défaut d'orthogonalité. De par l'inégalité de Hadamard, ce produit est un réel supérieur ou égal à 1, et valant 1 si, et seulement si, les vecteurs de la base sont orthogonaux deux à deux. L'exemple suivant illustre la relative inefficacité du round-off dans le cas d'une base $\mathbf{B}$ peu orthogonale.

Exemple 4.2 *Soit le réseau $\mathcal{R}$ de $\mathbb{R}^2$ défini par la base $\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}$, d'inverse $\mathbf{B}^{-1} = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix}$. La Figure 4.2 illustre le réseau $\mathcal{R}$, ainsi que le parallélotope fondamental donné par la base $\mathbf{B}$ choisie et contenant un vecteur $\mathbf{c} = \begin{pmatrix} \frac{7}{4} & \frac{3}{8} \end{pmatrix}$ (point bleu) dont le plus proche vecteur de $\mathcal{R}$ est noté $\mathbf{v}$. La figure du bas montre l'effet de la multiplication par $\mathbf{B}^{-1}$. $\mathbf{c}$ a pour image $\mathbf{cB}^{-1} = \begin{pmatrix} \frac{5}{8} & \frac{3}{8} \end{pmatrix}$. Comme $\mathbf{B}$ est peu orthogonale, le round-off ne renvoie pas dans ce cas le plus proche vecteur $\mathbf{v}$.*

Des algorithmes à complexité polynomiale de réduction de réseau, comme LLL (Lenstra et al. 1982) et BKZ (Schnorr 1987), permettent à partir d'une base donnée de calculer une nouvelle base possédant un moindre défaut d'orthogonalité, et donc de meilleure qualité pour la résolution des problèmes de type CVP et SVP. Par exemple, l'algorithme du round-off 19 appliqué à un réseau $\mathcal{R}$ de $\mathbb{R}^{\mathcal{N}}$ donné par une base LLL-réduite calcule en temps polynomial une solution au γ-CVP pour $\gamma = 1 + 2\mathcal{N}\left(\frac{9}{2}\right)^{\mathcal{N}/2}$.

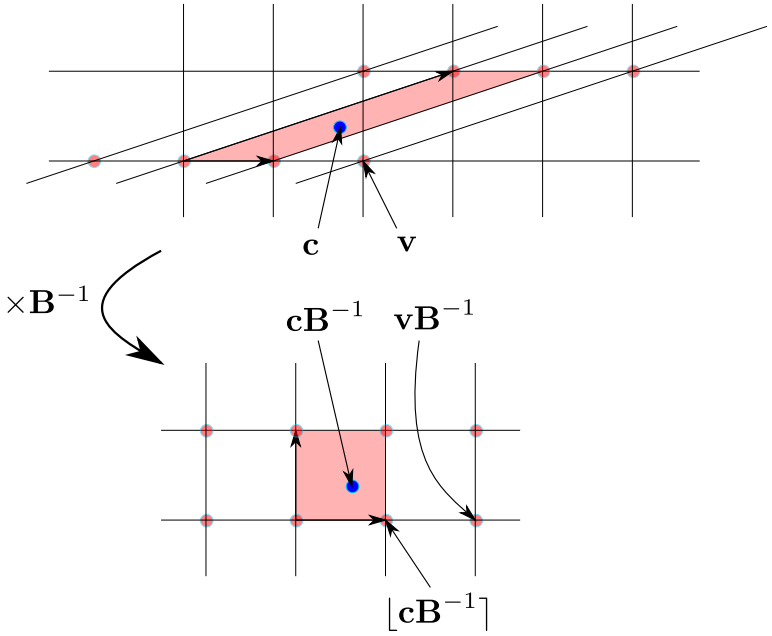


FIGURE 4.2 – *Application de la méthode du round-off dans le cas d'une base peu orthogonale.*

Algorithme 19 : RoundOff $(\mathbf{R}, \mathbf{c})$

Données : Un réseau $\mathcal{R}$ de $\mathbb{R}^{\mathcal{N}}$, $\mathbf{B}$ une base $\mathcal{R}$, et $\mathbf{c} \in \mathbb{R}^{\mathcal{N}}$ un vecteur.
Résultat : $\mathbf{v} \in \mathcal{R}$ un vecteur proche $\mathbf{c}$.

```
1 début
2     t ← ⌊cB^{-1}⌉
3     v ← tB
4     retourner v
```

4.1.2 Les cryptosystèmes de type GGH

Goldreich et al. (1997) ont proposé un type de cryptosystème asymétrique basé sur le CVP, mais ne disposant pas de preuve de sécurité à ce jour. Le principe est décrit ci-après.

Cryptosystème GGH

- ***[Génération des clefs]*** Pour un paramètre de sécurité $1^{\mathcal{N}}$ donné,
  - générer une matrice $\mathbf{R} \in M_{\mathcal{N}}(\mathbb{Z})$ de rang plein possédant de bonnes propriétés d'orthogonalité décrites par un paramètre $\sigma \in \mathbb{N}$, et définissant un réseau nommé $\mathcal{R}$;
  - générer une seconde base $\mathbf{B}$ de $\mathcal{R}$ peu orthogonale via des opérations élémentaires sur les lignes de $\mathbf{R}$, *i.e.* $\mathbf{B} = \mathbf{U}\mathbf{R}$ avec $\mathbf{U} \in GL_{\mathcal{N}}(\mathbb{Z})$;
  - renvoyer $(pk, sk) = \big(\{\mathbf{B}, \sigma\}, \{\mathbf{R}, \mathbf{R}^{-1}\}\big)$.
- ***[Fonction de chiffrement]*** Pour un message $\mathbf{p} \in \mathcal{P}_\sigma = [\![-\sigma, \sigma]\!]^{\mathcal{N}}$,
  - générer aléatoirement $\mathbf{k} \xleftarrow{f} \mathbb{Z}^{\mathcal{N}}$ selon une distribution f ;
  - calculer $\mathbf{c} \leftarrow \mathbf{p} + \mathbf{k}\mathbf{B}$;
  - renvoyer $\mathbf{c}$.
- ***[Fonction de déchiffrement]*** Pour un chiffré $\mathbf{c}$,
  - calculer $\mathbf{p} \leftarrow \mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}$ (cf. Algo. 19) ;
  - renvoyer $\mathbf{p}$.

Étant donné un réseau $\mathcal{R}$, la brèche secrète est une base $\mathbf{R}$ possédant de bonnes propriétés liées à son orthogonalité, qui autorisent un déchiffrement en utilisant le round-off. La clef publique est une autre base $\mathbf{B}$ à fort défaut d'orthogonalité. Chiffrer un message $\mathbf{p}$ consiste à lui ajouter un vecteur du réseau choisi aléatoirement selon une distribution de probabilité f. Dans la proposition originelle, le vecteur $\mathbf{k}$ est choisi uniformément dans $[\![-\mathcal{N}, \mathcal{N}]\!]^{\mathcal{N}}$. Ainsi, au message $\mathbf{p}$ est ajouté un vecteur du réseau $\mathcal{R}$ via la fonction de chiffrement suivante :

$$\begin{array}{rccl} \mathrm{C}_{\mathcal{R}} : & \mathcal{P}_\sigma & \to & \mathbb{Z}^{\mathcal{N}} \\ & \mathbf{p} & \mapsto & \mathbf{c} = \mathbf{p} + \mathbf{k}\mathbf{B},\ \mathbf{k} \xleftarrow{f} \mathbb{Z}^{\mathcal{N}}. \end{array} \tag{4.2}$$

Le fonction de déchiffrement retourne un vecteur du réseau $\mathcal{R}$ proche de $\mathbf{c}$ par la méthode du round-off, puis le soustrait au chiffré :

$$\begin{array}{rccl} \mathrm{D}_{\mathcal{R}} : & \mathbb{Z}^{\mathcal{N}} & \to & \mathbb{Z}^{\mathcal{N}} \\ & \mathbf{c} & \mapsto & \mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \times \mathbf{R}. \end{array} \tag{4.3}$$

Le paramètre σ doit garantir que $D_{\mathcal{R}} \circ C_{\mathcal{R}} = \mathrm{Id}_{\mathcal{P}_\sigma}$. Pour ce faire, la taille de ce paramètre est fixée par la condition dite de Babai pour la base $\mathbf{R}$:

$$\rho_{\mathbf{R}} \in]0, \frac{1}{2\sigma}[\tag{4.4}$$

où par définition $\rho_{\mathbf{R}} = \left\| \mathbf{R}^{-1} \right\|_\infty = \max\limits_{i=1,..,\mathcal{N}} \sum\limits_{j=1}^{\mathcal{N}} \left| (\mathbf{R}^{-1})_{i,j} \right|$. La raison de cette condition apparaît clairement dans la démonstration de la Proposition 4.2 (cf. Inégalités (4.6)).

Cette condition s'explique par le fait que la fonction d'arrondi permet de retrouver le vecteur de $\mathcal{R}$ le plus proche de tout vecteur de l'ensemble suivant :

$$\{\mathbf{r} + \mathbf{p}\mathbf{R}^{-1} \mid \mathbf{r} \in \mathcal{R},\ \mathbf{p} \in \left[B_\infty\left(0, \frac{1}{2}\right) \times \mathbf{R}\right] \cap \mathbb{Z}^{\mathcal{N}}\}. \tag{4.5}$$

Le paramètre σ est alors naturellement choisi comme étant le plus grand entier vérifiant :

$$\mathcal{P}_\sigma \subseteq B_\infty\left(0, \frac{1}{2}\right) \times \mathbf{R}.$$

Proposition 4.2 *Sous l'hypothèse (4.4), alors $D_{\mathcal{R}} \circ C_{\mathcal{R}} = \mathrm{Id}_{\mathcal{P}_\sigma}$. Autrement dit :*

$$\forall \mathbf{p} \in \mathcal{P}_\sigma = [\![-\sigma, \sigma]\!]^{\mathcal{N}},\ \mathrm{D}_{\mathcal{R}} \circ \mathrm{C}_{\mathcal{R}}(\mathbf{p}) = \mathbf{p}.$$

Démonstration. Pour $\mathbf{c} = \mathrm{C}_{\mathcal{R}}(\mathbf{p}) = \mathbf{p} + \mathbf{k}\mathbf{B}$, comme $\mathbf{k}\mathbf{U}$ est un vecteur entier, nous avons $\lfloor \mathbf{k}\mathbf{U} \rceil = \mathbf{k}\mathbf{U}$. De plus, la Condition (4.4) implique $\lfloor \mathbf{p}\mathbf{R}^{-1} \rceil = (0, \ldots, 0)$. En effet, la Condition 4.4 permet d'écrire les inégalités suivantes pour tout $j \in [\![1, \mathcal{N}]\!]$:

$$\left|\sum_{i=1}^{\mathcal{N}} \mathbf{p}_i \left(\mathbf{R}^{-1}\right)_{i,j}\right| \leqslant \sigma \times \sum_{i=1}^{\mathcal{N}} \left|\left(\mathbf{R}^{-1}\right)_{i,j}\right| \leqslant \sigma \times \left\|\mathbf{R}^{-1}\right\|_\infty = \sigma \times \rho_{\mathbf{R}} < \frac{1}{2}. \tag{4.6}$$

Par conséquent $\lfloor \mathbf{p}\mathbf{R}^{-1} \rceil = 0$ et donc :

$$\begin{aligned}
\mathrm{D}_{\mathcal{R}}(\mathbf{c}) &= \mathbf{p} + \mathbf{k}\mathbf{B} - \lfloor \mathbf{p}\mathbf{R}^{-1} + \mathbf{k}\mathbf{U}\mathbf{R}\mathbf{R}^{-1} \rceil \times \mathbf{R} \\
&= \mathbf{p} + \mathbf{k}\mathbf{B} - \lfloor \mathbf{p}\mathbf{R}^{-1} \rceil \times \mathbf{R} + \mathbf{k}\mathbf{U}\mathbf{R} \\
&= \mathbf{p} + \mathbf{k}\mathbf{B} - \mathbf{k}\mathbf{B} \\
&= \mathbf{p}.
\end{aligned}$$

□

Exemple 4.3 *Soit le réseau $\mathcal{R}$ de $\mathbb{R}^2$ défini par la base $\mathbf{R} = \begin{pmatrix} 3 & 8 \\ 8 & -2 \end{pmatrix}$. La norme infinie de l'inverse $\mathbf{R}^{-1} = \frac{1}{70}\begin{pmatrix} 2 & 8 \\ 8 & -3 \end{pmatrix}$ étant $\rho_{\mathbf{R}} = \frac{11}{70}$, le paramètre σ peut être fixé à 3.*

La Figure 4.3 montre dans un premier temps (figure du haut) les deux vecteurs (en rouge) de la base $\mathbf{R}$ du réseau dans la grille $\mathbb{Z}^2$, centrés sur un vecteur $\mathbf{k}\mathbf{B}$ où $\mathbf{B} = \mathbf{U}\mathbf{R}$ est une autre base quelconque de $\mathcal{R}$ et $\mathbf{k}$ est un vecteur entier.

L'ensemble des points bleus de la grille représente l'espace des textes clairs $\mathcal{P}_3 = [\![-3, 3]\!]^2$ centré sur $\mathbf{k}\mathbf{B}$. La condition de Babai (4.4) est illustrée par le fait que $\mathcal{P}_3$ est contenu dans $B_\infty\left(\mathbf{k}\mathbf{B}, \frac{1}{2}\right) \times \mathbf{R}$. Le parallélogramme vert représente $S_\infty\left(\mathbf{k}\mathbf{B}, \frac{1}{2}\right) \times \mathbf{R}$. Au final, tout point bleu est de la forme $\mathbf{p} + \mathbf{k}\mathbf{B}$ avec $\mathbf{p} \in \mathcal{P}_3$.

L'exemple d'un chiffré $\mathbf{c} = \mathrm{C}_{\mathcal{R}}((3,1)) = (3,1) + \mathbf{k}\mathbf{B}$ apparaît dans la figure.

La Figure 4.3 montre dans un second temps (figure du bas) le résultat de la transformation de l'espace $\mathbb{R}^2$ par la multiplication par $\mathbf{R}^{-1}$, transposant $\mathcal{R}$ sur $\mathbb{Z}^2$. Le parallélogramme vert représente donc cette fois la sphère $S_\infty\left(\mathbf{k}\mathbf{U}, \frac{1}{2}\right)$, à l'intérieur de laquelle est contenu $\mathbf{c}\mathbf{R}^{-1} = (3,1)\,\mathbf{R}^{-1} + \mathbf{k}\mathbf{B}\mathbf{R}^{-1}$. Ainsi $\lfloor (3,1)\,\mathbf{R}^{-1} \rceil = (0,0)$, et donc $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil = \mathbf{k}\mathbf{B}\mathbf{R}^{-1} = \mathbf{k}\mathbf{U} \in \mathbb{Z}^2$.

Ainsi, le calcul de $\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}$ retourne $\mathbf{k}\mathbf{B}$, le vecteur du réseau le plus proche de $\mathbf{c}$.

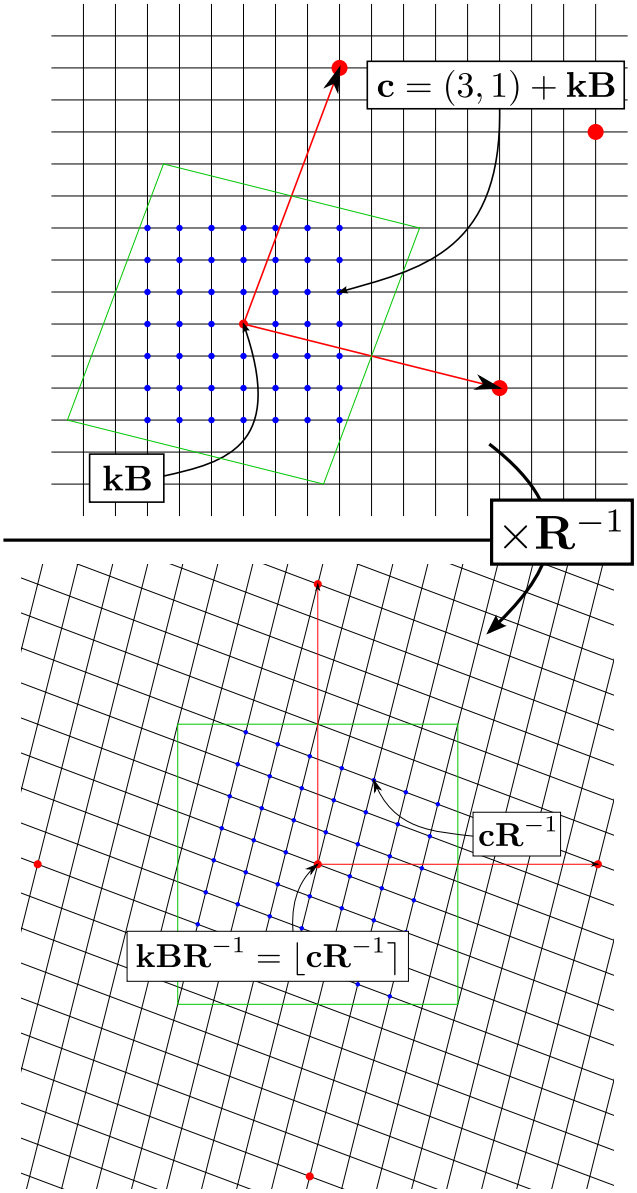


FIGURE 4.3 – *Illustration de l'Exemple 4.3.*

Dans la proposition originelle de Goldreich, Goldwasser, et Halevi (1997), afin de produire une base $\mathbf{R}$ suffisamment orthogonale, la matrice est choisie de la forme $\ell\sqrt{\mathcal{N}}\mathbf{I} + \mathbf{M}$ où $\mathbf{M}$ est tirée uniformément sur $[\![-\ell, \ell]\!]^{\mathcal{N}}$ suivant un paramètre empirique ℓ. De plus, afin que les chiffrés soient le moins proche possible d'un vecteur du réseau et que l'instance du CVP sous-jacente soit *a priori* plus difficile, $\mathcal{P}_\sigma$ est réduit à l'ensemble $\{-\sigma, \sigma\}$. Il est toujours possible de choisir σ plus grand que la limite donnée par la condition de Babai, mais cela engendre de possibles erreurs de déchiffrement. Goldreich et al. prouvent à ce sujet le théorème suivant :

Théorème 4.1 *En notant* $\frac{\gamma}{\mathcal{N}} = \max_{1 \leqslant i,j \leqslant \mathcal{N}} |\mathbf{R}_{i,j}|$, *alors à* σ *donné la probabilité d'erreur lors du déchiffrement est majorée par* $2\mathcal{N} \times \exp\left(-\frac{1}{8\sigma^2\gamma^2}\right)$.

Le schéma originel de ce cryptosystème présente de graves faiblesses, venant d'une part de la grande différence de taille entre la « perturbation » $\mathbf{p}$ et le

vecteur $\mathbf{kB}$ du réseau auquel cette perturbation est ajoutée, rendant l'instance CVP plus facile que la moyenne, et d'autre part par la forme de la perturbation lorsque celle-ci est choisie dans $\{-\sigma, \sigma\}$. Nguyen (1999) a montré que la forme particulière de $\mathbf{p}$ dans ce cas permet de se ramener à la résolution d'un CVP encore plus simple. Son attaque est basée sur la constatation que $\mathbf{p} + (\sigma, \ldots, \sigma) \in 2\sigma\mathbb{Z}^{\mathcal{N}}$. Ainsi, $\mathbf{c} + (\sigma, \ldots, \sigma) \equiv \mathbf{kB} \bmod (2\sigma)$. S'il est possible d'inverser cette équation dans $\mathbb{Z}/(2\sigma)\mathbb{Z}$, alors nous obtenons $\mathbf{k}_{2\sigma} = |\mathbf{k}|_{2\sigma}$. En notant $\mathbf{k} = 2\sigma\mathbf{k}' + \mathbf{k}_{2\sigma}$, alors le vecteur rationnel $\frac{\mathbf{c}-\mathbf{k}_{2\sigma}\mathbf{B}}{2\sigma} = \frac{\mathbf{p}}{2\sigma} + \mathbf{k}'\mathbf{B}$ est connu, et le problème qui se pose désormais est d'en trouver le plus proche vecteur de $\mathcal{R}$, soit $\mathbf{k}'\mathbf{B}$ modifié par la perturbation $\frac{\mathbf{p}}{2\sigma}$, dont la taille est désormais plus petite que la perturbation $\mathbf{p}$ originelle. En utilisant cette faille, Nguyen réussit à résoudre les challenges proposés par Goldreich et al. pour les dimensions 200, 250, 300, 350, échouant cependant à la dimension 400. La viabilité en terme de sécurité du cryptosystème GGH nécessite par conséquent l'utilisation de réseaux de grande taille, impliquant par suite une difficile mise en œuvre pratique à cause notamment de tailles de clef conséquentes.

Afin de réduire cet effet, Micciancio (2001) propose une nouvelle version de GGH. Pour réduire la taille de la clef publique, il utilise la forme normale de Hermite de la clef secrète $\mathbf{R}$ comme clef publique. L'impact de ce choix sur la sécurité comparativement au schéma originel est nul puisque de toute manière la forme normale de Hermite d'un réseau est unique et que son calcul s'effectue en temps polynomial. La fonction de chiffrement devient une réduction modulo $\mathbf{B}$ (cf. Algo. 18). Par conséquent, la génération de la clef publique ainsi que le chiffrement sont déterministes. Le principe du déchiffrement utilisant le round-off peut être conservé, puisque la réduction de $\mathbf{p}$ modulo $\mathbf{B}$ consiste toujours en un ajout d'un vecteur du réseau.

Cryptosystème GGH-FNH

- ***[Génération des clefs]*** Pour un paramètre de sécurité $1^{\mathcal{N}}$ donné,
  - générer une matrice $\mathbf{R} \in M_{\mathcal{N}}(\mathbb{Z})$ de rang plein possédant de bonnes propriétés d'orthogonalité décrites par un paramètre $\sigma \in \mathbb{N}$, et définissant un réseau nommé $\mathcal{R}$;
  - générer une seconde base $\mathbf{B}$ de $\mathcal{R}$ par $\mathbf{B} = FNH(\mathbf{R})$;
  - renvoyer $(pk, sk) = \big(\{\mathbf{B}, \sigma\}, \{\mathbf{R}, \mathbf{R}^{-1}\}\big)$.
- ***[Fonction de chiffrement]*** Pour un message $\mathbf{p} \in \mathcal{P}_\sigma = [\![-\sigma, \sigma]\!]^{\mathcal{N}}$,
  - calculer $\mathbf{c} \leftarrow \mathbf{p} \bmod \mathbf{B}$;
  - renvoyer $\mathbf{c}$.
- ***[Fonction de déchiffrement]*** Pour un chiffré $\mathbf{c}$,
  - calculer $\mathbf{p} \leftarrow \mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}$;
  - renvoyer $\mathbf{p}$.

La génération de clef secrète suggérée par Micciancio est différente de celle

proposée par Goldreich et al.. Cette dernière était créée de la forme $\ell\sqrt{\mathcal{N}}\mathbf{I} + \mathbf{M}$ de manière à présenter de bonnes propriétés d'orthogonalité. Cependant, arguant que le fait de contraindre les vecteurs de la base le long des axes principaux pourrait affaiblir le cryptosystème, Micciancio propose d'utiliser comme clef secrète la réduction LLL d'une matrice tirée aléatoirement et uniformément dans $[\![-\mathcal{N}, \mathcal{N}]\!]^{\mathcal{N}^2}$ par exemple.

Ces nouvelles dispositions permettent de diminuer les complexités asymptotiques spatiale et temporelle d'un facteur de $\mathcal{O}(\mathcal{N})$, réduisant ainsi les coûts pour une implantation pratique créés par le besoin de travailler en très grande dimension $\mathcal{N}$ pour atténuer les faiblesses inhérentes à GGH.

Rose, Plantard, et Susilo (2011) proposent une variante du GGH-FNH dans laquelle la génération de clef secrète proposée permet de se passer d'une coûteuse réduction LLL comme suggérée par Micciancio. Le principe consiste à appliquer des rotations à une matrice $\alpha\mathbf{I} + \mathbf{M}$ afin de conserver les bonnes propriétés d'orthogonalité de cette dernière tout en diminuant l'impact du choix d'une base dont les vecteurs seraient trop proches des axes principaux. Ceci permet de plus d'augmenter l'espace des clefs secrètes par rapport à GGH. Cependant, pour améliorer l'efficacité pratique, Rose et al. proposent de restreindre leur espace de clefs secrètes en se limitant à celles possédant une forme normale de Hermite optimisée, pour laquelle les coefficients diagonaux $H_{i,i}$ sont égaux à 1 pour $i \geqslant 2$. Ceci permet de disposer d'une clef publique dont seule la première colonne est constituée de coefficients non nuls, outre les coefficients diagonaux des autres colonnes. C'est par exemple trivialement le cas lorsque le déterminant de $\mathbf{R}$ est premier. Cependant, les coefficients de cette colonne peuvent être significativement plus grands que ceux d'une matrice FNH « standard », dans le sens où ils sont majorés par le déterminant. Ceci explique que le gain obtenu sur la taille moyenne de la clef publique reste relativement modeste.

Plantard, Rose, et Susilo (2010) détaillent une technique d'accélération du round-off, et plus précisément au niveau du calcul du produit $\mathbf{c}\mathbf{R}^{-1}$, en utilisant le théorème des restes chinois (TRC). Pour ce faire, ils considèrent la matrice $\mathbf{R}' = (\det \mathbf{R})\mathbf{R}^{-1}$ qui est une matrice entière. Le produit $\mathbf{c}\mathbf{R}^{-1}$ se réécrit alors $\mathbf{c}\mathbf{R}'\frac{1}{\det \mathbf{R}}$. L'approche par TRC s'applique au calcul de $\mathbf{c}\mathbf{R}'$. Il s'agit de se doter d'une base RNS $\mathcal{B} = \{m_1, \ldots, m_n\}$ vérifiant $M > 2\,\|\mathbf{c}\mathbf{R}'\|_\infty$. Le facteur 2 permet d'intégrer la possibilité de coefficients négatifs. Ainsi, l'intervalle dynamique $[\![0, M[\![$ se scinde en deux, $[\![0, \lfloor \frac{M}{2} \rfloor[\![$ représentant les entiers positifs, et $[\![\lfloor \frac{M}{2} \rfloor, M[\![$ les entiers négatifs. La méthode de calcul repose sur la formule $\mathbf{c}\mathbf{R}' \bmod M = \sum_{i=1}^{n} \left|\mathbf{c}\tilde{\mathbf{R}}^{(i)}\right|_{m_i} M_i \bmod M$, avec les matrices précalculables $\tilde{\mathbf{R}}^{(i)} = \left|M_i^{-1}\mathbf{R}'\right|_{m_i}$. Une comparaison finale avec $\lfloor \frac{M}{2} \rfloor$ permettant de récupérer le signe de chaque coefficient de $\mathbf{c}\mathbf{R}'$, il ne reste plus qu'à calculer $\lfloor \frac{\mathbf{c}\mathbf{R}'}{\det \mathbf{R}} \rceil$. Plantard et al. tirent une amélioration significative des performances, avec par exemple des facteurs de 8 à 10 sur le temps de déchiffrement par rapport au GGH originel.

Malgré les défauts du schéma originel, le principe du GGH reste donc un sujet de recherche d'actualité, menant à des propositions récentes de nouvelles versions (Yoshino et Kunihiro 2012, F. de Barros et Menasché Schechter 2014) pour lesquelles l'opération du round-off reste toujours une opération centrale dans les fonctions de déchiffrement. Dans la continuité de ces recherches

concernant l'amélioration des cryptosystèmes GGH, nous allons proposer dans les sections suivantes une contribution concernant une optimisation arithmétique de l'opération essentielle du round-off en utilisant les avantages du RNS de manière plus poussée que celle de Plantard et al. (2010).

4.2 De l'adaptation du round-off au RNS

Cette section est dédiée à la proposition d'une adaptation en RNS de la méthode de résolution du problème de plus proche vecteur dans un réseau euclidien via le round-off. Pour ce faire, les formulations du problème et des équations mathématiques associées, dont le round-off constitue le cœur, vont être reformulées pour pouvoir utiliser les RNS.

4.2.1 Reformulation adaptée pour le RNS

Le sujet de cette partie étant centré sur la fonction d'arrondi de Babai, nous allons utiliser les cryptosystèmes GGH et GGH-FNH détaillés précédemment pour donner un cadre illustratif à l'approche développée.

Nous rappelons donc que $\mathbf{R}$ désigne la clef secrète, et la clef publique $\mathbf{B}$ est dérivée de la matrice $\mathbf{R}$ par multiplication par une matrice unimodulaire notée $\mathbf{U} \in \mathrm{GL}_{\mathcal{N}}(\mathbb{Z})$, soit $\mathbf{B} = \mathbf{UR}$. Le réseau de $\mathbb{R}^{\mathcal{N}}$ défini par ces bases de rang plein est noté $\mathcal{R}$. Enfin, nous supposons que pour un paramètre entier σ donné, $\mathbf{R}$ vérifie la condition de Babai (4.4), à savoir $\rho_{\mathbf{R}} \in]0, \frac{1}{2\sigma}[$, où $\rho_{\mathbf{R}} = \left\|\mathbf{R}^{-1}\right\|_{\infty}$.

L'objectif que nous poursuivons dans ce chapitre est le suivant :

Objectif 4.1 **Proposer un algorithme RNS qui, pour un réseau $\mathcal{R}$ donné et décrit par une base R, résout exactement le problème du plus proche vecteur pour tout élément de l'ensemble $\{\mathbf{r} + \mathbf{pR}^{-1} \mid \mathbf{r} \in \mathcal{R},\ \mathbf{p} \in \left[B_{\infty}\left(0, \frac{1}{2}\right) \times \mathbf{R}\right] \cap \mathbb{Z}^{\mathcal{N}}\}$.**

À des fins d'illustration, il s'agit donc de proposer une version RNS de la fonction $\mathrm{D}_{\mathcal{R}}$ (4.3) (p. 130).

Dans notre démarche d'adaptation au RNS, nous introduisons dans un premier temps la définition du reste modulaire centré.

Définition 4.5 *La fonction de reste centré modulo m est définie par :*

$$\begin{array}{rccl} \mathrm{modc}: & \mathbb{Z} \times \mathbb{N}^* & \to & [\![-\lfloor \frac{m-1}{2} \rfloor, \lfloor \frac{m}{2} \rfloor]\!] \\ & (x, m) & \mapsto & x \bmod^{c} m = \begin{cases} |x|_m \ \textit{si}\ |x|_m \in [\![0, \lfloor \frac{m}{2} \rfloor]\!], \\ |x|_m - m \ \textit{sinon.} \end{cases} \end{array}$$

Cette fonction est correctement définie car pour tout modulus pair ou impair $m \in \mathbb{N}^*$, l'intervalle $[\![0, m-1]\!]$ est le résultat de la réunion disjointe suivante :

$$[\![0, m-1]\!] = [\![0, \lfloor \frac{m}{2} \rfloor]\!] \sqcup \left(m + [\![-\lfloor \frac{m-1}{2} \rfloor, -1]\!]\right).$$

Une première étape menant vers la réalisation de l'objectif est basée sur la remarque suivante.

Remarque 4.1 *Si m_σ est un modulus vérifiant $m_\sigma \geqslant 2\sigma + 1$, alors pour obtenir $\mathrm{D}_{\mathcal{R}}(\mathbf{c})$ il suffit de calculer son reste centré modulo m_σ.*

La remarque précédente fournit un premier moyen pour réduire le coût du calcul du plus proche vecteur en utilisant une base RNS réduite à un élément m_σ. Cependant, si la matrice $\mathbf{R} \bmod m_\sigma$ peut être précalculée et le vecteur $\mathbf{c}$ être réduit modulo m_σ, le problème central qui reste posé est celui du calcul en RNS de l'expression $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \bmod m_\sigma$.

Une fois qu'il sera possible d'obtenir $\left(\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \times \mathbf{R}\right) \bmod m_\sigma$, $\mathbf{p}$ se retrouvera alors simplement par le calcul des restes centrés modulo m_σ :

$$\mathbf{p} = \left(\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}\right) \operatorname{modc} m_\sigma. \tag{4.7}$$

En pratique, le paramètre σ est assez petit pour que la comparaison entre résidus de taille m_σ soit peu coûteuse, rendant de fait le calcul des restes centrés aisé. Dans les challenges proposés par Goldreich et al. à la suite de leur construction du GGH, σ vaut typiquement 2 ou 3.

La deuxième étape est donc de réécrire l'expression $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil$ de manière à pouvoir la calculer en RNS. Pour ce faire, il est nécessaire d'utiliser une expression ne faisant intervenir que des matrices et vecteurs entiers. En réutilisant l'approche de Plantard et al., nous introduisons la matrice entière $\mathbf{R}' = (\det \mathbf{R})\, \mathbf{R}^{-1}$, qui n'est autre que la transposée de la comatrice de $\mathbf{R}$.

Ensuite, nous remarquons que le calcul en RNS du plus proche entier inférieur à une fraction $\frac{a}{b}$, soit $\lfloor \frac{a}{b} \rfloor$, est réalisable via l'utilisation de la formule suivante :

$$\lfloor \frac{a}{b} \rfloor = \frac{a - (a \bmod b)}{b}. \tag{4.8}$$

La division du membre de droite de (4.8) étant exacte est donc facilement réalisable en RNS. Afin de calculer cette fois l'arrondi d'une fraction $\lfloor \frac{a}{b} \rceil$ en RNS, il suffit d'utiliser par exemple l'arrondi vers plus l'infini, défini par l'équation suivante :

$$\forall x \in \mathbb{R},\ \lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor. \tag{4.9}$$

Pour la suite des propos, nous utiliserons les notations suivantes :

$$\begin{cases} d = \det \mathcal{R} \\ \mathbf{d} = (d, \ldots, d) \in \mathbb{Z}^{\mathcal{N}} \end{cases} \tag{4.10}$$

En utilisant la matrice $\mathbf{R}'$ et en combinant les Équations (4.8) et (4.9), il vient alors :

$$\begin{aligned} \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil &= \lfloor \frac{\mathbf{c}\mathbf{R}'}{d} \rceil \\ &= \lfloor \frac{\mathbf{c}\mathbf{R}'}{d} + (\frac{1}{2}, \ldots, \frac{1}{2}) \rfloor \\ &= \lfloor \frac{2\mathbf{c}\mathbf{R}' + \mathbf{d}}{2d} \rfloor \\ &= \frac{1}{2d} \times \left\{ 2\mathbf{c}\mathbf{R}' + \mathbf{d} - \left[\left(2\mathbf{c}\mathbf{R}' + \mathbf{d} \right) \bmod (2d) \right] \right\}. \end{aligned} \tag{4.11}$$

La division par $2d$ étant exacte, il en résulte donc :

$$\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \bmod m_\sigma = |2d|_{m_\sigma}^{-1} \times \left\{ \left| 2\mathbf{c}\mathbf{R}' + \mathbf{d} \right|_{m_\sigma} - \left| \left(2\mathbf{c}\mathbf{R}' + \mathbf{d} \right) \bmod (2d) \right|_{m_\sigma} \right\} \bmod m_\sigma. \tag{4.12}$$

L'efficacité du calcul du membre de droite de l'Équation (4.12) va donc essentiellement se résumer au coût du calcul en RNS de $(2\mathbf{c}\mathbf{R}' + \mathbf{d}) \bmod (2d)$.

Celui-ci étant effectué via l'Algorithme 8 de réduction modulaire de Montgomery qui donne un résultat seulement dans $[\![0, 2 \times 2d[\![$, à savoir $(2\mathbf{cR}' + \mathbf{d}) \bmod (2d)$ ou $(2\mathbf{cR}' + \mathbf{d}) \bmod (2d) + 2d$, il va notamment falloir trouver un moyen efficace de calculer la réduction complète. Dans le cas contraire, le résultat obtenu pourrait être entaché d'une erreur et avoir la forme suivante :

$$\lfloor \mathbf{cR}^{-1} \rceil - \mathbf{v_e}, \ \mathbf{v_e} \in \{0,1\}^{\mathcal{N}}. \tag{4.13}$$

Exemple 4.4 *Reprenant l'Exemple 4.3 (p. 131), si nous calculons $\lfloor \mathbf{cR}^{-1} \rceil$ via la Formule (4.11) avec une réduction de Montgomery pouvant être incomplète, le résultat peut être un vecteur de la forme $\lfloor \mathbf{cR}^{-1} \rceil - \mathbf{v_e}$ avec $\mathbf{v_e} \in \{0,1\}^2$.*

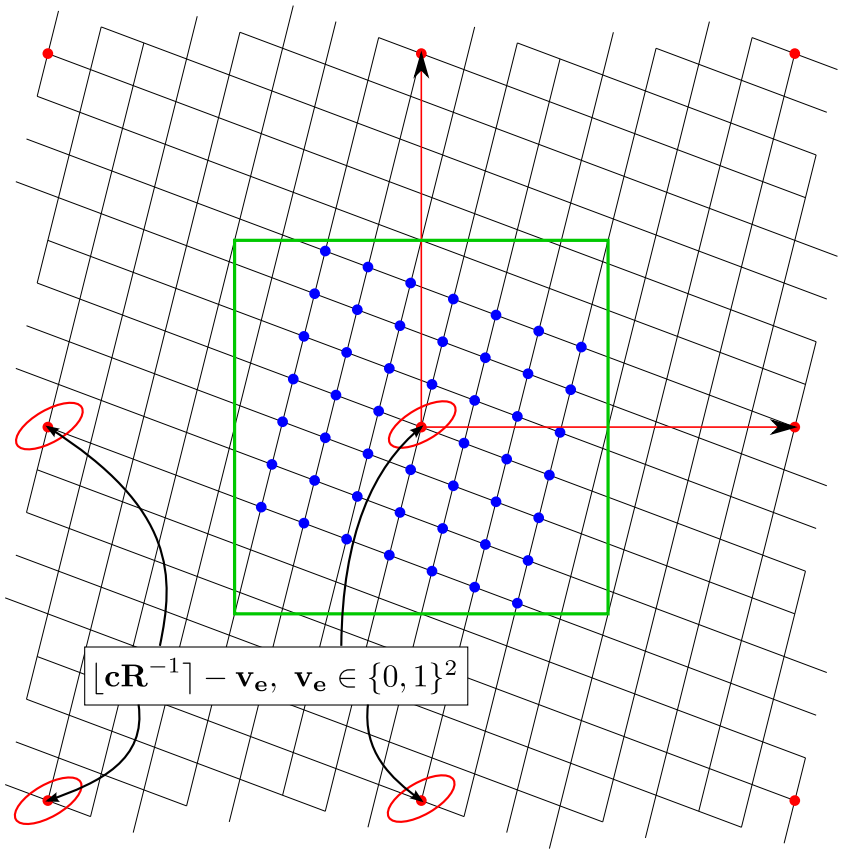


FIGURE 4.4 – *Illustration pour l'Exemple 4.3 des vecteurs possiblement retournés par la Formule (4.13) avec réduction modulaire de Montgomery incomplète.*

4.2.2 Du calcul exact de $[(2\mathbf{cR}' + \mathbf{d}) \bmod (2d)] \bmod m_\sigma$ en RNS

La démarche naturelle est d'effectuer cette réduction via un simple appel de l'Algorithme 8, en utilisant une base RNS principale $\mathcal{B}$, dont le produit des moduli est noté M, et une base auxiliaire d'un seul élément $\{m_\sigma\}$. L'exécution de la seconde conversion de base devient inutile. Ainsi, le résultat obtenu est de la forme suivante :

$$\begin{aligned} &\text{RedModRNS}\left(\mathcal{B}, \{m_\sigma\}, 2\mathbf{cR}' + \mathbf{d}, 2d\right) = \\ &\left\{ \left[\left(2\mathbf{cR}' + \mathbf{d}\right) \times \left| M^{-1} \right|_{2 \det \mathbf{R}} \right] \bmod (2d) + 2d \times \mathbf{v_e} \right\} \bmod m_\sigma \end{aligned} \tag{4.14}$$

où le vecteur $\mathbf{v_e}$ est un élément de $\{0,1\}^{\mathcal{N}}$.

Bien que ces notations soient vectorielles, les réduction s'effectuent individuellement sur chaque coefficient du vecteur $2\mathbf{cR}' + \mathbf{d}$. Ainsi, les indices i des coefficients non nuls de $\mathbf{v_e}$ sont ceux pour lesquels le résultat de la réduction de Montgomery de $(2\,(\mathbf{cR}')_i + \mathbf{d}_i)$ modulo $(2d)$ est dans l'intervalle $[\![2d, 2 \times 2d[\![$.

Calculer efficacement la réduction en RNS de $(2\mathbf{cR}' + \mathbf{d})$ modulo $(2d)$ fait apparaître trois problèmes. Le premier consiste à supprimer le facteur de Montgomery après la réduction ou calculer les représentations de Montgomery avant la réduction. Le second concerne le fait que, vu la définition (4.2) de la fonction de chiffrement, les coefficients de $\mathbf{c}$, et donc de $2\mathbf{cR}' + \mathbf{d}$, peuvent être négatifs. Enfin, le troisième est de réduire complètement le résultat dans $[\![0, 2d[\![$, en recouvrant le plus efficacement possible le vecteur $\mathbf{v_e}$ afin de le corriger.

Gérer le facteur de Montgomery

Soit $\mathcal{B} = \{m_1, \ldots, m_n\}$ la base RNS principale utilisée pour le calcul de la réduction et M le produit de ses élément. La matrice $\mathbf{R}'$ et le vecteur $\mathbf{d}$ étant connus, une solution est de précalculer et mémoriser les valeurs suivantes :

$$\begin{cases} \tilde{\mathbf{R}} = 2M \times \mathbf{R}' \bmod (2d)\,, \\ \tilde{\mathbf{d}} = M \times \mathbf{d} \bmod (2d)\,. \end{cases} \tag{4.15}$$

De cette manière, il s'agit désormais de calculer la réduction suivante :

$$\begin{aligned} &\mathrm{RedModRNS}\left(\mathcal{B}, \{m_\sigma\}, \mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}, 2d\right) \\ &= \left\{\left[\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right) \times \left|M^{-1}\right|_{2d}\right] \bmod (2d) + 2d \times \mathbf{v_e}\right\} \bmod m_\sigma \\ &= \left\{\left(2\mathbf{cR}' + \mathbf{d}\right) \bmod (2d) + 2d \times \mathbf{v_e}\right\} \bmod m_\sigma. \end{aligned} \tag{4.16}$$

De la possible négativité des coefficients du vecteur $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$

Afin de pouvoir utiliser les RNS pour représenter l'ensemble des valeurs possibles de $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$, il est nécessaire de poser l'hypothèse suivante.

Hypothèse 4.1 *Soit $\mathcal{C}$ l'ensemble de tous les vecteurs $\mathbf{c}$ pouvant être traités par la procédure du round-off dans un contexte donné. Alors $\mathcal{C}$ est supposé borné.*

Cette hypothèse permet de construire une base RNS principale $\mathcal{B}$ avec laquelle le calcul de la réduction modulaire $\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right) \bmod (2d)$ est possible quel que soit le vecteur $\mathbf{c} \in \mathcal{C}$ considéré. Nous définissons l'entier c_∞ par :

$$c_\infty = \max\{\|\mathbf{c}\|_\infty \mid \mathbf{c} \in \mathcal{C}\}. \tag{4.17}$$

Autrement dit, comme les coefficients de $\tilde{\mathbf{R}}$ et $\tilde{\mathbf{d}}$ sont positifs, la boule pour la norme infinie $[-\mathcal{N}c_\infty \times 2d, (\mathcal{N}c_\infty + 1) \times 2d]^{\mathcal{N}}$ contient l'ensemble $\{\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}} \mid \mathbf{c} \in \mathcal{C}\}$.

Par exemple, dans le contexte du cryptosystème décrit par les fonctions (4.2) et (4.3) (p. 130), $\mathcal{C}$ et c_∞ dépendent notamment de la manière dont est calculé le vecteur aléatoire $\mathbf{k}$ via la distribution f. L'Hypothèse 4.1 exprime seulement le fait que l'ensemble des vecteurs $\mathbf{k}$ possibles est borné. Goldreich et al. (1997) suggèrent par exemple de choisir $\mathbf{k}$ dans $[\![-\mathcal{N}, \mathcal{N}]\!]^{\mathcal{N}}$. Dans ce cas, $c_\infty = \mathcal{N} \times \|\mathbf{B}\|_\infty + \sigma$.

Dans l'amélioration apportée par (Micciancio 2001), la clef publique $\mathbf{B}$ est la forme normale de Hermite de la clef privée $\mathbf{R}$, et le vecteur chiffré $\mathbf{c}$ est le résultat de la réduction de $\mathbf{p}$ modulo $\mathbf{B}$. Ceci a notamment pour conséquence que pour tout indice $i \in [\![0, \mathcal{N}]\!]$, $\mathbf{c}_i \in [\![0, \mathbf{B}_{i,i}[\![$. Par conséquent, $c_\infty = \max_{i \in [\![1,\mathcal{N}]\!]} (\mathbf{B}_{i,i})$. De plus, les vecteurs $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ n'ont dans ce cas précis aucun coefficient négatif.

Plus généralement, connaissant c_∞, il suffit de trouver un vecteur $\mathbf{b}$ du réseau $\mathcal{R}$ dont chaque coefficient est au moins égal à c_∞. De ce fait, pour tout $\mathbf{c} \in \mathcal{C}$, le vecteur $\mathbf{c} + \mathbf{b}$ est positif. Le calcul du plus proche vecteur de $\mathbf{c}$ peut alors être substitué par celui du plus proche vecteur de $\mathbf{c} + \mathbf{b}$. En effet, vu que $\mathbf{b} \in \mathcal{R}$, $\mathbf{b}\mathbf{R}^{-1}$ est un vecteur entier et $\mathbf{p}$ est ainsi retrouvé par :

$$\begin{aligned}(\mathbf{c}+\mathbf{b}) - \lfloor (\mathbf{c}+\mathbf{b})\,\mathbf{R}^{-1} \rceil \mathbf{R} &= (\mathbf{c}+\mathbf{b}) - \lfloor \mathbf{p}\mathbf{R}^{-1} + \mathbf{k}\mathbf{U} + \mathbf{b}\mathbf{R}^{-1} \rceil \mathbf{R} \\ &= (\mathbf{p}+\mathbf{k}\mathbf{B}+\mathbf{b}) - \lfloor \mathbf{p}\mathbf{R}^{-1} \rceil \mathbf{R} - \left(\mathbf{k}\mathbf{U} + \mathbf{b}\mathbf{R}^{-1}\right)\mathbf{R} \\ &= \mathbf{p} + \mathbf{k}\mathbf{B} + \mathbf{b} - \mathbf{k}\mathbf{B} - \mathbf{b} \\ &= \mathbf{p}.\end{aligned}$$

Un tel vecteur $\mathbf{b}$ peut par exemple être choisi comme étant un multiple du vecteur $\mathbf{d}$, ou encore être calculé en utilisant la forme normale de Hermite $\mathbf{H}$ de $\mathbf{R}$, puisque tous les coefficients de $\mathbf{H}$ ont l'avantage d'être positifs. Par conséquent, il est toujours possible de supposer que les vecteurs $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ sont positifs.

Plus généralement, une réduction de Montgomery RNS modulo z appliquée à un entier x pouvant être négatif peut donner un résultat dans un intervalle de taille $2z$ dont l'intersection avec $]\!] - \infty, 0[\!]$ peut être non vide. Cela dépend de la taille choisie pour la base RNS principale $\mathcal{B}$.

Par exemple, si les entrées x possibles vérifient la condition $|x| < x_\infty$, et si la base principale $\mathcal{B}$ vérifie $M > \frac{2x_\infty}{p}$, alors le résultat s de la réduction, exprimé dans la base auxiliaire $\mathcal{B}'$, satisfait aux inégalités suivantes :

$$-\frac{z}{2} < s = \frac{x + qz}{M} < \frac{3}{2}z.$$

(pour l'illustration précédente, la conversion de $q = \left|-xz^{-1}\right|_M$ est supposée complètement réduite modulo M). Dans ce cas, les coefficients du vecteur $\mathbf{v_e}$ apparaissant dans l'Équation (4.14) prennent leur valeur dans $\{-1, 0, 1\}$.

Par la suite, nous considérerons, sauf mention contraire, que les vecteurs $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ peuvent contenir des coefficients positifs et négatifs.

Réduction exacte avec conversion de base unique pour d impair

Lorsque M vérifie la condition suivante :

$$M > \left\|\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right\|_\infty \tag{4.18}$$

alors, pour tout indice $i \in [\![1, \mathcal{N}]\!]$, l'exécution de la réduction modulaire RNS $\text{RedModRNS}\left(\mathcal{B}, \mathcal{B}', \left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i, 2d\right)$ avec $\text{Bex}_1 = \text{Bex}_{mrs}$ permet d'obtenir les résidus dans la base auxiliaire $\mathcal{B}'$ de la quantité s vérifiant les inégalités suivantes :

$$-1 < s = \frac{\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i + \left|-\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i \times (2d)^{-1}\right|_M \times 2d}{M} < 2d + 1.$$

Ainsi, le résultat obtenu est dans l'intervalle $[\![0, 2d]\!]$. Il ne reste donc plus qu'à vérifier que s n'est pas égal à $2d$. Si $2d$ s'inscrit dans l'intervalle dynamique de la base $\mathcal{B}'$, autrement dit si $M' > 2d$, alors une comparaison de ses résidus RNS avec ceux de s dans $\mathcal{B}'$ permet de conclure. Cependant il se trouve que l'hypothèse de départ donnant d impair garantit que s ne peut être égal à $2d$. En effet, comme M est nécessairement premier avec $2d$ et est donc impair, il suffit de voir que :

$$\begin{aligned} |s|_2 &= \left|\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i\right|_2 \\ &= \left|\left(\mathbf{c} \times \left|2M \times \mathbf{R}'\right|_{2d}\right)_i + \left(|M \times \mathbf{d}|_{2d}\right)_i\right|_2 \\ &= \left|2M \times \left(\mathbf{c}\mathbf{R}'\right)_i + M \times d\right|_2 \\ &= |M \times d|_2 \\ &= 1. \end{aligned}$$

Par suite, s est impair et la réduction modulaire donne un résultat exact, c'est-à-dire que le vecteur $\mathbf{v_e}$ de l'Équation (4.14) est dans ce cas toujours nul. Il est donc possible d'utiliser comme base auxiliaire $\mathcal{B}' = \{m_\sigma\}$. L'Algorithme 20 met en œuvre cette approche.

Algorithme 20 : ReducExacte_v1 $\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}, 2d, \mathcal{B} \cup \{m_\sigma\}\right)$ *(cas d impair)*

Données : Le vecteur $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ exprimé dans la base RNS $\mathcal{B} \cup \{m_\sigma\}$, le modulus m_σ, et une base RNS $\mathcal{B}$ première à $2m_\sigma d$ telle que $M > 2d\left(\mathcal{N}c_\infty + 1\right) \geqslant \left\|\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right\|_\infty$ pour tout $\mathbf{c} \in \mathcal{C}$.
Résultat : $[(2\mathbf{c}\mathbf{R}' + \mathbf{d}) \bmod (2d)] \bmod m_\sigma$.

1 **début**
2 $\quad \mathbf{s} \leftarrow (0, \ldots, 0) \in \mathbb{Z}^{\mathcal{N}}$
3 $\quad$ **pour** $i \leftarrow 1$ **à** $\mathcal{N}$ **faire**
$\quad\quad$ `/* en // sur les coeff. de` $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ `*/`
4 $\quad\quad q_\mathcal{B} \leftarrow \left|-\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i \times (2d)^{-1}\right|_M$
$\quad\quad$ `/* en // dans` $\mathcal{B}$ `*/`
5 $\quad\quad q_\sigma \leftarrow \mathrm{Bex}_{mrs}\left(\mathcal{B}, \{m_\sigma\}, q_\mathcal{B}\right)$
6 $\quad\quad \mathbf{s}_i \leftarrow \left|\left(\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i + q_\sigma \times 2d\right) \times M^{-1}\right|_{m_\sigma}$
7 $\quad$ **retourner s**

Coût de l'Algorithme 20 Soit β la taille donnée des moduli de $\mathcal{B}$. Le nombre n d'éléments de $\mathcal{B}$ est alors $n = \lceil \log_\beta\left(\mathcal{N}c_\infty + 1\right) + \log_\beta(2d) \rceil$. Nous rappelons de plus que le modulus m_σ vérifie $m_\sigma \geqslant 2\sigma + 1$.

Pour chaque indice $i \in [\![1, \mathcal{N}]\!]$, le calcul des résidus $q_\mathcal{B}$ requiert n $\texttt{MMEl}_\mathcal{B}$. Le coût de la conversion Bex_{mrs} nécessite $\frac{n(n-1)}{2}$ $\texttt{MMEl}_\mathcal{B}$ et $\frac{n(n-1)}{2}$ $\texttt{AMEl}_\mathcal{B}$ pour le calcul des coefficients MRS, et $(n-1)$ $\texttt{MMEl}_{m_\sigma}$ et $(n-1)$ $\texttt{AMEl}_{m_\sigma}$ pour la reconstruction dans la base $\{m_\sigma\}$ (cf. Éq. (1.12)). Enfin, le calcul de $\mathbf{s}_i$ est finalisé par 2 $\texttt{MMEl}_{m_\sigma}$ et 1 $\texttt{AMEl}_{m_\sigma}$.

En conclusion, le coût total de l'Algorithme 20 est le suivant :

$$\begin{cases} \mathcal{N} \times \left(\dfrac{n(n+1)}{2} \ \texttt{MMEl}_{\mathcal{B}} + \dfrac{n(n-1)}{2} \ \texttt{AMEl}_{\mathcal{B}} + (n+1) \ \texttt{MMEl}_{m_\sigma} + n \ \texttt{AMEl}_{m_\sigma} \right), \\ n = \lceil \log_\beta (\mathcal{N} c_\infty + 1) + \log_\beta (2d) \rceil. \end{cases} \tag{4.19}$$

Réduction exacte avec conversion accélérée

La réduction précédente présente l'avantage de ne nécessiter qu'une seule conversion de base vers une base à un seul élément. Cependant, aucune flexibilité n'est possible car cette conversion doit être de type MRS (puisqu'il faut une conversion complètement réduite). De plus, celle-ci est coûteuse car le calcul des coefficients MRS fait intervenir beaucoup de dépendances entre les résultats intermédiaires, brisant de ce fait le caractère parallèle apporté par le RNS. Enfin, la base principale $\mathcal{B}$ doit être choisie assez grande (4.18) pour y représenter chaque coefficient de tous les vecteurs $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ possibles, ce qui n'avantage pas non plus cette approche (cf. Remarque 1.10 p. 30).

La version présentée maintenant s'appuie sur une conversion efficace, à savoir Bex_{crt}. Le problème qui va se présenter est que le résultat de la réduction peut ne plus être automatiquement obtenu exactement. Il sera donc nécessaire de retrouver le vecteur $\mathbf{v_e}$ via une comparaison entre le résultat de la réduction et $2d$.

La méthode proposée est décrite par l'Algorithme 21. Celle-ci se fonde sur deux réductions de Montgomery successives, avec une base $\mathcal{B}$ à n moduli dont le produit est noté M, puis avec une seconde base réduite à un unique modulus $\tilde{m}$. Les représentations de Montgomery doivent être modifiées en conséquence :

$$\begin{cases} \tilde{\mathbf{R}} = 2\tilde{m}M \times \mathbf{R}' \bmod (2d), \\ \tilde{\mathbf{d}} = \tilde{m}M \times \mathbf{d} \bmod (2d). \end{cases} \tag{4.20}$$

Preuve de correction de l'Algorithme 21 La première étape dans la réduction du i-ième coefficient de $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ consiste en une réduction de Montgomery RNS s'appuyant sur les bases $\mathcal{B}$ et $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$.

L'entier $q = \left| -\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i \times (2d)^{-1} \right|_M$ calculé dans les canaux de $\mathcal{B}$ est converti via Bex_{crt} dans les canaux de $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$. Or, la conversion n'étant pas nécessairement complètement réduite, la valeur $\widehat{q} = \text{Bex}_{crt}(\mathcal{B}, \boldsymbol{q}_{\mathcal{B}})$ est égale à $q + \delta M$ avec $\delta \in [\![0, n-1]\!]$ (cf. Éq. (1.27), p.29). La quantité $r = \frac{\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i + \widehat{q} \times 2d}{M}$, calculée dans la base auxiliaire $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$, vérifie :

$$\begin{aligned} r &\equiv \left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i \times \left| M^{-1} \right|_{2d} \bmod (2d) \\ &\equiv \left(2\tilde{m}M \times \mathbf{c}\mathbf{R}' + \tilde{m}M \times \mathbf{d}\right)_i \times \left| M^{-1} \right|_{2d} \bmod (2d) \\ &\equiv \left(2\tilde{m} \times \mathbf{c}\mathbf{R}' + \tilde{m} \times \mathbf{d}\right)_i \bmod (2d). \end{aligned} \tag{4.21}$$

De plus, r vérifie les inégalités suivantes :

$$-2d = \frac{-M \times 2d}{M} < r < \frac{M \times 2d + nM \times 2d}{M} = (n+1) \times 2d.$$

Algorithme 21 : ReducExacte_v2 $\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}, 2d, \mathcal{B} \cup \mathcal{B}' \cup \{\tilde{m}, m_\sigma\}\right)$

Données :

- le vecteur $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ exprimé dans la base RNS $\mathcal{B} \cup \mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$
- deux bases RNS copremières $\mathcal{B}$ et $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$ ($n = |\mathcal{B}|$, $\ell = |\mathcal{B}'|$), vérifiant M et $\tilde{m}$ premiers avec $2m_\sigma M'd$, M premier avec $\tilde{m}$, $M > \mathcal{N}c_\infty + 1$, $M' > 2 \times 2d$, $\tilde{m} \geqslant n + 1$ et $\tilde{m} < m'$ pour tout $m' \in \mathcal{B}'$,
- les coefficients dans la base MRS associée à $\mathcal{B}'$ de $2d$, notés $(2d)_{mrs}$, ainsi que ceux d'un nombre $t \in [\![(1 + \frac{n}{\tilde{m}}) \times 2d, M' - \frac{1}{\tilde{m}} \times 2d]\!]$, notés t_{mrs}

Résultat : $[(2\mathbf{c}\mathbf{R}' + \mathbf{d}) \bmod (2d)] \bmod m_\sigma$.

début
- $\mathbf{s} \leftarrow (0, \ldots, 0) \in \mathbb{Z}^{\mathcal{N}}$
- **pour** $i \leftarrow 1$ **à** $\mathcal{N}$ **faire**
  - `/* en // sur les coeff. de` $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ `*/`
  - $q_{\mathcal{B}} \leftarrow \left| -\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i \times (2d)^{-1} \right|_M$
  - `/* en // dans` $\mathcal{B}$ `*/`
  - $(\widehat{q}_{\mathcal{B}'}, \widehat{q}_{\tilde{m}}, \widehat{q}_{m_\sigma}) \leftarrow \mathrm{Bex}_{crt}\left(\mathcal{B}, \mathcal{B}' \cup \{\tilde{m}, m_\sigma\}, q_{\mathcal{B}}\right)$
  - **pour** $m \in \mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$ **faire**
    - `/* en // dans` $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$ `*/`
    - $r_m \leftarrow \left|\left(\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i + \widehat{q}_m \times 2d\right) M^{-1}\right|_m$
    - `/* 1ère réduc. de Montgomery */`
  - $\tilde{q} \leftarrow \left| -r_{\tilde{m}} \times (2d)^{-1} \right|_{\tilde{m}}$
  - **pour** $m \in \mathcal{B}' \cup \{m_\sigma\}$ **faire**
    - `/* en // dans` $\mathcal{B}' \cup \{m_\sigma\}$ `*/`
    - $s_m \leftarrow \left|(r_m + \tilde{q} \times 2d)\, \tilde{m}^{-1}\right|_m$
    - `/* 2nde réduc. de Montgomery */`
  - $s_{mrs} \leftarrow \mathrm{MRScoeff}\left(\mathcal{B}', s_{\mathcal{B}'}\right)$
  - **si** $s_{mrs} \geqslant t_{mrs}$ **alors**
    - `/* comparaison en MRS */`
    - $s_{m_\sigma} \leftarrow |s_{m_\sigma} + 2d|_{m_\sigma}$
  - **sinon si** $s_{mrs} \geqslant (2d)_{mrs}$ **alors**
    - `/* comparaison en MRS */`
    - $s_{m_\sigma} \leftarrow |s_{m_\sigma} - 2d|_{m_\sigma}$
  - $\mathbf{s}_i \leftarrow s_{m_\sigma}$
- **retourner s**

Par conséquent,

$$r = \left|\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i\right|_{2d} + \alpha \times 2d,\ \alpha \in [\![-1, n]\!].$$

Le modulus supplémentaire $\tilde{m}$ permet d'effectuer une seconde réduction de Montgomery de r modulo $2d$. Lorsque ce modulus est choisi plus petit que les moduli de $\mathcal{B}'$, la conversion du coefficient $\tilde{q} = \left|-r \times (2d)^{-1}\right|_{\tilde{m}}$ vers la base $\mathcal{B}'$ (resp. $\{m_\sigma\}$) est une simple duplication (resp. une simple réduction modulo m_σ). Au final, la quantité $s = \frac{r+\tilde{q}\times 2d}{\tilde{m}}$ calculée dans la base $\mathcal{B}' \cup \{m_\sigma\}$ vérifie :

$$\begin{aligned} s &\equiv r \times \left|\tilde{m}^{-1}\right|_{2d} \bmod (2d) \\ &\equiv \left(2\tilde{m} \times \mathbf{c}\mathbf{R}' + \tilde{m} \times \mathbf{d}\right)_i \times \left|\tilde{m}^{-1}\right|_{2d} \bmod (2d) \\ &\equiv \left(2 \times \mathbf{c}\mathbf{R}' + \times \mathbf{d}\right)_i \bmod (2d). \end{aligned} \tag{4.22}$$

s satisfait de plus aux inégalités suivantes :

$$-\frac{1}{\tilde{m}} \times 2d < s < \frac{n+1+\tilde{m}-1}{\tilde{m}} \times 2d = \left(1 + \frac{n}{\tilde{m}}\right) \times 2d.$$

La condition $\tilde{m} \geqslant n+1$ garantit alors que s appartient à un intervalle de diamètre inférieur à $4d$. Vu la condition sur la taille $\mathcal{B}'$, à savoir $M' > 4d$, il vient en conséquence qu'en calculant les coefficients MRS de s depuis ses résidus dans $\mathcal{B}'$, il est possible de réduire complètement s dans l'intervalle $[\![0, 2d[\![$.

La Figure 4.5 illustre cette dernière étape. L'entier t utilisé sert à déterminer le signe de s. Il peut être choisi librement dans l'intervalle $[\![\left(1 + \frac{n}{\tilde{m}}\right) \times 2d, M' - \frac{1}{\tilde{m}} \times 2d]\!]$. Par conséquent, s vérifie $-M' + t < s < t$ et ainsi :

$$\begin{cases} s \geqslant 0 \text{ si } |s|_{M'} < t \\ s < 0 \text{ si } |s|_{M'} > t \end{cases} \tag{4.23}$$

Puis, une ultime comparaison de $|s|_{M'}$ avec $2d$ permet d'obtenir le résultat attendu $|(2\mathbf{c}\mathbf{R}' + \mathbf{d})_i|_{2d}$. Pour ce faire, il suffit de voir que :

$$\begin{cases} s = \left|\left(2\mathbf{c}\mathbf{R}' + \mathbf{d}\right)_i\right|_{2d} - 2d \text{ si } s < 0 \\ s = \left|\left(2\mathbf{c}\mathbf{R}' + \mathbf{d}\right)_i\right|_{2d} \text{ si } 0 \leqslant s < 2d \\ s = \left|\left(2\mathbf{c}\mathbf{R}' + \mathbf{d}\right)_i\right|_{2d} + 2d \text{ sinon (\textit{i.e.} si } 2d \leqslant s < t) \end{cases} \tag{4.24}$$

Nous déduisons que la réduction finale s'effectue alors comme suit :

$$\begin{cases} s \leftarrow s + 2d \text{ si } t < |s|_{M'} \\ s \leftarrow s - 2d \text{ si } 2d \leqslant |s|_{M'} < t \end{cases} \tag{4.25}$$

Coût de l'Algorithme 21 Les tailles n et ℓ des bases $\mathcal{B}$ et $\mathcal{B}'$ sont respectivement $\lceil \log_\beta (\mathcal{N} c_\infty + 1) \rceil$ et $\lceil \log_\beta (4d) \rceil$.

Le coût est calculé pour un coefficient d'indice $i \in [\![1, \mathcal{N}]\!]$. Le calcul des résidus $\boldsymbol{q}_\mathcal{B}$ et des coefficients $\xi_{j,q,\mathcal{B}}$ utilisés lors de la première conversion est réalisable en n `MMEl` puisque les résidus $\left|-(2d)^{-1} \times M_j^{-1}\right|_{m_j}$, pour $m_j \in \mathcal{B}$, sont précalculables. Reprenant le coût (1.28), la conversion de q vers $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$

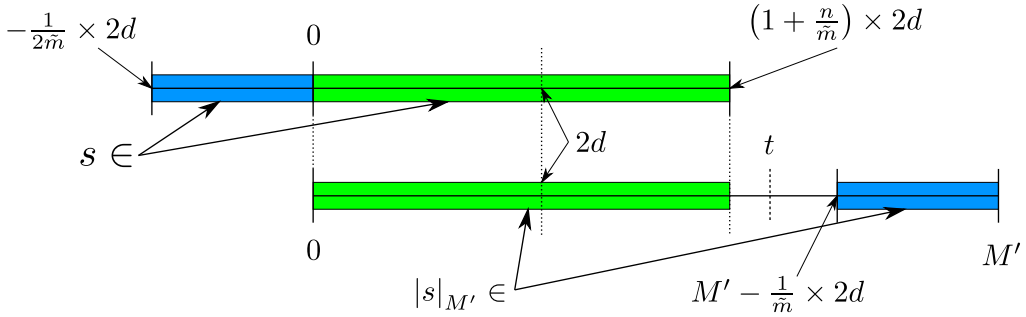


FIGURE 4.5 – *Intervalles contenant s et $|s|_{M'}$ dans l'Algorithme 21 avant la comparaison finale.*

s'effectue alors en $n\ell$ $\texttt{MMEl} + (n-1)\ell$ $\texttt{AMEl} + n$ $\texttt{MMEl}_{\tilde{m}} + (n-1)$ $\texttt{AMEl}_{\tilde{m}} + n$ $\texttt{MMEl}_{m_\sigma} + (n-1)$ $\texttt{AMEl}_{m_\sigma}$.

Ensuite, le calcul des résidus de r dans $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$ requiert 2ℓ $\texttt{MMEl} + \ell$ $\texttt{AMEl} + 2$ $\texttt{MMEl}_{\tilde{m}} + 1$ $\texttt{AMEl}_{\tilde{m}} + 2$ $\texttt{MMEl}_{m_\sigma} + 1$ $\texttt{AMEl}_{m_\sigma}$. Le calcul de $\tilde{q}$ s'effectue lui simplement en 1 $\texttt{MMEl}_{\tilde{m}}$.

Viennent alors les calculs des résidus de s dans $\mathcal{B}' \cup \{m_\sigma\}$ via 2ℓ $\texttt{MMEl} + \ell$ $\texttt{AMEl} + 2$ $\texttt{MMEl}_{m_\sigma} + 1$ $\texttt{AMEl}_{m_\sigma}$.

Enfin, la comparaison finale nécessite de calculer les coefficients MRS de $|s|_{M'}$, ce qui nécessite $\frac{\ell(\ell-1)}{2}$ $\texttt{MMEl} + \frac{\ell(\ell-1)}{2}$ $\texttt{AMEl}$. Puis l'éventuelle correction finale coûte au plus 2 $\texttt{AMEl}_{m_\sigma}$.

Finalement, le coût total de l'Algorithme 21 est le suivant :

$$\begin{cases} \mathcal{N} \times \Big((n\ell + \frac{\ell^2}{2} + \frac{7\ell}{2} + n)\ \texttt{MMEl} + (n\ell + \frac{\ell^2}{2} + \frac{3\ell}{2})\ \texttt{AMEl} \\ \quad + (n+3)\ \texttt{MMEl}_{\tilde{m}} + n\ \texttt{AMEl}_{\tilde{m}} + (n+4)\ \texttt{MMEl}_{m_\sigma} + (n+3)\ \texttt{AMEl}_{m_\sigma} \Big), \\ n = \lceil \log_\beta (\mathcal{N} c_\infty + 1) \rceil, \\ \ell = \lceil \log_\beta (4d) \rceil, \\ \tilde{m} \geqslant n+1. \end{cases} \tag{4.26}$$

Dans le cas d'entrées à coefficients uniquement positifs, l'approche précédente est simplifiée à l'étape de la comparaison finale. En effet, dans ce cas, le résultat avant comparaison se situe dans l'intervalle $[\![0, 4d[\![$. Une seule comparaison de $\boldsymbol{s}_{mrs}$ avec $(2d)_{mrs}$ suffit alors.

Exemple 4.5 *Soit le réseau $\mathcal{R}$ de $\mathbb{Z}^4$ muni de la base*

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & -10 & 1 \\ -10 & 0 & 1 & -1 \\ 1 & -20 & 0 & 17 \\ 5 & -79 & -5 & -76 \end{pmatrix}.$$

Ainsi, $d = 287489 = 19 \times 15131$ et $\rho_{\mathbf{R}} < \frac{1}{8}$, de manière à ce que $\sigma = 4$, et m_σ est fixé à 9. Pour fixer la taille des bases RNS, les vecteurs de $\mathcal{C}$ sont réduits modulo d, fixant ainsi $c_\infty = d$. Soit par exemple le vecteur $\mathbf{c} = \begin{pmatrix} 287321 & 144 & 287385 & 287078 \end{pmatrix}$, somme de $\mathbf{p} = \begin{pmatrix} -3 & 2 & 0 & 4 \end{pmatrix}$ et d'un vecteur $\mathbf{z} = \begin{pmatrix} 287324 & 142 & 287385 & 287077 \end{pmatrix}$

du réseau. La taille des moduli est fixée à 7 bits, soit $\beta = 2^7 = 128$. *L'exécution des Algorithmes 20 et 21 doit retourner* $2\mathbf{c}\mathbf{R}' + \mathbf{d} = \begin{pmatrix}7 & 1 & 4 & 8\end{pmatrix}$.

1. *Algorithme 20 :* $\mathcal{B}$ *est choisie comme étant* $\{101, 103, 107, 109, 113, 127\}$, *de manière à ce que* $M = 1.741.209.542.339 > 2d(\mathcal{N}c_\infty + 1) = 8d^2 + 2d = 661.199.975.946$. *Ainsi, les représentations de Montgomery de* $\mathbf{R}$ *et* $\mathbf{d}$ *sont les suivantes :*

$$\tilde{\mathbf{R}} = \left|2M\mathbf{R}'\right|_{2d} = \begin{pmatrix}215480 & 22494 & 57334 & 378508\\ 403370 & 469072 & 205656 & 552026\\ 80010 & 166666 & 182 & 218300\\ 225122 & 516704 & 1820 & 458066\end{pmatrix},$$

$$\tilde{\mathbf{d}} = \left|M\mathbf{d}\right|_{2d} = \begin{pmatrix}287489 & 287489 & 287489 & 287489\end{pmatrix}.$$

En utilisant les résidus de $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ *donnés dans* $\mathcal{B}$, *le calcul des résidus du vecteur* $\mathbf{q}$ *donne :*

mod	$\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$	$\mathbf{q}$
101	$(47, 95, 88, 32)$	$(57, 40, 53, 56)$
103	$(99, 75, 75, 89)$	$(13, 91, 91, 97)$
107	$(70, 50, 23, 4)$	$(82, 28, 30, 75)$
109	$(11, 62, 29, 87)$	$(69, 52, 63, 80)$
113	$(33, 102, 107, 104)$	$(9, 110, 60, 90)$
127	$(17, 99, 100, 37)$	$(80, 55, 112, 77)$

Des résidus précédents, les coefficients MRS dans la base $\{1, 101, \ldots, 101 \times \ldots \times 113\}$ *de chaque coefficient* $\mathbf{q}_i$ *sont calculés. Par exemple pour* $\mathbf{q}_1$ *nous obtenons :*

$$\begin{cases}
\tilde{q}_{101} = 57\\
\tilde{q}_{103} = \left|(13 - 57\dfrac{1}{101}\right|_{103} = 22\\
\tilde{q}_{107} = \left|((82 - 57)\dfrac{1}{101} - 22)\dfrac{1}{103}\right|_{107} = 11\\
\tilde{q}_{109} = \left|(((69 - 57)\dfrac{1}{101} - 22)\dfrac{1}{103} - 11)\dfrac{1}{107}\right|_{109} = 108\\
\tilde{q}_{113} = \left|((((9 - 57)\dfrac{1}{101} - 22)\dfrac{1}{103} - 11)\dfrac{1}{107} - 108)\dfrac{1}{109}\right|_{113} = 85\\
\tilde{q}_{127} = \left|(((((80 - 57)\dfrac{1}{101} - 22)\dfrac{1}{103} - 11)\dfrac{1}{107} - 108)\dfrac{1}{109} - 85)\dfrac{1}{113}\right|_{127} = 69
\end{cases}$$

Par suite, le résidu de $\mathbf{q}_1$ *modulo* $m_\sigma = 9$ *est donné par*

$$\mathbf{q}_1 \bmod 9 = \left|57 + 22 \times 101 + \ldots + 69 \times 101 \times 103 \times 107 \times 109 \times 113\right|_9 = 7.$$

En procédant de même pour les autres coefficients, le résultat final de la conversion est alors $\mathbf{q} \bmod 9 = \begin{pmatrix}7 & 8 & 6 & 2\end{pmatrix}$. *Enfin, comme* $(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}) \bmod 9 = \begin{pmatrix}7 & 0 & 5 & 5\end{pmatrix}$, *il vient finalement :*

$$\frac{\begin{pmatrix}7 & 0 & 5 & 5\end{pmatrix} + \begin{pmatrix}7 & 8 & 6 & 2\end{pmatrix} \times 2d}{M} \bmod 9 = \begin{pmatrix}7 & 1 & 4 & 8\end{pmatrix}$$

soit le résultat attendu.

2. *Algorithme 21 : $\mathcal{B}$ et $\mathcal{B}'$ sont respectivement* $\{101, 113, 127\}$ *et* $\{103, 107, 109\}$. *Ainsi,* $M > \mathcal{N}c_\infty + 1 = 8d + 1$ *et* $M' > 4d$. *Par suite,* $\tilde{m} = 5 > n + 1$, *et* $m_\sigma = 9$. *Les représentations de Montgomery de* $\mathbf{R}$ *et* $\mathbf{d}$ *sont :*

$$\tilde{\mathbf{R}} = \left|2\tilde{m}M\mathbf{R}'\right|_{2d} = \begin{pmatrix} 173602 & 544200 & 309062 & 336614 \\ 474164 & 445664 & 127946 & 520752 \\ 254314 & 353630 & 551008 & 137076 \\ 243228 & 86432 & 335278 & 220804 \end{pmatrix},$$

$$\tilde{\mathbf{d}} = \left|\tilde{m}M\mathbf{d}\right|_{2d} = \begin{pmatrix} 287489 & 287489 & 287489 & 287489 \end{pmatrix}.$$

Étant donné les résidus de $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ *dans* $\mathcal{B}$, *le calcul de* $\mathbf{q}$ *dans* $\mathcal{B}$ *puis du résultat* $\hat{\mathbf{q}}$ *de sa conversion vers* $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$ *sont détaillés ci-après :*

mod	$\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$	$\mathbf{q}$	$\hat{\mathbf{q}} = \mathrm{Bex}_{crt}(\mathbf{q})$
101	$(41, 94, 79, 72)$	$(97, 13, 12, 25)$	$\times$
113	$(102, 93, 46, 43)$	$(110, 87, 105, 22)$	$\times$
127	$(46, 16, 39, 49)$	$(82, 23, 64, 126)$	$\times$
103	$(46, 57, 3, 12)$	$\times$	$(79, 40, 6, 79)$
107	$(75, 25, 72, 60)$	$\times$	$(78, 60, 97, 77)$
109	$(27, 90, 49, 108)$	$\times$	$(16, 71, 22, 103)$
5	$(0, 2, 3, 0)$	$\times$	$(4, 4, 3, 4)$
9	$(1, 3, 8, 2)$	$\times$	$(3, 3, 2, 4)$

Suivent alors les calculs des vecteurs $\mathbf{r}_m = \frac{\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}} + \hat{\mathbf{q}}p}{M} \bmod m$ *pour tout* $m \in \mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$, *puis du vecteur* $\tilde{\mathbf{q}} = \left|-\mathbf{r}_{\tilde{m}} p^{-1}\right|_{\tilde{m}}$ *nécessaire pour la seconde conversion, qui est une simple duplication dans les canaux de* $\mathcal{B}' \cup \{m_\sigma\}$, *et enfin les vecteurs* $\mathbf{s}_m = \frac{\mathbf{r}_m + \tilde{\mathbf{q}}p}{\tilde{m}} \bmod m$. *Le détail des résultats de ces calculs est résumé dans le tableau suivant :*

mod	$\mathbf{r}_m$	$\tilde{\mathbf{q}}$	$\mathrm{Bex}(\tilde{\mathbf{q}})$	$\mathbf{s}_m$
103	$(50, 100, 35, 102)$	$\times$	$(1, 2, 1, 1)$	$(37, 74, 34, 68)$
107	$(2, 62, 79, 25)$	$\times$	$(1, 2, 1, 1)$	$(78, 82, 72, 104)$
109	$(89, 50, 42, 85)$	$\times$	$(1, 2, 1, 1)$	$(62, 33, 9, 83)$
5	$(2, 4, 2, 2)$	$(1, 2, 1, 1)$	$\times$	$\times$
9	$(4, 6, 7, 0)$	$\times$	$(1, 2, 1, 1)$	$(7, 1, 4, 8)$

Comme les coefficients de $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ *sont positifs, les coefficients de* $\mathbf{s}$ *le sont aussi. L'utilisation d'une comparaison avec un entier* t *pour déterminer leur négativité est donc inutile.*

La correction éventuelle finale fait intervenir le calcul des coefficients MRS dans la base $\{1, 103, 103 \times 107\}$ *de* $\mathbf{s}$ *depuis ses résidus dans* $\mathcal{B}'$. *Il suffit alors de comparer les coefficients MRS de* $\mathbf{s}$ *aux coefficients MRS de* $2 \times d$, *qui sont* $(2d)_{mrs} = (32, 18, 52)$. *Par exemple, les coefficients MRS de* $\mathbf{s}_1$ *sont :*

$$\begin{cases} \tilde{s}_{103} = 37 \\ \tilde{s}_{107} = \left|(78 - 37\dfrac{1}{103}\right|_{107} = 70 \\ \tilde{s}_{109} = \left|((62 - 37)\dfrac{1}{103} - 70)\dfrac{1}{107}\right|_{109} = 28 \end{cases}$$

Comme $(\mathbf{s}_1)_{mrs} = (37, 70, 28) < (2d)_{mrs} = (32, 18, 52)$*, aucune correction sur le premier coefficient* $\mathbf{s}_1 \bmod m_\sigma$ *n'est requis. De manière analogue et en passant sur les détails pour les trois autres coefficients, le résultat final obtenu est* $\begin{pmatrix} 7 & 1 & 4 & 8 \end{pmatrix}$.

Analyse comparative

Les complexités (4.19) et (4.26) des deux versions de l'algorithme calculant la réduction des coefficients du vecteur $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ modulo $2d$ dépendent des tailles des bases RNS utilisées. Nous rappelons que le nombre de moduli de la base $\mathcal{B}$ de l'Algorithme 20 est $n_1 = \lceil \log_\beta (\mathcal{N} c_\infty + 1) + \log_\beta (2d) \rceil$, celui de la base principale $\mathcal{B}$ de l'Algorithme 21 est $n_2 = \lceil \log_\beta (\mathcal{N} c_\infty + 1) \rceil$, et la base auxiliaire $\mathcal{B}'$ possède quant à elle $\ell_2 = \lceil \log_\beta (4d) \rceil$ moduli. Par conséquent, afin de simplifier la comparaison, nous pouvons raisonnablement considérer que le nombre de moduli de taille β est le même pour les deux méthodes proposées :

$$n_1 = n_2 + \ell_2.$$

Afin de tirer avantage des propriétés de parallélisation du RNS, le nombre d'étapes élémentaires de calcul pour exécuter les deux versions d'algorithme proposées est détaillé par la suite. Par définition, une étape élémentaire de calcul dans un canal $\mathbb{Z}/m\mathbb{Z}$, notée `ETEl`, est constituée d'une opération du type $a \leftarrow |a + b \times c|_m$, typiquement exécutable en un cycle par un Rower pour les conversions basées sur le théorème des restes chinois, ou du type $a \leftarrow |(a + b) \times c|_m$ pour les conversions basées sur le MRS.

Nombre d'étapes de calcul pour l'Algorithme 20 Comme souligné dans la partie 1.2.4 concernant le calcul des coefficients MRS, après l'étape de calcul des résidus $\boldsymbol{q}_\mathcal{B}$, leur conversion en MRS s'effectue au mieux en $n_1 - 1$ étapes. L'étape du calcul du coefficient $\mathbf{s}_i$ porte le total à $n_1 + 1$ étapes. Le Tableau 4.1 donne le détail de ces étapes. Il découle alors que la réduction du vecteur entier, si exécutée séquentiellement sur chacun des coefficients, s'exécute en $\mathcal{N}(n_1 + 1)$ `ETEl`.

étape	$\mathcal{B}$					$\{m_\sigma\}$
1	$\tilde{q}_1$	q_2	q_3	$\dots$	q_{n_1}	$\mathbf{s}_i \leftarrow A \times \left\lvert PM^{-1} \right\rvert_{m_\sigma}$
2	$\times$	$\tilde{q}_2$	q_3	$\dots$	q_{n_1}	$\mathbf{s}_i \leftarrow \mathbf{s}_i + \tilde{q}_1 \times \left\lvert PM^{-1} \right\rvert_{m_\sigma}$
3	$\times$	$\times$	$\tilde{q}_3$	$\dots$	q_{n_1}	$\mathbf{s}_i \leftarrow \mathbf{s}_i + \tilde{q}_2 \times \left\lvert m_1 PM^{-1} \right\rvert_{m_\sigma}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n_1	$\times$	$\times$	$\times$	$\times$	$\tilde{q}_{n_1}$	$\mathbf{s}_i \leftarrow \mathbf{s}_i + \tilde{q}_{n_1-1} \times \left\lvert m_1 \dots m_{n_1-2} PM^{-1} \right\rvert_{m_\sigma}$
$n_1 + 1$	$\times$	$\times$	$\times$	$\times$	$\times$	$\mathbf{s}_i \leftarrow \mathbf{s}_i + \tilde{q}_{n_1} \times \left\lvert m_1 \dots m_{n_1-1} PM^{-1} \right\rvert_{m_\sigma}$

TABLE 4.1 – *Étapes de calcul de l'Algorithme 20 pour un coefficient (notations :* $A = \left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i$ *pour* $i \in [\![1, \mathcal{N}]\!]$ *et* $P = 2d$*).*

Nombre d'étapes de calcul dans les canaux RNS pour l'Algorithme 21 En parallélisant au mieux l'Algorithme 21 sur l'ensemble des canaux de $\mathcal{B} \cup \mathcal{B}' \cup \{m_\sigma, \tilde{m}\}$, le nombre d'étapes nécessaires pour la réduction d'un coefficient du vecteur d'entrée $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ dans le cas d'une réduction nécessitant deux comparaisons finales pour la correction est $n_2 + \ell_2 + 5$. Soit un total de $\mathcal{N}(n_2 + \ell_2 + 5)$ `ETEl`

pour une exécution séquentielle sur les coefficients. L'avantage de cette méthode par rapport à la précédente est qu'en divisant l'étape 1 du Tableau 4.2 en deux étapes successives, il est aisé de diviser par 2 le nombre de canaux RNS actifs simultanément.

étape	$\mathcal{B}$	$\mathcal{B}'$	$\{m_\sigma\}$	$\{\tilde{m}\}$
1	$\xi_{j,q,\mathcal{B}}$	$r_m \leftarrow A \times \left\lvert PM^{-1}\right\rvert_m$		
2	$\times$	$r_m \leftarrow r_m + \xi_{1,q,\mathcal{B}} \left\lvert P(m_1\tilde{m})^{-1}\right\rvert_m$		$r_{\tilde{m}} \leftarrow r_{\tilde{m}} + \xi_{1,q,\mathcal{B}} \left\lvert Pm_1^{-1}\right\rvert_{\tilde{m}}$
$\vdots$	$\times$	$\vdots$		$\vdots$
n_2+1	$\times$	$r_m \leftarrow r_m + \xi_{n_2,q,\mathcal{B}} \left\lvert P(m_{n_2}\tilde{m})^{-1}\right\rvert_m$		$r_{\tilde{m}} \leftarrow r_{\tilde{m}} + \xi_{n_2,q,\mathcal{B}} \left\lvert Pm_{n_2}^{-1}\right\rvert_{\tilde{m}}$
n_2+2	$\times$	$\times$	$\times$	$\tilde{q} \leftarrow -r_{\tilde{m}} \times \left\lvert P^{-1}\right\rvert_{\tilde{m}}$
n_2+3	$\times$	$s_m \leftarrow r_m + \tilde{q} \times \left\lvert P\tilde{m}^{-1}\right\rvert_m$		$\times$
n_2+4	$\times$	$\tilde{s}_{m'_1} \quad \dots \quad s_{m'_{\ell_2}}$	$\times$	$\times$
$\vdots$	$\vdots$	$\vdots \quad \vdots \quad \vdots$	$\vdots$	$\vdots$
$n_2+\ell_2+2$	$\times$	$\tilde{s}_{m'_1} \quad \dots \quad \tilde{s}_{m'_{\ell_2}}$	$\times$	$\times$
$n_2+\ell_2+3$	$\times$	$\overset{?}{\geqslant} t$	$\times$	$\times$
$n_2+\ell_2+4$	$\times$	$\overset{?}{\geqslant} P$	$\times$	$\times$
$n_2+\ell_2+5$	$\times$	$\times$	$\mathbf{s}_i \leftarrow s_{m_\sigma}\ (\pm P)$	$\times$

TABLE 4.2 – *Étapes de calcul dans les canaux RNS de l'Algorithme 21 dans le cas d'une double comparaison finale et pour un coefficient (notations : $A = \left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_i$ pour $i \in [\![1, \mathcal{N}]\!]$ et $P = 2d$).*

Conclusion

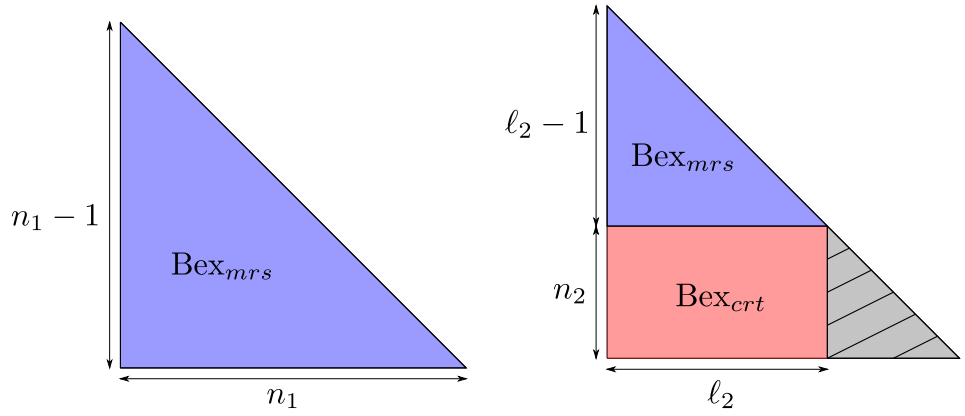
Le Tableau 4.3 illustre les différences d'efficacité des deux algorithmes précédents. Ceux-ci emploient un même nombre total de moduli de taille β, et s'exécutent en un nombre d'étapes sensiblement identique. La première approche exploite peu le RNS, puisqu'elle recourt précocement à une conversion en MRS, contrairement à la seconde approche, qui nécessite alors l'emploi de deux conversions de base.

	Algorithme 20	Algorithme 21
$\texttt{MME1}_\beta$	$\frac{n_1(n_1+1)}{2} = \frac{n_2^2+\ell_2^2}{2} + n_2\ell_2 + \frac{n_2+\ell_2}{2}$	$n_2\ell_2 + \frac{\ell_2^2}{2} + \frac{7\ell_2}{2} + n_2$
$\texttt{AME1}_\beta$	$\frac{n_1(n_1-1)}{2} = \frac{n_2^2+\ell_2^2}{2} + n_2\ell_2 - \frac{n_2+\ell_2}{2}$	$n_2\ell_2 + \frac{\ell_2^2}{2} + \frac{3\ell_2}{2}$
nb $\texttt{ETE1}$ Bex_1	$n_1 = n_2 + \ell_2$	$n_2 + 1$
nb $\texttt{ETE1}$ Bex_2	$\times$	$\ell_2 - 1$
nb $\texttt{ETE1}$ total $I \to O$	$n_1 + 1 = n_2 + \ell_2 + 1$	$n_2 + \ell_2 + 5$

TABLE 4.3 – *Complexités et nombre d'étapes de calcul des Algorithmes 20 et 21 pour la réduction d'un coefficient de $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$.*

Dans les deux méthodes, la présence d'une conversion dans le système po-

sitionnel MRS est un facteur limitant l'efficacité de l'approche RNS proposée. Comme cette conversion dans l'Algorithme 21 s'applique à une base $\mathcal{B}$ plus petite que celle de l'Algorithme 20, l'impact est alors moindre. Par exemple, le Tableau 4.2 concernant l'Algorithme 21 montre que si l'étape 1 est divisée en deux étapes consécutives, la première étant consacrée aux calculs dans les canaux de $\mathcal{B}$ et la seconde à ceux dans les canaux de $\mathcal{B}' \cup \{m_\sigma, \tilde{m}\}$, alors il est possible d'abaisser le niveau de parallélisation à $\ell_2 + 2$ canaux. En revanche, la conversion de base mise en œuvre dans l'Algorithme 20 étant de type MRS, il devient compliqué d'abaisser le niveau de parallélisation tout en limitant l'augmentation du nombre d'étapes. De plus, la présence de la conversion de type TRC dans la seconde version permet notamment de diminuer la complexité totale de $\frac{n_2(n_2-1)}{2} - 3\ell_2$ `MME1`. La Figure 4.6 montre comment ce coût quadratique gagné par la seconde approche par rapport à la première intervient au niveau des conversions de base.



(a) Algorithme 20 (b) Algorithme 21

FIGURE 4.6 – *Illustration de la différence de coût due aux conversions de base RNS entre les algorithmes (a) 20 et (b) 21, sachant que* $n_1 \sim n_2 + \ell_2$.

Finalement, si le second algorithme nécessite deux conversions de base au lieu d'une, il bénéficie mieux des avantages apportés par le RNS. Dans la partie qui suit, nous allons utiliser cette approche pour proposer un algorithme RNS-MRS de résolution du CVP par round-off.

4.2.3 Schéma général d'un algorithme RNS-MRS pour la résolution du CVP

Un algorithme permettant de calculer la Fonction (4.3) (p. 130) et basé sur l'Algorithme 21 (p. 142) va être présenté. Autrement dit, cet algorithme calcule le vecteur $\left(\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}\right) \operatorname{modc} m_\sigma$ grâce à

$$\left| \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \right|_{m_\sigma} = \left| |2d|_{m_\sigma}^{-1} \times \left\{ 2\mathbf{c}\mathbf{R}' + \mathbf{d} - \left(2\mathbf{c}\mathbf{R}' + \mathbf{d}\right) \bmod (2d) \right\} \right|_{m_\sigma} \tag{4.27}$$

via la réduction modulaire RedModRNS $\left(\mathcal{B}, \{m_\sigma\}, \mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}, 2d\right)$, où les notations suivantes restent en vigueur :

$$\begin{cases} \tilde{\mathbf{R}} = 2\tilde{m}M \times \mathbf{R}' \bmod (2d), \\ \tilde{\mathbf{d}} = \tilde{m}M \times \mathbf{d} \bmod (2d). \end{cases} \tag{4.28}$$

Afin de diminuer au mieux le nombre d'étapes de calcul, nous détaillons dans un premier temps les précalculs possibles. Pour alléger l'écriture, l'entier δ désignera $2d$:

$$\delta \stackrel{def}{=} 2d. \tag{4.29}$$

Précalculs dans $\mathcal{B}$

Ces précalculs permettent de réduire le nombre d'étapes de l'Algorithme 21 implantant la fonction ReducExacte_v2 en regroupant le calcul des résidus $\boldsymbol{q}_{\mathcal{B}}$ et de leurs coefficients $\xi_{q,\mathcal{B},i}$ pour la conversion Bex_{crt} vers $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$. Pour chaque $m_i \in \mathcal{B}$, nous définissons donc :

$$\boxed{\forall m_i \in \mathcal{B}, \begin{cases} \tilde{\mathbf{R}}_{m_i} = -\dfrac{1}{\delta M_i} \times \tilde{\mathbf{R}} \bmod m_i \\ \tilde{\mathbf{d}}_{m_i} = -\dfrac{1}{\delta M_i} \times \tilde{\mathbf{d}} \bmod m_i \end{cases}} \tag{4.30}$$

Ainsi, pour chaque modulus $m_i \in \mathcal{B}$, et pour chaque indice $k \in [\![1, \mathcal{N}]\!]$, nous pouvons écrire :

$$\xi_{q,\mathcal{B},i} = \left(\mathbf{c}\tilde{\mathbf{R}}_{m_i} + \tilde{\mathbf{d}}_{m_i}\right)_k \bmod m_i.$$

Par conséquent, le coefficient $\widehat{\mathbf{q}}_k$ se retrouve par l'égalité suivante :

$$\widehat{\mathbf{q}}_k = \sum_{i=1}^{n} \xi_{q_k,\mathcal{B},i} M_i = \sum_{i=1}^{n} \left|\left(\mathbf{c}\tilde{\mathbf{R}}_{m_i} + \tilde{\mathbf{d}}_{m_i}\right)_k\right|_{m_i} M_i = \sum_{i=1}^{n} \mathbf{a}_k^{(m_i)} M_i, \tag{4.31}$$

où les vecteurs $\mathbf{a}^{(m_i)}$ sont définis par :

$$\mathbf{a}^{(m_i)} = \mathbf{c}\tilde{\mathbf{R}}_{m_i} + \tilde{\mathbf{d}}_{m_i} \bmod m_i. \tag{4.32}$$

Précalculs dans $\mathcal{B}'$

Il s'agit de regrouper au mieux dans l'Algorithme 21 l'étape de reconstruction de $\widehat{\mathbf{q}}_k = \sum_{i=1}^{n} \left(\mathbf{c}\tilde{\mathbf{R}}_{m_i} + \tilde{\mathbf{d}}_{m_i}\right)_k M_i$ avec le calcul des quantités $r_m = \frac{\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_k + \widehat{\mathbf{q}}_k \delta}{M}$ et $s_m = \frac{r_m + \tilde{\mathbf{q}}_k \delta}{\tilde{m}}$ pour chaque $m \in \mathcal{B}'$. Le détail du calcul du résidu s_m est le suivant :

$$\begin{aligned} s_m &= \frac{r_m + \tilde{\mathbf{q}}_k \delta}{\tilde{m}} \bmod m \\ &= \frac{\frac{\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_k + \widehat{\mathbf{q}}_k \delta}{M} + \tilde{\mathbf{q}}_k \delta}{\tilde{m}} \bmod m \\ &= \frac{\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_k + \left(\sum_{i=1}^{n} \mathbf{a}_k^{(m_i)} M_i\right) \delta}{\tilde{m} M} + \frac{\tilde{\mathbf{q}}_k \delta}{\tilde{m}} \bmod m \\ &= \frac{1}{\tilde{m} M} \left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)_k + \left(\sum_{i=1}^{n} \frac{\mathbf{a}_k^{(m_i)}}{m_i}\right) \frac{\delta}{\tilde{m}} + \tilde{\mathbf{q}}_k \frac{\delta}{\tilde{m}} \bmod m \\ &= \left(\left|2\mathbf{c}\mathbf{R}' + \mathbf{d}\right|_\delta + \delta \mathbf{v}_\mathbf{e}\right)_k \bmod m. \end{aligned}$$

D'où les précalculs suivants :

$$\boxed{\forall m'_j \in \mathcal{B}', \begin{cases} \tilde{\mathbf{R}}_{m'_j} = \dfrac{1}{\tilde{m}M} \times \tilde{\mathbf{R}} \bmod m'_j \\ \tilde{\mathbf{d}}_{m'_j} = \dfrac{1}{\tilde{m}M} \times \tilde{\mathbf{d}} \bmod m'_j \\ \tilde{\delta}^{(i)}_{m'_j} = \dfrac{\delta}{m_i \tilde{m}} \bmod m'_j,\ \forall m_i \in \mathcal{B} \\ \tilde{\delta}^{(0)}_{m'_j} = \dfrac{\delta}{\tilde{m}} \bmod m'_j \end{cases}} \tag{4.33}$$

La notation suivante est également introduite :

$$\mathbf{a}^{(m'_j)} = \mathbf{c}\tilde{\mathbf{R}}_{m'_j} + \tilde{\mathbf{d}}_{m'_j} \bmod m'_j. \tag{4.34}$$

Par conséquent, il vient alors pour $k \in [\![1, \mathcal{N}]\!]$ et pour tout $m \in \mathcal{B}'$:

$$\begin{aligned} s_m &= \left(\mathbf{c}\tilde{\mathbf{R}}_m + \tilde{\mathbf{d}}_m\right)_k + \sum_{i=1}^{n} \mathbf{a}_k^{(m_i)} \tilde{\delta}_m^{(i)} + \tilde{\mathbf{q}}_k \tilde{\delta}_m^{(0)} \bmod m \\ &= \mathbf{a}_k^{(m)} + \sum_{i=1}^{n} \mathbf{a}_k^{(m_i)} \tilde{\delta}_m^{(i)} + \tilde{\mathbf{q}}_k \tilde{\delta}_m^{(0)} \bmod m \\ &= \left(\left|2\mathbf{c}\mathbf{R}' + \mathbf{d}\right|_\delta + \delta \mathbf{v}_\mathbf{e}\right)_k \bmod m. \end{aligned} \tag{4.35}$$

Précalculs dans $\{m_\sigma\}$

Le canal $\mathbb{Z}/m_\sigma\mathbb{Z}$ est dédié au calcul de la formule complète $\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}$. Détailler celle-ci nous permettra d'extraire les précalculs possibles. En utilisant les Équations (4.27) et (4.14) avec $\mathbf{v}_\mathbf{e}$ désignant le vecteur d'erreur dû à la réduction modulaire de Montgomery incomplète, ainsi que les notations de l'Algorithme 21, il vient :

$$\begin{cases} \mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R} = \mathbf{c} - \dfrac{2\mathbf{c}\mathbf{R}' + \mathbf{d} - \left|2\mathbf{c}\mathbf{R}' + \mathbf{d}\right|_\delta}{\delta} \times \mathbf{R}, \\ \left|2\mathbf{c}\mathbf{R}' + \mathbf{d}\right|_\delta + \delta \mathbf{v}_\mathbf{e} = \dfrac{\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}} + \widehat{\mathbf{q}}\delta}{\tilde{m}M} + \dfrac{\delta}{\tilde{m}} \tilde{\mathbf{q}}. \end{cases}$$

Ainsi, en notant $\mathbf{v}_{1/2}$ le vecteur $\left(\frac{1}{2}, \ldots, \frac{1}{2}\right) \in \mathbb{Z}^\mathcal{N}$, et en rappelant que $\mathbf{R}' = \frac{\delta}{2}\mathbf{R}^{-1}$, la combinaison des deux précédentes équations donne :

$$\begin{aligned} \mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R} &= \mathbf{c} - \frac{2\mathbf{c}\mathbf{R}' + \mathbf{d}}{\delta}\mathbf{R} + \left(\frac{\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}} + \widehat{\mathbf{q}}\delta}{\delta \tilde{m} M} + \frac{1}{\tilde{m}}\tilde{\mathbf{q}} - \mathbf{v}_\mathbf{e}\right)\mathbf{R} \\ &= \mathbf{c} - \frac{2}{\delta}\mathbf{c}\mathbf{R}'\mathbf{R} - \mathbf{v}_{1/2}\mathbf{R} + \frac{1}{\delta \tilde{m} M}\mathbf{c}\tilde{\mathbf{R}}\mathbf{R} + \frac{1}{\delta \tilde{m} M}\tilde{\mathbf{d}}\mathbf{R} + \left(\frac{1}{\tilde{m}M}\widehat{\mathbf{q}} + \frac{1}{\tilde{m}}\tilde{\mathbf{q}} - \mathbf{v}_\mathbf{e}\right)\mathbf{R} \\ &= \frac{1}{\delta \tilde{m} M}\mathbf{c}\tilde{\mathbf{R}}\mathbf{R} + \left(\frac{1}{\delta \tilde{m} M}\tilde{\mathbf{d}} - \mathbf{v}_{1/2}\right)\mathbf{R} + \left(\frac{1}{\tilde{m}M}\widehat{\mathbf{q}} + \frac{1}{\tilde{m}}\tilde{\mathbf{q}} - \mathbf{v}_\mathbf{e}\right)\mathbf{R}. \end{aligned}$$

Or, nous avons vu précédemment (4.31) que $\widehat{\mathbf{q}} = \sum_{i=1}^{n} \left(\mathbf{c}\tilde{\mathbf{R}}_{m_i} + \tilde{\mathbf{d}}_{m_i}\right) M_i = \sum_{i=1}^{n} \mathbf{a}^{(m_i)} M_i$.
Par suite, nous obtenons finalement :

$$\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R} = \frac{1}{\delta \tilde{m} M} \mathbf{c}\tilde{\mathbf{R}}\mathbf{R} + \left(\frac{1}{\delta \tilde{m} M} \tilde{\mathbf{d}} - \mathbf{v}_{1/2} \right) \mathbf{R} + \left(\sum_{i=1}^{n} \frac{1}{\tilde{m} m_i} \mathbf{a}^{(m_i)} + \frac{1}{\tilde{m}} \tilde{\mathbf{q}} - \mathbf{v}_{\mathbf{e}} \right) \mathbf{R}.$$

Soit alors les constantes suivantes :

$$(m_\sigma) \begin{cases} \mathbf{R}_{m_\sigma} = \mathbf{R} \bmod m_\sigma \\ \tilde{\mathbf{R}}_{m_\sigma} = \dfrac{1}{\delta \tilde{m} M} \times \tilde{\mathbf{R}}\mathbf{R} \bmod m_\sigma \\ \tilde{\mathbf{d}}_{m_\sigma} = \left(\dfrac{1}{\delta \tilde{m} M} \tilde{\mathbf{d}} - \mathbf{v}_{1/2} \right) \times \mathbf{R} \bmod m_\sigma \\ \tilde{\delta}^{(i)}_{m_\sigma} = \dfrac{1}{m_i \tilde{m}} \bmod m_\sigma, \ \forall m_i \in \mathcal{B} \\ \tilde{\delta}^{(0)}_{m_\sigma} = \dfrac{1}{\tilde{m}} \bmod m_\sigma \end{cases} \tag{4.36}$$

ainsi que la notation :

$$\mathbf{b}^{(m_\sigma)} = \mathbf{c}\tilde{\mathbf{R}}_{m_\sigma} + \tilde{\mathbf{d}}_{m_\sigma} \bmod m_\sigma. \tag{4.37}$$

De ce fait, nous obtenons finalement l'expression suivante :

$$\left(\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R} \right) \bmod m_\sigma = \mathbf{b}^{(m_\sigma)} + \left(\sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m_\sigma} \mathbf{a}^{(m_i)} + \tilde{\delta}^{(0)}_{m_\sigma} \tilde{\mathbf{q}} - \mathbf{v}_{\mathbf{e}} \right) \mathbf{R}_{m_\sigma} \bmod m_\sigma. \tag{4.38}$$

Précalculs dans $\{\tilde{m}\}$

Il est ici question du calcul de $\tilde{q} = -r_{\tilde{m}} \delta^{-1} \bmod \tilde{m}$, où $r_{\tilde{m}}$ vaut :

$$\begin{aligned} r_{\tilde{m}} &= \frac{\left(\mathbf{c}\tilde{\mathbf{R}}_m + \tilde{\mathbf{d}}_m \right)_k + \widehat{\mathbf{q}}_k \delta}{M} \bmod \tilde{m} \\ &= \frac{1}{M} \left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}} \right)_k + \left(\sum_{i=1}^{n} \frac{\mathbf{a}_k^{(m_i)}}{m_i} \right) \delta \bmod \tilde{m}. \end{aligned}$$

Les précalculs introduits sont donc les suivants :

$$(\tilde{m}) \begin{cases} \tilde{\mathbf{R}}_{\tilde{m}} = -\dfrac{1}{\delta M} \times \tilde{\mathbf{R}} \bmod \tilde{m} \\ \tilde{\mathbf{d}}_{\tilde{m}} = -\dfrac{1}{\delta M} \times \tilde{\mathbf{d}} \bmod \tilde{m} \\ \tilde{\delta}^{(i)}_{\tilde{m}} = -\dfrac{1}{m_i} \bmod \tilde{m}, \ \forall m_i \in \mathcal{B} \end{cases} \tag{4.39}$$

puis la notation :

$$\mathbf{a}^{(\tilde{m})} = \mathbf{c}\tilde{\mathbf{R}}_{\tilde{m}} + \tilde{\mathbf{d}}_{\tilde{m}} \bmod \tilde{m}. \tag{4.40}$$

Ainsi, le calcul effectué dans le canal $\{\tilde{m}\}$ est le suivant :

$$\tilde{\mathbf{q}} = \mathbf{a}^{(\tilde{m})} + \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{\tilde{m}} \mathbf{a}^{(m_i)} \bmod \tilde{m}. \tag{4.41}$$

Conversions de base

Avec les notations précédentes, la première conversion, de la base $\mathcal{B}$ vers la base $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$, du vecteur ayant pour résidus les vecteurs $\left(\mathbf{a}^{(m)} = \left|\mathbf{c}\tilde{\mathbf{R}}_m + \tilde{\mathbf{d}}_m\right|_m\right)_{m\in\mathcal{B}}$ dans $\mathcal{B}$, se résume finalement au produit de matrices $(\ell+2, n) \times (n, \mathcal{N})$ suivant :

$$\begin{aligned}
&\begin{pmatrix} \tilde{\delta}^{(1)}_{m'_1} & \tilde{\delta}^{(2)}_{m'_1} & \dots & \tilde{\delta}^{(n)}_{m'_1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\delta}^{(1)}_{m'_\ell} & \tilde{\delta}^{(2)}_{m'_\ell} & \dots & \tilde{\delta}^{(n)}_{m'_\ell} \\ \tilde{\delta}^{(1)}_{\tilde{m}} & \tilde{\delta}^{(2)}_{\tilde{m}} & \dots & \tilde{\delta}^{(n)}_{\tilde{m}} \\ \tilde{\delta}^{(1)}_{m_\sigma} & \tilde{\delta}^{(2)}_{m_\sigma} & \dots & \tilde{\delta}^{(n)}_{m_\sigma} \end{pmatrix} \times \begin{pmatrix} \mathbf{a}_1^{(m_1)} & \mathbf{a}_2^{(m_1)} & \dots & \mathbf{a}_{\mathcal{N}}^{(m_1)} \\ \mathbf{a}_1^{(m_2)} & \mathbf{a}_2^{(m_2)} & \dots & \mathbf{a}_{\mathcal{N}}^{(m_2)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1^{(m_n)} & \mathbf{a}_2^{(m_n)} & \dots & \mathbf{a}_{\mathcal{N}}^{(m_n)} \end{pmatrix} \bmod \begin{pmatrix} m'_1 \\ m'_2 \\ \vdots \\ m'_\ell \\ \tilde{m} \\ m_\sigma \end{pmatrix} \\
&= \begin{pmatrix} \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m'_1} \mathbf{a}_1^{(m_i)} & \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m'_1} \mathbf{a}_2^{(m_i)} & \dots & \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m'_1} \mathbf{a}_{\mathcal{N}}^{(m_i)} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m'_\ell} \mathbf{a}_1^{(m_i)} & \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m'_\ell} \mathbf{a}_2^{(m_i)} & \dots & \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m'_\ell} \mathbf{a}_{\mathcal{N}}^{(m_i)} \\ \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m_\sigma} \mathbf{a}_1^{(m_i)} & \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m_\sigma} \mathbf{a}_2^{(m_i)} & \dots & \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m_\sigma} \mathbf{a}_{\mathcal{N}}^{(m_i)} \end{pmatrix} \bmod \begin{pmatrix} m'_1 \\ \vdots \\ m'_\ell \\ m_\sigma \end{pmatrix}
\end{aligned} \tag{4.42}$$

La seconde conversion de $\{\tilde{m}\}$ vers $\mathcal{B}' \cup \{m_\sigma\}$ du vecteur $\tilde{\mathbf{q}} = \begin{pmatrix} \tilde{q}_1 & \tilde{q}_2 & \dots & \tilde{q}_{\mathcal{N}} \end{pmatrix}$ est le produit vecteur-vecteur $(\ell+1, 1) \times (1, \mathcal{N})$ suivant :

$$\begin{pmatrix} \tilde{\delta}^{(0)}_{m'_1} \\ \tilde{\delta}^{(0)}_{m'_2} \\ \vdots \\ \tilde{\delta}^{(0)}_{m'_\ell} \\ \tilde{\delta}^{(0)}_{m_\sigma} \end{pmatrix} \times \tilde{\mathbf{q}} \bmod \begin{pmatrix} m'_1 \\ m'_2 \\ \vdots \\ m'_\ell \\ m_\sigma \end{pmatrix} = \begin{pmatrix} \tilde{\delta}^{(0)}_{m'_1} \tilde{\mathbf{q}} \bmod m'_1 \\ \tilde{\delta}^{(0)}_{m'_2} \tilde{\mathbf{q}} \bmod m'_2 \\ \vdots \\ \tilde{\delta}^{(0)}_{m'_\ell} \tilde{\mathbf{q}} \bmod m'_\ell \\ \tilde{\delta}^{(0)}_{m_\sigma} \tilde{\mathbf{q}} \bmod m_\sigma \end{pmatrix} \tag{4.43}$$

Algorithme final

L'algorithme final 22 utilise les quatre ensembles de précalculs introduits précédemment : (4.30), (4.33), (4.36) et (4.39). Les bases RNS $\mathcal{B}$ et $\mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$ vérifient les mêmes hypothèses que pour l'Algorithme 21, à savoir :

$$\begin{cases} |\mathcal{B}| = n, \ \left|\mathcal{B}'\right| = \ell \\ M > \mathcal{N} c_\infty + 1, \ M' > 4d, \ m_\sigma > 2\sigma + 1, \ \tilde{m} > n, \\ \operatorname{pgcd}(2d, M\tilde{m}m_\sigma) = 1, \ \operatorname{pgcd}\left(M'm_\sigma, M\tilde{m}\right) = 1, \ \operatorname{pgcd}(M, \tilde{m}) = 1. \end{cases} \tag{4.44}$$

À l'étape 10, le contenu de la variable vectorielle $\mathbf{a}^{(m_\sigma)}$ est donné par (4.38), c'est-à-dire

$$\mathbf{a}^{(m_\sigma)} = \sum_{i=1}^{n} \tilde{\delta}^{(i)}_{m_\sigma} \mathbf{a}^{(m_i)} + \tilde{\delta}^{(0)}_{m_\sigma} \tilde{\mathbf{q}}.$$

Le résultat obtenu à l'étape 10 dans les variables $\mathbf{a}^{(m)}$ pour $m \in \mathcal{B}'$ provient de la formule (4.35). Ainsi, après cette étape nous avons :

$$\mathbf{a}^{(m)} = \left(\left|2\mathbf{c}\mathbf{R}' + \mathbf{d}\right|_{2d} + 2d\mathbf{v}_\mathbf{e}\right) \bmod m \text{ pour tout } m \in \mathcal{B}'.$$

Algorithme 22 : CVP_RNS_MRS $(\mathbf{c}, \mathbf{R})$

Données : Les $n + \ell + 2$ vecteurs $\mathbf{c}_m = \mathbf{c} \bmod m$ pour tout $m \in \mathcal{B} \cup \mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$. Les valeurs précalculées (4.30), (4.33), (4.36) et (4.39), ainsi que les coefficients dans la base MRS associée à $\mathcal{B}'$ de $2d$, notées $(2d)_{mrs}$ et ceux d'un nombre $t \in [\![\left(1 + \frac{n}{\tilde{m}} 2d, M' - \frac{2d}{\tilde{m}}\right)]\!]$ notés t_{mrs}.

Résultat : $\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \times \mathbf{R} \bmod m_\sigma$.

début
- **pour** $m \in \mathcal{B} \cup \mathcal{B}' \cup \{\tilde{m}\}$ **faire**
  - $\mathbf{a}^{(m)} \leftarrow \mathbf{c}_m \tilde{\mathbf{R}}_m + \tilde{\mathbf{d}}_m \bmod m$
- $\mathbf{b}^{(m_\sigma)} \leftarrow \mathbf{c}_m \tilde{\mathbf{R}}_{m_\sigma} + \tilde{\mathbf{d}}_{m_\sigma} \bmod m_\sigma$
- $\mathbf{a}^{(m_\sigma)} \leftarrow (0, \ldots, 0) \in \mathbb{Z}^{\mathcal{N}}$
- **pour** $m \in \mathcal{B}' \cup \{m_\sigma\}$ **faire**
  - $\mathbf{a}^{(m)} \leftarrow \mathbf{a}^{(m)} + \sum_{i=1}^{n} \tilde{\delta}_m^{(i)} \times \mathbf{a}^{(m_i)} \bmod m$
    `/* 1ère réduc. de Montgomery */`
- $\tilde{q} \leftarrow \mathbf{a}^{(\tilde{m})} + \sum_{i=1}^{n} \tilde{\delta}_{\tilde{m}}^{(i)} \times \mathbf{a}^{(m_i)} \bmod \tilde{m}$
  `/* cf. formule (4.41) */`
- **pour** $m \in \mathcal{B}' \cup \{m_\sigma\}$ **faire**
  - $\mathbf{a}^{(m)} \leftarrow \mathbf{a}^{(m)} + \tilde{q} \times \tilde{\delta}_m^{(0)} \bmod m$
    `/* 2nde réduc. de Montgomery */`
- **pour** $k \leftarrow 1$ **à** $\mathcal{N}$ **faire**
  - `/* en // sur chaque coeff. */`
  - $\mathbf{a}_{k,mrs} \leftarrow \text{MRSCoeff}\left(\mathcal{B}', \left(\mathbf{a}_k^{(m'_1)}, \ldots, \mathbf{a}_k^{(m'_\ell)}\right)\right)$
  - **si** $\mathbf{a}_{k,mrs} > t_{mrs}$ **alors**
    - $\mathbf{a}_k^{(m_\sigma)} \leftarrow \mathbf{a}_k^{(m_\sigma)} + 1 \bmod m_\sigma$
  - **sinon si** $\mathbf{a}_{k,mrs} \geqslant (2d)_{mrs}$ **alors**
    - $\mathbf{a}_k^{(m_\sigma)} \leftarrow \mathbf{a}_k^{(m_\sigma)} - 1 \bmod m_\sigma$
- $\mathbf{p}_{m_\sigma} \leftarrow \mathbf{b}^{(m_\sigma)} + \mathbf{a}^{(m_\sigma)} \mathbf{R}_{m_\sigma} \bmod m_\sigma$
- **retourner** $\mathbf{p}_{m_\sigma}$

Par conséquent, la boucle de l'étape 11 est dédiée à la conversion en MRS de chaque coefficient du vecteur $(|2\mathbf{cR}' + \mathbf{d}|_{2d} + 2d\mathbf{v_e})$ ce qui permet de retrouver le vecteur d'erreur $\mathbf{v_e}$ dû à une réduction de Montgomery possiblement incomplète. Celui-ci prend ses valeurs dans $\{-1, 0, 1\}$, et chacun de ses coefficients $(\mathbf{v_e})_k$ se retrouve par (cf. (4.24)) :

$$\begin{cases} (\mathbf{v_e})_k = -1 \text{ si } \mathbf{a}_{k,mrs} > t_{mrs} \\ (\mathbf{v_e})_k = 0 \text{ si } 0 \leqslant \mathbf{a}_{k,mrs} < p_{mrs} \\ (\mathbf{v_e})_k = 1 \text{ si } p_{mrs} \leqslant \mathbf{a}_{k,mrs} < t_{mrs} \end{cases} \tag{4.45}$$

Il est ainsi possible de corriger $\mathbf{a}^{(m_\sigma)}$ (étapes 13 à 16). Par suite, à l'étape 17 nous avons obtenu $\mathbf{a}^{(m_\sigma)} \leftarrow \sum_{i=1}^{n} \tilde{\delta}_{m_\sigma}^{(i)} \mathbf{a}^{(m_i)} + \tilde{\delta}_{m_\sigma}^{(0)} \tilde{\mathbf{q}} - \mathbf{v_e}$. Finalement, la valeur contenue dans la variable $\mathbf{p}_{m_\sigma}$ est déduite de (4.38), à savoir :

$$\mathbf{p}_{m_\sigma} = \mathbf{b}^{(m_\sigma)} + \mathbf{a}^{(m_\sigma)} \mathbf{R}_{m_\sigma} \bmod m_\sigma = \Big(\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}\Big) \bmod m_\sigma.$$

Complexités spatiale et temporelle

Pour étudier les coûts de l'Algorithme 22, il convient de préciser que les tailles des moduli de $\mathcal{B}$ et $\mathcal{B}'$ sont usuellement fixées par une contrainte matérielle, comme la taille du mot-machine. Nous la notons β. Vu la Condition (4.44) sur la taille des bases $\mathcal{B}$ et $\mathcal{B}'$, n peut être approximé par $\log_\beta(\mathcal{N} c_\infty)$, et ℓ par $\log_\beta(4d)$. De même, $\tilde{m} \sim n \sim \log_\beta(\mathcal{N} c_\infty)$, et $m_\sigma \sim 2\sigma$.

De plus, comme n est un nombre d'entiers supposés inférieurs à β, il en découle immédiatement que $\tilde{m}$ tient sur un unique β-mot. Le Tableau 4.4 montre par exemple le nombre de bits suffisant pour représenter l'ensemble des nombres premiers contenu sur un β-mot, pour plusieurs valeurs de β.

β	2^{16}	2^{20}	2^{24}	2^{28}	2^{32}	2^{36}	2^{40}
$\lceil \log_2(\pi(\beta)) \rceil$	13	17	21	24	28	32	36

TABLE 4.4 – *Taille binaire du nombre total de moduli premiers constitués d'un digit en base β, pour différents β.*

De plus, à β fixé, le nombre de premiers tenant sur un unique β-mot suffit à satisfaire les besoins des bases $\mathcal{B}$ et $\mathcal{B}'$ pour les ordres de grandeur des dimensions de réseau considérées en pratique. Étant donné que la réduction des coefficients de $\mathbf{c}$ par d ne modifie pas le résultat, nous pouvons poser $c_\infty = d$. Ensuite, nous considérons par exemple la suggestion de Micciancio (2001) concernant le choix de la base $\mathbf{R}$. Celle-ci est issue de la réduction d'une matrice choisie dans $[\![-\mathcal{N}, \mathcal{N}]\!]^{\mathcal{N}^2}$. La borne de Hadamard sur le déterminant d de $\mathbf{R}$ est $d \leqslant \mathcal{N}^{\frac{3}{2}\mathcal{N}}$. Par conséquent, il suffit que $\mathcal{B}$ et $\mathcal{B}'$ vérifient

$$M > \mathcal{N} d + 1 = \mathcal{N}^{\frac{3}{2}\mathcal{N}+1} + 1 \text{ et } M' > 4d = 4\mathcal{N}^{\frac{3}{2}\mathcal{N}}. \tag{4.46}$$

Le Tableau 4.5 fournit, pour plusieurs tailles de mot β, le nombre de premiers suffisant pour donner les n et ℓ moduli des bases $\mathcal{B}$ et $\mathcal{B}'$, de manière à ce que leur produit soit supérieur à $(\mathcal{N}^{\frac{3}{2}\mathcal{N}+1} + 1) \times 4\mathcal{N}^{\frac{3}{2}\mathcal{N}}$, ainsi que les quantités $\lceil \log_2(n + \ell) \rceil$, ce qui montre que dans tous les cas $\tilde{m}$ tient sur un β-mot. Le cas du modulus m_σ est considéré à part puisqu'en pratique le paramètre σ peut se

$\mathcal{N}$	$\beta = 2^{16}$	$\beta = 2^{24}$	$\beta = 2^{32}$	$\beta = 2^{48}$	$\beta = 2^{64}$
100	$(126, 7)$	$(84, 7, pm)$	$(63, 6, pm)$	$(42, 6, pm)$	$(32, 5, pm)$
200	$(288, 9)$	$(192, 8, pm)$	$(144, 8, pm)$	$(96, 7, pm)$	$(72, 7, pm)$
400	$(653, 10)$	$(433, 9)$	$(325, 9, pm)$	$(217, 8, pm)$	$(163, 8, pm)$
800	$(1466, 11)$	$(965, 10)$	$(724, 10, pm)$	$(483, 9, pm)$	$(362, 9, pm)$
1600	$(3297, 12)$	$(2130, 12)$	$(1598, 11, pm)$	$(1065, 11, pm)$	$(799, 10, pm)$

TABLE 4.5 – *Nombre suffisant de $n + \ell$ moduli premiers d'un β-digit pour $\mathcal{B} \cup \mathcal{B}'$ vérifiant (4.46) et taille binaire $\log_2(n + \ell)$ pour différents β et différentes dimensions de réseau $\mathcal{N}$ (pm : tous les moduli sont pseudo Mersenne).*

révéler sans commune mesure avec le paramètre β (*e.g.* σ est de l'ordre de 2 ou 3 dans l'article originel de Goldreich et al. (1997)).

Dans la suite, nous considérons donc que les moduli de $\mathcal{B} \cup \mathcal{B}' \cup \{\tilde{m}\}$ s'écrivent en un β-mot. Le Tableau 4.6 récapitule le coût des précalculs (4.30), (4.33), (4.39) et (4.36).

base	$\mathcal{B}$	$\mathcal{B}'$	$\{\tilde{m}\}$	$\{m_\sigma\}$
β-mots	$n\mathcal{N}(\mathcal{N}+1)$	$\ell\mathcal{N}(\mathcal{N}+1)$ $+\ell(n+1)$	$\mathcal{N}(\mathcal{N}+1)+n$	$\times$
m_σ-mots	$\times$	$\times$	$\times$	$2\mathcal{N}^2+\mathcal{N}+n+1$

TABLE 4.6 – *Coût des précalculs de l'Algorithme 22.*

Ainsi, la complexité spatiale binaire est la suivante :

$$\begin{aligned}\mathcal{C}_S\left(\text{CVP\_RNS\_MRS}\right) &= \left(2\mathcal{N}^2+\mathcal{N}+1+\log_\beta\left(c_\infty\mathcal{N}\right)\right)\log_2\left(2\sigma\right)\\ &+\left(\mathcal{N}^2+\mathcal{N}\right)\log_2\left(4dc_\infty\beta\mathcal{N}\right)+\log_2\left(c_\infty\mathcal{N}\right)\log_\beta\left(4d\beta\right)+\log_2\left(4d\right).\end{aligned} \tag{4.47}$$

Le coût de l'Algorithme 22 en termes d'opérations élémentaires est détaillé dans le Tableau 4.7, et est le suivant :

$$\begin{aligned}\mathcal{C}_T\left(\text{CVP\_RNS\_MRS}\right) &= \left[2\mathcal{N}^2+\log_\beta(c_\infty\beta\mathcal{N})\mathcal{N}\right]\texttt{A/MMEl}_{2\sigma}\\ &+\left[\log_\beta\left(4dc_\infty\beta\mathcal{N}\right)\mathcal{N}^2+\log_\beta(2d^{\frac{1}{2}}c_\infty\mathcal{N})\log_\beta(4d\beta)\mathcal{N}\right]\texttt{A/MMEl}_\beta.\end{aligned} \tag{4.48}$$

Étape	$\mathcal{B}$	$\mathcal{B}'$	$\{\tilde{m}\}$	$\{m_\sigma\}$
3	$n\mathcal{N}^2$	$\ell\mathcal{N}^2$	$\mathcal{N}^2$	$\times$
4	$\times$	$\times$	$\times$	$\mathcal{N}^2$
7	$\times$	$n\ell\mathcal{N}$	$\times$	$n\mathcal{N}$
8	$\times$	$\times$	$n\mathcal{N}$	$\times$
10	$\times$	$\ell\mathcal{N}$	$\times$	$\mathcal{N}$
12	$\times$	$\frac{\ell(\ell-1)}{2}\mathcal{N}$	$\times$	$\times$
17	$\times$	$\times$	$\times$	$\mathcal{N}^2$
Total	$n\mathcal{N}^2$	$\ell\mathcal{N}^2$ $+\left(n+\frac{\ell+1}{2}\right)\ell\mathcal{N}$	$\mathcal{N}^2+n\mathcal{N}$	$2\mathcal{N}^2+(n+1)\mathcal{N}$

TABLE 4.7 – *Nombre de multiplications et additions modulaires élémentaires de l'Algorithme 22.*

En supposant que $c_\infty \leqslant d$, nous pouvons poser l'hypothèse $n \geqslant \ell + 2$. Ainsi, pour une parallélisation optimale sur n canaux, l'étape 3 requiert pour chaque

coefficient $\mathcal{N}^2$ `ETEl` en parallèle dans $\mathcal{B} \cup \mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$. Ensuite, les étapes 6 à 16 s'exécutent au plus $n + \ell + 4$ `ETEl` pour chaque coefficient. Le calcul final de $\mathbf{p}_{m_\sigma}$ est réalisé en $\mathcal{N}^2$ `ETEl` dans $\mathbb{Z}/m_\sigma\mathbb{Z}$.

Finalement, le temps d'exécution pour une parallélisation optimale s'évalue en :

$$\mathcal{C}_{\texttt{ETEl}}\,(\text{CVP\_RNS\_MRS}) = \left[\mathcal{N}^2 + \mathcal{N}\,(n + \ell + 4)\right] \ \texttt{ETEl} + \mathcal{N}^2 \ \texttt{ETEl}_{m_\sigma}. \quad (4.49)$$

Exemple 4.6 *Nous considérons le réseau de $\mathbb{Z}^4$ défini par la matrice suivante :*

$$\mathbf{R} = \begin{pmatrix} 1 & -2 & -2 & 15 \\ -17 & 1 & 1 & 1 \\ -3 & 11 & -13 & 6 \\ 2 & 18 & 6 & -2 \end{pmatrix},\ d = \det \mathbf{R} = 74268,\ \rho_{\mathbf{R}} < \frac{1}{6}.$$

Nous avons donc $\mathcal{P}_\sigma = [\![-3, 3]\!]^4$. Nous considérons par exemple le message $\mathbf{p} = \begin{pmatrix} -3 & 2 & 1 & -2 \end{pmatrix}$. Soit le chiffré réduit modulo d suivant :

$$\begin{aligned} \mathbf{c} &= \begin{pmatrix} 22143 & 357 & 7328 & 17985 \end{pmatrix} \\ &= \begin{pmatrix} -3 & 2 & 1 & -2 \end{pmatrix} + \begin{pmatrix} 1171 & 3230 & -397 & 213 \end{pmatrix} \mathbf{R}. \end{aligned}$$

Soit les bases RNS $\mathcal{B} = \{19, 23, 29, 31\}$ ($n = 4$), $\mathcal{B}' = \{13, 17, 37, 41\}$ ($\ell = 4$), $\tilde{m} = 11$ et $m_\sigma = 7$. Ainsi, ces bases sont bien copremières, et $\text{pgcd}(m_\sigma \tilde{m} M', 2d) = 1$. De plus, $M = 392863 > 4d + 1 = 297073$, $M' = 335257 > 4d = 297072$, $\tilde{m} > n + 1 = 5$, et $m_\sigma \geqslant 2\sigma + 1 = 7$.

L'exécution de l'Algorithme 22 doit donc retourner $\left(\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}\right) \bmod m_\sigma = \begin{pmatrix} -3 & 2 & 1 & -2 \end{pmatrix} \bmod m_\sigma = \begin{pmatrix} 4 & 2 & 1 & 5 \end{pmatrix}$.

- *Résidus de $\mathbf{c}$ dans $\mathcal{B} \cup \mathcal{B}' \cup \{\tilde{m}, m_\sigma\}$:*

$$\begin{aligned} \mathcal{B} &: \begin{pmatrix} 8 & 15 & 13 & 11 \end{pmatrix}, \begin{pmatrix} 17 & 12 & 14 & 22 \end{pmatrix}, \begin{pmatrix} 16 & 9 & 20 & 5 \end{pmatrix}, \begin{pmatrix} 9 & 16 & 12 & 5 \end{pmatrix} \\ \mathcal{B}' &: \begin{pmatrix} 4 & 6 & 9 & 6 \end{pmatrix}, \begin{pmatrix} 9 & 0 & 1 & 16 \end{pmatrix}, \begin{pmatrix} 17 & 24 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 3 & 29 & 30 & 27 \end{pmatrix} \\ \tilde{m} &: \begin{pmatrix} 0 & 5 & 2 & 0 \end{pmatrix} \\ m_\sigma &: \begin{pmatrix} 2 & 0 & 6 & 2 \end{pmatrix} \end{aligned}$$

- *Les précalculs se basent notamment sur les données suivantes :*

$$\mathbf{R}' = \begin{pmatrix} 420 & -4260 & -204 & 408 \\ -184 & 98 & 1504 & 3181 \\ 2140 & 1282 & -4576 & 2963 \\ 5184 & 468 & -396 & 792 \end{pmatrix},$$

$$\tilde{\mathbf{R}} = \left|2\tilde{m} M \mathbf{R}'\right|_{2d} = \begin{pmatrix} 131352 & 131856 & 101712 & 93648 \\ 65528 & 60356 & 71440 & 67546 \\ 138784 & 116596 & 79712 & 75758 \\ 97704 & 133632 & 92592 & 111888 \end{pmatrix},$$

$$\tilde{\mathbf{d}} = \left|\tilde{m} M \mathbf{d}\right|_{2d} = \begin{pmatrix} 74268 & 74268 & 74268 & 74268 \end{pmatrix}.$$

- *Précalculs (4.30) pour la base $\mathcal{B}$:*

$$\tilde{\mathbf{R}}_{19} = \begin{pmatrix} 16 & 10 & 16 & 17 \\ 17 & 8 & 0 & 7 \\ 18 & 8 & 11 & 16 \\ 4 & 16 & 16 & 17 \end{pmatrix}, \qquad \tilde{\mathbf{R}}_{23} = \begin{pmatrix} 10 & 7 & 9 & 11 \\ 13 & 6 & 3 & 4 \\ 3 & 2 & 14 & 17 \\ 0 & 3 & 14 & 1 \end{pmatrix}$$

$$\tilde{\mathbf{R}}_{29} = \begin{pmatrix} 5 & 10 & 12 & 19 \\ 13 & 19 & 27 & 26 \\ 6 & 2 & 17 & 23 \\ 4 & 0 & 3 & 8 \end{pmatrix}, \qquad \tilde{\mathbf{R}}_{31} = \begin{pmatrix} 19 & 6 & 10 & 1 \\ 2 & 21 & 5 & 1 \\ 1 & 19 & 17 & 2 \\ 13 & 3 & 12 & 28 \end{pmatrix},$$

$$\tilde{\mathbf{d}}_{19} = \begin{pmatrix} 17 & 17 & 17 & 17 \end{pmatrix}, \qquad \tilde{\mathbf{d}}_{23} = \begin{pmatrix} 13 & 13 & 13 & 13 \end{pmatrix},$$

$$\tilde{\mathbf{d}}_{29} = \begin{pmatrix} 18 & 18 & 18 & 18 \end{pmatrix}, \qquad \tilde{\mathbf{d}}_{31} = \begin{pmatrix} 13 & 13 & 13 & 13 \end{pmatrix}.$$

- *Précalculs (4.33) pour la base $\mathcal{B}'$:*

$$\tilde{\mathbf{R}}_{13} = \begin{pmatrix} 0 & 7 & 0 & 5 \\ 3 & 7 & 10 & 9 \\ 5 & 11 & 5 & 1 \\ 5 & 10 & 12 & 7 \end{pmatrix}, \qquad \tilde{\mathbf{R}}_{17} = \begin{pmatrix} 14 & 9 & 15 & 10 \\ 14 & 5 & 5 & 7 \\ 8 & 14 & 2 & 5 \\ 7 & 10 & 14 & 12 \end{pmatrix},$$

$$\tilde{\mathbf{R}}_{37} = \begin{pmatrix} 19 & 34 & 9 & 28 \\ 28 & 30 & 26 & 33 \\ 27 & 30 & 22 & 14 \\ 6 & 34 & 23 & 0 \end{pmatrix}, \qquad \tilde{\mathbf{R}}_{41} = \begin{pmatrix} 25 & 0 & 29 & 19 \\ 27 & 19 & 24 & 39 \\ 26 & 3 & 38 & 14 \\ 15 & 31 & 5 & 26 \end{pmatrix},$$

$$\tilde{\mathbf{d}}_{13} = \begin{pmatrix} 11 & 11 & 11 & 11 \end{pmatrix}, \qquad \tilde{\mathbf{d}}_{17} = \begin{pmatrix} 10 & 10 & 10 & 10 \end{pmatrix},$$

$$\tilde{\mathbf{d}}_{37} = \begin{pmatrix} 30 & 30 & 30 & 30 \end{pmatrix}, \qquad \tilde{\mathbf{d}}_{41} = \begin{pmatrix} 9 & 9 & 9 & 9 \end{pmatrix}.$$

$$\begin{array}{lllll} \tilde{\delta}_{13}^{(0)} = 1, & \tilde{\delta}_{13}^{(1)} = 11, & \tilde{\delta}_{13}^{(2)} = 4, & \tilde{\delta}_{13}^{(3)} = 9, & \tilde{\delta}_{13}^{(4)} = 8, \\ \tilde{\delta}_{17}^{(0)} = 13, & \tilde{\delta}_{17}^{(1)} = 15, & \tilde{\delta}_{17}^{(2)} = 5, & \tilde{\delta}_{17}^{(3)} = 11, & \tilde{\delta}_{13}^{(4)} = 7, \\ \tilde{\delta}_{37}^{(0)} = 5, & \tilde{\delta}_{37}^{(1)} = 10, & \tilde{\delta}_{37}^{(2)} = 34, & \tilde{\delta}_{37}^{(3)} = 4, & \tilde{\delta}_{37}^{(4)} = 30, \\ \tilde{\delta}_{41}^{(0)} = 18, & \tilde{\delta}_{41}^{(1)} = 29, & \tilde{\delta}_{41}^{(2)} = 40, & \tilde{\delta}_{41}^{(3)} = 19, & \tilde{\delta}_{41}^{(4)} = 31. \end{array}$$

- *Précalculs (4.39) pour $\{\tilde{m}\}$:*

$$\tilde{\mathbf{R}}_{11} = \begin{pmatrix} 2 & 9 & 1 & 10 \\ 2 & 9 & 1 & 1 \\ 5 & 3 & 1 & 2 \\ 4 & 8 & 10 & 3 \end{pmatrix}, \ \tilde{\mathbf{d}}_{11} = \begin{pmatrix} 3 & 3 & 3 & 3 \end{pmatrix},$$

$$\tilde{\delta}_{11}^{(1)} = 4, \ \tilde{\delta}_{11}^{(2)} = 10, \ \tilde{\delta}_{11}^{(3)} = 3, \ \tilde{\delta}_{11}^{(4)} = 6.$$

- *Précalculs (4.39) pour $\{m_\sigma\}$:*

$$\mathbf{R}_7 = \begin{pmatrix} 1 & 5 & 5 & 1 \\ 4 & 1 & 1 & 1 \\ 4 & 4 & 1 & 6 \\ 2 & 4 & 6 & 5 \end{pmatrix}, \ \tilde{\mathbf{R}}_7 = \begin{pmatrix} 6 & 4 & 1 & 3 \\ 0 & 6 & 3 & 2 \\ 1 & 0 & 2 & 3 \\ 6 & 6 & 3 & 6 \end{pmatrix}, \ \tilde{\mathbf{d}}_7 = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\tilde{\delta}_7^{(0)} = 2, \ \tilde{\delta}_7^{(1)} = 6, \ \tilde{\delta}_7^{(2)} = 1, \ \tilde{\delta}_7^{(3)} = 2, \ \tilde{\delta}_7^{(4)} = 3.$$

- *La première étape de l'Algorithme 22 est celle du calcul des vecteurs* $\mathbf{a}^{(m)} = \left(\mathbf{c}_m\tilde{\mathbf{R}}_m + \tilde{\mathbf{d}}_m\right) \bmod m$ *en parallèle pour tout* $m \in \mathcal{B} \cup \mathcal{B}' \cup \{\tilde{m}\}$, *et* $\mathbf{b}^{(m_\sigma)} = \left(\mathbf{c}_{m_\sigma}\tilde{\mathbf{R}}_{m_\sigma} + \tilde{\mathbf{d}}_{m_\sigma}\right) \bmod m_\sigma$, *via les précalculs précédents. Nous obtenons alors les résultats suivants :*

$$\mathcal{B} : \begin{cases} \mathbf{a}^{(19)} = \begin{pmatrix}13 & 3 & 8 & 7\end{pmatrix} \\ \mathbf{a}^{(23)} = \begin{pmatrix}13 & 22 & 16 & 2\end{pmatrix} \\ \mathbf{a}^{(29)} = \begin{pmatrix}7 & 12 & 25 & 12\end{pmatrix} \\ \mathbf{a}^{(31)} = \begin{pmatrix}14 & 26 & 13 & 16\end{pmatrix} \end{cases}, \mathcal{B}' : \begin{cases} \mathbf{a}^{(13)} = \begin{pmatrix}0 & 6 & 6 & 6\end{pmatrix} \\ \mathbf{a}^{(17)} = \begin{pmatrix}1 & 10 & 14 & 8\end{pmatrix} \\ \mathbf{a}^{(37)} = \begin{pmatrix}24 & 10 & 32 & 31\end{pmatrix} \\ \mathbf{a}^{(41)} = \begin{pmatrix}2 & 11 & 17 & 23\end{pmatrix}. \end{cases}$$

$$\tilde{m} : \mathbf{a}^{(11)} = \begin{pmatrix}1 & 10 & 10 & 1\end{pmatrix}$$

$$m_\sigma : \mathbf{b}^{(7)} = \begin{pmatrix}2 & 6 & 6 & 1\end{pmatrix}.$$

- *La seconde étape est la conversion de base (4.42). Le résultat obtenu, restocké dans les variables* $\mathbf{a}^{(m)}$ *pour* $m \in \mathcal{B}' \cup \{m_\sigma\}$ *et* $\tilde{\mathbf{q}}$, *est alors :*

$$\mathcal{B}' : \begin{cases} \mathbf{a}^{(13)} = \begin{pmatrix}6 & 1 & 6 & 2\end{pmatrix} \\ \mathbf{a}^{(17)} = \begin{pmatrix}11 & 3 & 2 & 10\end{pmatrix} \\ \mathbf{a}^{(37)} = \begin{pmatrix}8 & 25 & 36 & 31\end{pmatrix} \\ \mathbf{a}^{(41)} = \begin{pmatrix}31 & 3 & 4 & 5\end{pmatrix}. \end{cases}$$

$$\tilde{m} : \tilde{\mathbf{q}} = \begin{pmatrix}2 & 5 & 3 & 5\end{pmatrix}$$

$$m_\sigma : \mathbf{a}^{(7)} = \begin{pmatrix}0 & 2 & 6 & 4\end{pmatrix}.$$

- *La troisième étape correspond à la seconde conversion (4.43), modifiant les variables* $\mathbf{a}^{(m)}$ *pour* $m \in \mathcal{B}' \cup \{m_\sigma\}$ *:*

$$\mathcal{B}' : \begin{cases} \mathbf{a}^{(13)} = \begin{pmatrix}8 & 6 & 9 & 7\end{pmatrix} \\ \mathbf{a}^{(17)} = \begin{pmatrix}3 & 0 & 7 & 7\end{pmatrix} \\ \mathbf{a}^{(37)} = \begin{pmatrix}18 & 13 & 14 & 19\end{pmatrix} \\ \mathbf{a}^{(41)} = \begin{pmatrix}26 & 11 & 17 & 13\end{pmatrix}. \end{cases}$$

$$m_\sigma : \mathbf{a}^{(7)} = \begin{pmatrix}4 & 5 & 5 & 0\end{pmatrix}.$$

- *À l'issue de l'étape précédente, nous obtenons dans* $\mathcal{B}'$ *les résidus du vecteur* $|2\mathbf{c}\mathbf{R}' + \mathbf{d}|_{2d} + 2d\mathbf{v}_\mathbf{e}$ *(cf. (4.35)). La quatrième étape consiste en le calcul des coefficients MRS de ce vecteur afin d'effectuer une comparaison avec* $2d$ *et de pouvoir retrouver l'erreur* $\mathbf{v}_\mathbf{e}$, *pour une correction des résidus dans* $\{m_\sigma\}$. *Comme* $|2\mathbf{c}\mathbf{R}' + \mathbf{d}|_{2d} + p\mathbf{v}_\mathbf{e}$ *est calculé via la réduction modulo* $2d$ *de Montgomery de* $\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}$ *et que ce dernier vecteur possède uniquement des coefficients positifs, nous n'avons pas à considérer une comparaison avec un entier noté* t *(cf. (4.25) p. 143).*

  Pour ce faire, les coefficients MRS précalculés de $2d$ *dans la base MRS*

$$\{1, m'_1, m'_1m'_2, m'_1m'_2m'_3\}$$

  sont $(11, 1, 6, 18)$.

$$\begin{aligned} \text{MRScoeff}\left(\mathcal{B}', (8,3,18,26)\right) &= (8,14,24,6) < 2d \\ \text{MRScoeff}\left(\mathcal{B}', (6,0,13,11)\right) &= (6,10,12,12) < 2d \\ \text{MRScoeff}\left(\mathcal{B}', (9,7,14,17)\right) &= (9,9,1,9) < 2d \\ \text{MRScoeff}\left(\mathcal{B}', (7,7,19,13)\right) &= (7,0,25,10) < 2d \end{aligned}$$

Nous en déduisons alors que $\mathbf{v_e} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$.

- *La cinquième et dernière étape retourne le résultat attendu, soit :*

$$\left(\mathbf{b}^{(7)} + \mathbf{a}^{(7)}\mathbf{R}_7\right) \bmod 7 = \begin{pmatrix} 4 & 2 & 1 & 5 \end{pmatrix}.$$

En calculant les résidus centrés, nous avons

$$\begin{pmatrix} -3 & 2 & 1 & -2 \end{pmatrix} = \mathbf{p} = \mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}.$$

4.3 Technique d'accélération applicable à certaines bases

Le principal problème rencontré précédemment pour l'adaptation en RNS de la technique du round-off réside dans l'opération de réduction modulaire présente dans l'Équation 4.12 (p. 136). Afin de retrouver le vecteur d'erreur $\mathbf{v_e}$ de la Formule (4.14), les deux Algorithmes 20 et 21 précédents font appel une conversion dans le système positionnel MRS. Il en résulte l'Algorithme 22 qui implante une procédure de round-off en représentation hybride RNS-MRS.

L'objectif de cette section est de trouver une stratégie différente pour corriger le vecteur $\mathbf{v_e}$, et donc obtenir une réduction modulaire exacte, permettant de rester uniquement dans le système RNS.

4.3.1 Stratégie nouvelle pour un round-off RNS

Avec les précédentes techniques, nous avons choisi de supprimer l'erreur $\mathbf{v_e}$ en cherchant à calculer la réduction modulaire complète $(2\mathbf{c}\mathbf{R}' + \mathbf{d}) \bmod (2 \det \mathbf{R})$. La stratégie développée maintenant s'appuie quant à elle sur la forme de la formule complète 4.11 que nous rappelons ci-après, et non plus seulement sur le résultat de la réduction modulaire :

$$\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil = \frac{1}{2d} \times \left\{ 2\mathbf{c}\mathbf{R}' + \mathbf{d} - \left[\left(2\mathbf{c}\mathbf{R}' + \mathbf{d} \right) \bmod (2d) \right] \right\}. \tag{4.50}$$

Lorsque celle-ci est calculée via une réduction de Montgomery comme dans le cas RNS, alors nous rappelons que le résultat obtenu est le vecteur suivant :

$$\frac{1}{2d} \times \left\{ 2\mathbf{c}\mathbf{R}' + \mathbf{d} - \left[\left(2\mathbf{c}\mathbf{R}' + \mathbf{d} \right) \bmod (2d) + 2d \times \mathbf{v_e} \right] \right\} = \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil - \mathbf{v_e} \tag{4.51}$$

où $\mathbf{v_e} \in \{0,1\}^{\mathcal{N}}$.

La remarque suivante va constituer le fil directeur de la technique d'accélération développée par la suite.

Remarque 4.2 *S'il existe un entier γ tel que $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \bmod \gamma = 0$ pour tout vecteur $\mathbf{c} \in \mathcal{C}$ et $\mathbf{v_e} \bmod \gamma \neq 0$ pour tout vecteur $\mathbf{v_e}$ non nul pouvant apparaître dans le calcul de la Formule (4.51), alors $\mathbf{v_e}$ peut être retrouvé, corrigé, et par suite il est aisé de calculer entièrement en RNS la quantité $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \bmod m_\sigma$ via l'Équation (4.50) transposée dans $\mathbb{Z}/m_\sigma\mathbb{Z}$.*

Sous l'hypothèse de l'existence d'un tel entier γ, alors en calculant le membre de gauche de l'Équation (4.51) dans la base RNS $\{m_\sigma, \gamma\}$, le résidu modulo γ permettrait la détection des coefficients non nuls de $\mathbf{v_e}$, d'où une correction immédiate sur le résidu modulo m_σ.

Obtenir un multiple de γ

Étant donné qu'il n'y a *a priori* aucune raison justifiant l'existence d'un entier γ tel que décrit par la Remarque 4.2, il va s'agir de modifier les vecteurs $\mathbf{c}$ pour forcer l'apparition d'une telle propriété. Pour ce faire, nous fixons un entier $\gamma > 0$, et nous analysons la forme du résultat du calcul de $\lfloor \gamma \mathbf{c} \mathbf{R}^{-1} \rceil$.

D'une part, lorsque $\lfloor \gamma \mathbf{c} \mathbf{R}^{-1} \rceil$ est calculé via la Formule (4.50) couplée avec une réduction modulaire de Montgomery, le résultat obtenu est toujours affecté par un vecteur d'erreur semblable à celui présent dans l'Équation (4.51). En notant toujours ce vecteur $\mathbf{v_e}$, alors nous obtenons pour résultat $\lfloor \gamma \mathbf{c} \mathbf{R}^{-1} \rceil - \mathbf{v_e}$, avec $\mathbf{v_e} \in \{0,1\}^{\mathcal{N}}$. D'autre part, en substituant $\mathbf{c}$ par $\mathbf{p} + \mathbf{k}\mathbf{B}$, il vient alors :

$$\begin{aligned} \lfloor \gamma \mathbf{c} \mathbf{R}^{-1} \rceil &= \lfloor \gamma \mathbf{p} \mathbf{R}^{-1} + \gamma \mathbf{k} \mathbf{B} \mathbf{R}^{-\mathbf{1}} \rceil \\ &= \lfloor \gamma \mathbf{p} \mathbf{R}^{-1} + \gamma \mathbf{k} \mathbf{U} \rceil \\ &= \lfloor \gamma \mathbf{p} \mathbf{R}^{-1} \rceil + \gamma \mathbf{k} \mathbf{U} \\ &= \lfloor \gamma \mathbf{p} \mathbf{R}^{-1} \rceil + \gamma \lfloor \mathbf{c} \mathbf{R}^{-1} \rceil. \end{aligned} \tag{4.52}$$

Par conséquent, le calcul de $\lfloor \gamma \mathbf{c} \mathbf{R}^{-1} \rceil$ via la Formule (4.50) nous donne le résultat suivant :

$$\lfloor \gamma \mathbf{c} \mathbf{R}^{-1} \rceil - \mathbf{v_e} = \gamma \lfloor \mathbf{c} \mathbf{R}^{-1} \rceil + \lfloor \gamma \mathbf{p} \mathbf{R}^{-1} \rceil - \mathbf{v_e}. \tag{4.53}$$

Technique de correction

Étant donné l'Équation (4.53), la stratégie de détection suggérée par la Remarque 4.2 consiste à calculer :

$$\left(\lfloor \gamma \mathbf{c} \mathbf{R}^{-1} \rceil - \mathbf{v_e} \right) \bmod \gamma = \left(\lfloor \gamma \mathbf{p} \mathbf{R}^{-1} \rceil - \mathbf{v_e} \right) \bmod \gamma. \tag{4.54}$$

Vu que notre objectif est, outre la détection de l'erreur, sa correction, il nous est nécessaire de trouver des conditions sur γ et $\mathbf{R}$ permettant de retrouver aisément le vecteur entier $\lfloor \gamma \mathbf{p} \mathbf{R}^{-1} \rceil - \mathbf{v_e}$ depuis son résidu modulo γ (4.54). Une condition suffisante pour rendre ceci possible serait que les coefficients $\lfloor \gamma \mathbf{p} \mathbf{R}^{-1} \rceil - \mathbf{v_e}$ prennent leur valeur dans un intervalle de longueur au plus γ. Le lemme suivant définit une fonction qui permet, dans un contexte précis, de récupérer une valeur connaissant son résidu modulo γ.

Lemme 4.1 *Soit $\mathcal{I}$ un intervalle contenant 0 et γ un entier vérifiant $\gamma \geqslant \operatorname{card}(\mathcal{I})$. Soit de plus un entier $a \in [\![0, \gamma - 1]\!]$ tel que $\mathcal{I} \subseteq [\![-a, \gamma - a - 1]\!]$, et $f_{\gamma,a}$ la fonction suivante :*

$$\begin{array}{rccl} f_{\gamma,a} & : & \{|x|_\gamma \mid x \in \mathcal{I}\} & \to & \mathcal{I} \\ & & z & \mapsto & \begin{cases} z - \gamma \text{ si } z \geqslant |-a|_\gamma, \\ z \text{ sinon.} \end{cases} \end{array} \tag{4.55}$$

Alors $f_{\gamma,a}$ est bijective et pour tout entier $x \in \mathcal{I}$, $f_{\gamma,a}\left(|x|_\gamma\right) = x$.

Démonstration. Soit $\mathcal{I}_+ = \mathcal{I} \cap [\![0, \gamma - a - 1]\!]$, et $\mathcal{I}_- = \mathcal{I} \cap [\![-a, -1]\!]$. Par hypothèse sur a, $\mathcal{I}_+ \bmod \gamma = \mathcal{I}_+ \subseteq [\![0, \gamma - a - 1]\!] = [\![0, |a|_\gamma - 1]\!]$ et $\mathcal{I}_- \bmod \gamma = \gamma + \mathcal{I}_- \subseteq [\![\gamma - a, \gamma - 1]\!] = [\![|a|_\gamma, \gamma - 1]\!]$. Par conséquent l'ensemble $\{|x|_\gamma \mid x \in \mathcal{I}\}$ est l'union disjointe $\mathcal{I}_+ \sqcup \mathcal{I}_-$. Ainsi, si $x \in \mathcal{I}_+$ alors $|x|_\gamma = x$ et $|x|_\gamma < |-a|_\gamma$.

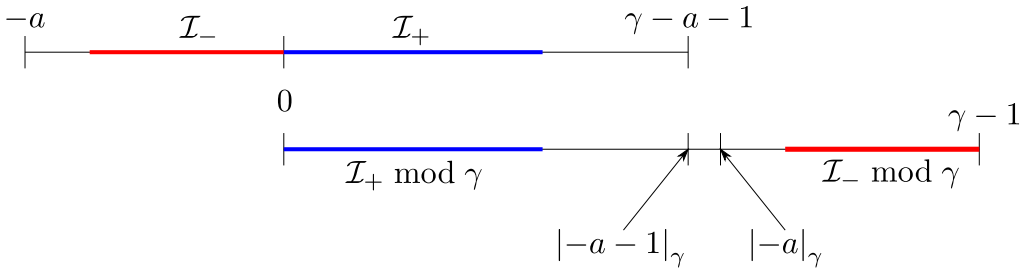


FIGURE 4.7 – *Illustration du Lemme 4.1.*

Donc $x = |x|_\gamma = f_{\gamma,a}\left(|x|_\gamma\right)$. Et si $x \in \mathcal{I}_-$, alors $|x|_\gamma = x + \gamma$ et $|x|_\gamma \geqslant |-a|_\gamma$. Et par suite $x = |x|_\gamma - \gamma = f_{\gamma,a}\left(|x|_\gamma\right)$.

□

Pour un intervalle $\mathcal{I}$ donné, si le paramètre γ est choisi assez grand pour pouvoir avoir $a = \lfloor \frac{\gamma - 1}{2} \rfloor$, alors la fonction $f_{\gamma,a}$ coïncide avec l'opérateur de reste modulaire centré modc.

Remarque 4.3 *Dans le contexte du Lemme 4.1, un choix judicieux du paramètre a en fonction de γ peut faciliter l'opération de comparaison. Par exemple, lorsque γ est de la forme $2^{\alpha+1}+1$, alors si $a = 2^\alpha + 2$, la comparaison avec $|-a|_\gamma = |-2^\alpha - 2|_{2^{\alpha+1}+1} = \left|2^{\alpha+1} + 1 - 2^\alpha - 2\right|_{2^{\alpha+1}+1} = 2^\alpha - 1$ s'effectue simplement par la vérification du bit de poids fort.*

4.3.2 Recherche de paramètre et bases concernées

Recherche d'un entier γ adéquat

L'entier γ recherché doit satisfaire les hypothèses du Lemme 4.1. Dans notre contexte, les paramètres a et γ de ce même lemme sont contraints par les inégalités suivantes, pour tout $i \in [\![1, \mathcal{N}]\!]$:

$$-a \leqslant \lfloor \gamma \left(\mathbf{p}\mathbf{R}^{-1}\right)_i \rceil - \mathbf{v}_{\mathbf{e},i} \leqslant \gamma - a - 1. \tag{4.56}$$

Si la Condition (4.56) est vérifiée, alors le vecteur $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil$ peut être retrouvé par l'équation suivante :

$$\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil = \frac{1}{\gamma}\left\{\lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil - \mathbf{v}_\mathbf{e} - f_{\gamma,a}\left(\lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil - \mathbf{v}_\mathbf{e}\right)\right\}. \tag{4.57}$$

Le théorème suivant reprend ces propos et exhibe des conditions suffisantes sur γ et a garantissant l'obtention des inégalités (4.56) pour tout $\mathbf{c} \in C_\mathcal{R}(\mathcal{P}_\sigma)$. Ces conditions découlent de propriétés géométriques du réseau $\mathcal{R}$ et plus précisément de la base secrète $\mathbf{R}$ sous-jacente.

Théorème 4.2 *Soit un paramètre entier σ, $(\mathbf{R}, \mathbf{B}) \in \left(\mathbb{Z}^{\mathcal{N}\times\mathcal{N}}\right)^2$ deux bases d'un réseau $\mathcal{R}$ telles que $\mathbf{B}\mathbf{R}^{-1} = \mathbf{U} \in GL_\mathcal{N}(\mathbb{Z})$, $\rho_{\mathbf{R}^{-1}} = \left\|\mathbf{R}^{-1}\right\|_\infty$, et la base $\mathbf{R}$ est telle que $\sigma\rho_{\mathbf{R}^{-1}} < \frac{1}{2}$. Soit le réel strictement positif $\epsilon_\mathbf{R} = \frac{1}{2} - \sigma\rho_\mathbf{R} = \frac{1-2\sigma\rho_\mathbf{R}}{2}$, et E_1 et E_2 des entiers positifs.*

Pour tout entier γ vérifiant la condition suivante :

$$\gamma \geqslant E_2 + E_1 + 1 + \lfloor \gamma\sigma\rho_\mathbf{R} \rceil - \lfloor -\gamma\sigma\rho_\mathbf{R} \rceil \tag{4.58}$$

alors il existe un entier a tel que :

$$E_1 - \lfloor -\gamma\sigma\rho_{\mathbf{R}} \rceil \leqslant a \leqslant \gamma - 1 - E_2 - \lfloor \gamma\sigma\rho_{\mathbf{R}} \rceil \tag{4.59}$$

et tel que pour tout $\mathbf{v_e} \in [-E_1, E_2]^{\mathcal{N}}$ *et tout* $\mathbf{c} \in C_{\mathcal{R}}(\mathcal{P}_\sigma)$, $\lfloor \mathbf{cR}^{-1} \rceil$ *se déduit de* $\lfloor \gamma\mathbf{cR}^{-1} + \mathbf{v_e} \rceil$ *grâce à l'égalité suivante :*

$$\gamma \lfloor \mathbf{cR}^{-1} \rceil = \lfloor \gamma\mathbf{cR}^{-1} + \mathbf{v_e} \rceil - f_{\gamma,a}\left(\lfloor \gamma\mathbf{cR}^{-1} + \mathbf{v_e} \rceil \bmod \gamma \right). \tag{4.60}$$

De plus, une condition suffisante pour que γ *satisfasse (4.58) est donnée par :*

$$\gamma\epsilon_{\mathbf{R}} \geqslant \frac{E_2 + E_1 + 2}{2} \Leftrightarrow \gamma \geqslant \lceil \frac{E_2 + E_1 + 2}{2\epsilon_{\mathbf{R}}} \rceil. \tag{4.61}$$

Démonstration. Soit $\mathbf{v_e} \in [-E_1, E_2]^{\mathcal{N}}$, $\mathbf{p} \in \mathcal{P}_\sigma$, $\mathbf{k} \in \mathbb{Z}^{\mathcal{N}}$ et $\mathbf{c} = \mathbf{p} + \mathbf{kB}$. Ainsi, $\gamma\mathbf{cR}^{-1} + \mathbf{v_e} = \gamma\mathbf{kU} + \gamma\mathbf{pR}^{-1} + \mathbf{v_e}$ et $\gamma\mathbf{kU}$ est un vecteur entier. Par suite, nous avons $\lfloor \gamma\mathbf{cR}^{-1} + \mathbf{v_e} \rceil \bmod \gamma = \lfloor \gamma\mathbf{pR}^{-1} + \mathbf{v_e} \rceil \bmod \gamma$. Ensuite, par hypothèse, $\mathbf{pR}^{-1} \in [-\frac{1}{2} + \epsilon_{\mathbf{R}}, \frac{1}{2} - \epsilon_{\mathbf{R}}]^{\ell}$. En conséquence, tout i-ième coefficient du vecteur $\gamma\mathbf{pR}^{-1} + \mathbf{v_e}$ vérifie les inégalités suivantes :

$$\alpha_1 = -\frac{\gamma}{2} + \gamma\epsilon_{\mathbf{R}} - E_1 \leqslant \left(\gamma\mathbf{pR}^{-1} + \mathbf{v_e} \right)_i \leqslant \frac{\gamma}{2} - \gamma\epsilon_{\mathbf{R}} + E_2 = \alpha_2.$$

Si γ est un entier vérifiant :

$$\gamma \geqslant \lfloor \alpha_2 \rceil - \lfloor \alpha_1 \rceil + 1 = E_2 + E_1 + 1 + \lfloor \gamma\sigma\rho_{\mathbf{R}} \rceil - \lfloor -\gamma\sigma\rho_{\mathbf{R}} \rceil,$$

alors il existe un entier a (*e.g.* $a = -\lfloor \alpha_1 \rceil$) vérifiant :

$$\begin{cases} -a \leqslant \lfloor \alpha_1 \rceil \\ \gamma - a - 1 \geqslant \lfloor \alpha_2 \rceil \end{cases} \Leftrightarrow E_1 - \lfloor -\gamma\sigma\rho_{\mathbf{R}} \rceil \leqslant a \leqslant \gamma - 1 - E_2 - \lfloor \gamma\sigma\rho_{\mathbf{R}} \rceil.$$

Par conséquent, par le Lemme 4.1 nous obtenons finalement dans ce cas que pour tout $\mathbf{c} \in C_{\mathcal{R}}(\mathcal{P}_\sigma)$ et tout $\mathbf{v_e} \in [-E_1, E_2]^{\mathcal{N}}$:

$$\begin{aligned} f_{\gamma,a}\left(\lfloor \gamma\mathbf{cR}^{-1} + \mathbf{v_e} \rceil \bmod \gamma \right) &= f_{\gamma,a}\left(\left\{ \gamma\mathbf{kU} + \lfloor \gamma\mathbf{pR}^{-1} + \mathbf{v_e} \rceil \right\} \bmod \gamma \right) \\ &= f_{\gamma,a}\left(\lfloor \gamma\mathbf{cR}^{-1} + \mathbf{v_e} \rceil \bmod \gamma \right) \\ &= \lfloor \gamma\mathbf{cR}^{-1} + \mathbf{v_e} \rceil. \end{aligned} \tag{4.62}$$

Comme $x - \frac{1}{2} \leqslant \lfloor x \rceil < x + \frac{1}{2}$ pour tout réel x, alors une condition suffisante sur γ est donnée par :

$$\begin{aligned} &\gamma \geqslant \alpha_2 - \alpha_1 + 2 = \gamma - 2\gamma\epsilon_{\mathbf{R}} + E_2 + E_1 + 2 \\ &\Leftrightarrow \gamma\epsilon_{\mathbf{R}} \geqslant \frac{E_2 + E_1 + 2}{2} \\ &\Leftrightarrow \gamma \geqslant \lceil \frac{E_2 + E_1 + 2}{2\epsilon_{\mathbf{R}}} \rceil. \end{aligned}$$

Ceci conclut la preuve. □

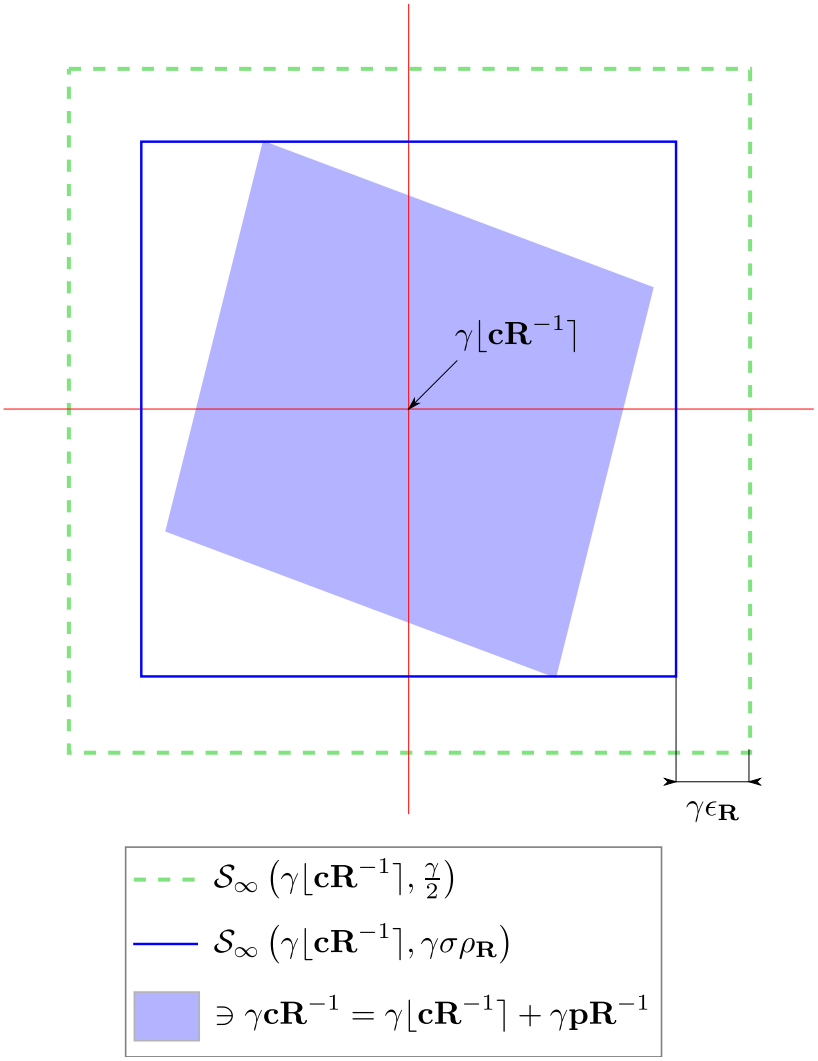


FIGURE 4.8 – *Dilatation par γ des vecteurs $\mathbf{cR}^{-1} = \mathbf{pR}^{-1} + \lfloor \mathbf{cR}^{-1} \rceil$, pour $\mathbf{p} \in \mathcal{P}_\sigma$.*

La Figure 4.8 illustre l'effet de la dilatation par γ du vecteur $\mathbf{cR}^{-1} = \mathbf{pR}^{-1} + \lfloor \mathbf{cR}^{-1} \rceil$. Le vecteur résultat de cette dilatation, $\gamma\mathbf{cR}^{-1} = \gamma\lfloor \mathbf{cR}^{-1} \rceil + \gamma\mathbf{pR}^{-1}$, appartient au parallélogramme bleu.

La Figure 4.9 montre l'effet de l'addition d'un vecteur d'erreur $\mathbf{v_e} \in [-E_1, E_2]^2$. Lorsque l'ensemble des vecteurs possibles $\gamma\mathbf{cR}^{-1} + \mathbf{v_e}$ peut être contenu dans un parallélogramme $[-a, \gamma - a - 1]^2$ avec a entier, alors l'arrondi de chaque coefficient $\left(\gamma\mathbf{cR}^{-1} + \mathbf{v_e}\right)_i$ est dans l'intervalle $[\![-a, \gamma - a - 1]\!]$, et peut donc être déduit de son résidu modulo γ.

Corollaire 4.1 *Soit les hypothèses du Théorème 4.2, et $n \geqslant 1$ un entier. Soit γ un entier premier avec $2d$ et vérifiant la condition suivante :*

$$\gamma \geqslant n + 1 + \lfloor \gamma\sigma\rho_{\mathbf{R}} \rceil - \lfloor -\gamma\sigma\rho_{\mathbf{R}} \rceil. \tag{4.63}$$

Alors il existe un entier a tel que pour tout $\mathbf{c} \in \mathrm{C}_{\mathcal{R}}(\mathcal{P}_\sigma)$ et tout $\mathbf{v_e} \in [\![0, n]\!]^{\mathcal{N}}$, si $\mathbf{r}$ désigne le vecteur $(2\gamma\mathbf{cR}' + \mathbf{d}) \bmod (2d) + 2d \times \mathbf{v_e}$, l'égalité suivante est vérifiée :

$$\lfloor \mathbf{cR}^{-1} \rceil = \frac{1}{\gamma} \times \left[\frac{2\gamma\mathbf{cR}' + \mathbf{d} - \mathbf{r}}{2d} - f_{\gamma,a}\left(\frac{2\gamma\mathbf{cR}' + \mathbf{d} - \mathbf{r}}{2d} \bmod \gamma \right) \right]. \tag{4.64}$$

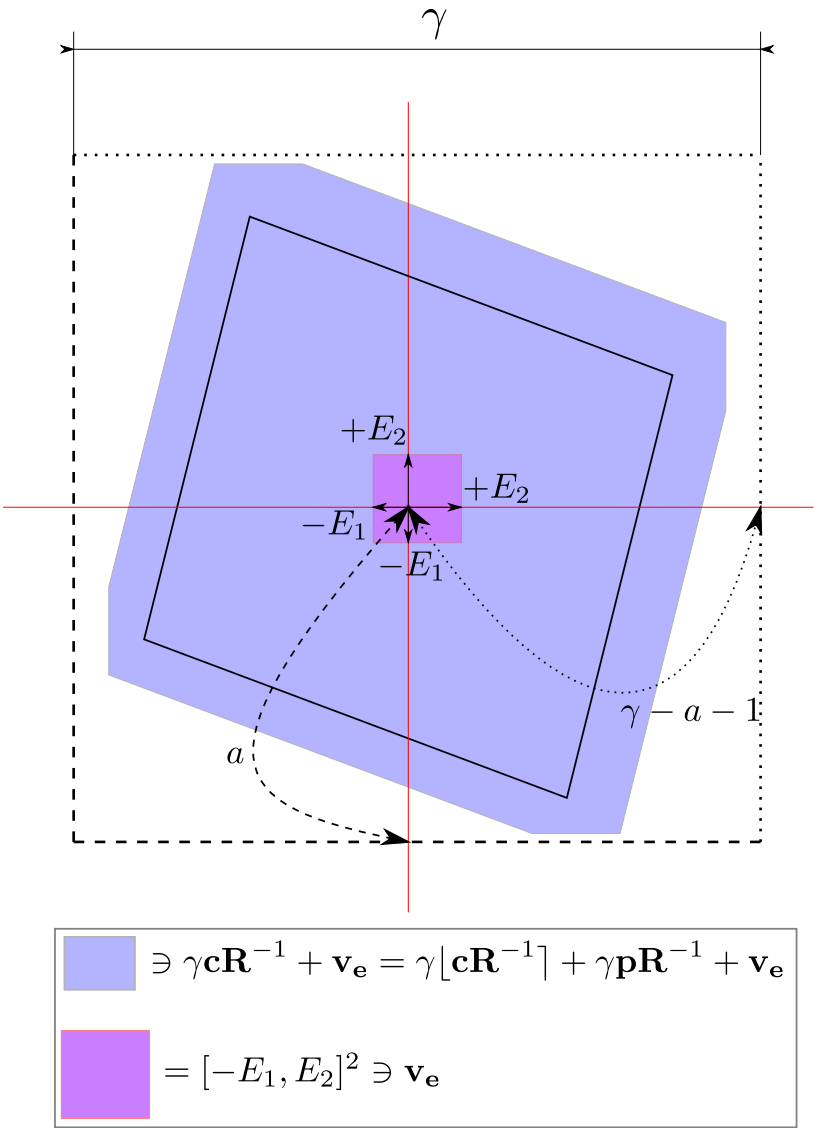


FIGURE 4.9 – *Effet des translations des vecteurs* $\mathbf{c}\mathbf{R}^{-1}$ *par les vecteurs d'erreurs possibles* $\mathbf{v_e} \in [-E_1, E_2]^2$.

Enfin, soit l'entier $\gamma_{\mathbf{R},n} = \lceil \frac{n+2}{2\epsilon_{\mathbf{R}}} \rceil$. *Une condition suffisante pour l'existence de a est également donnée par :*

$$\gamma \geqslant \gamma_{\mathbf{R},n}. \tag{4.65}$$

De plus, un tel entier a peut prendre sa valeur dans l'ensemble suivant :

$$a \in [\![n - \lfloor -\gamma\sigma\rho_{\mathbf{R}} \rceil, \gamma - 1 - \lfloor \gamma\sigma\rho_{\mathbf{R}} \rceil]\!]. \tag{4.66}$$

Démonstration. Les Conditions (4.63) et (4.65) ne sont autres que respectivement les Condition (4.58) et (4.61) du Théorème 4.2 avec $E_1 = n$ et $E_2 = 0$.

De par l'Équation (4.50) (p. 160),

$$\frac{2\gamma\mathbf{c}\mathbf{R}' + \mathbf{d} - \mathbf{r}}{2d} = \lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil - \mathbf{v_e}. \tag{4.67}$$

Étant donné que $\mathbf{v_e}$ est un vecteur entier, le Théorème 4.2 permet d'établir, sous la condition $\gamma \geqslant \gamma_{\mathbf{R},n}$, l'existence de a tel que :

$$\lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil - \mathbf{v_e} = \gamma \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil + f_{\gamma,a}\left(\left(\lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil - \mathbf{v_e} \right) \bmod \gamma \right). \tag{4.68}$$

L'Équation (4.64) découle de la combinaison des Équations (4.67) et (4.68).

Enfin, (4.66) vient directement des inégalités (4.59). □

Application au contexte présent

Nous rappelons que l'objectif est de calculer $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \bmod m_\sigma$ via la Formule (4.12), et rappelée ci-après :

$$\left| \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \right|_{m_\sigma} = \left| |2 \det \mathbf{R}|_{m_\sigma}^{-1} \times \left\{ \left| 2\mathbf{c}\mathbf{R}' + \mathbf{d} \right|_{m_\sigma} - \left| \left(2\mathbf{c}\mathbf{R}' + \mathbf{d}\right) \bmod (2 \det \mathbf{R}) \right|_{m_\sigma} \right\} \right|_{m_\sigma}.$$

Contrairement aux techniques précédentes utilisant une conversion en MRS pour calculer exactement $[(2\mathbf{c}\mathbf{R}' + \mathbf{d}) \bmod (2d)] \bmod m_\sigma$, la nouvelle approche se fonde sur le calcul en RNS, via la Formule (4.64) du corollaire précédent, de la quantité $\lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil$ dans la base $\{m_\sigma, \gamma\}$. Pour des raisons d'efficacité, le calcul du vecteur suivant :

$$\mathbf{r} = \left(2\mathbf{c}\mathbf{R}' + \mathbf{d}\right) \bmod (2d)$$

peut s'effectuer via l'algorithme de réduction modulaire RNS avec pour seule opération de conversion de base $\mathrm{Bex}_1 = \mathrm{Bex}_{crt}$. Ceci a pour conséquence d'augmenter la taille possible des coefficients du vecteur d'erreur $\mathbf{v_e}$, celui-ci appartenant alors à $[\![0, n]\!]^{\mathcal{N}}$. Nous modifions donc les précalculs donnant initialement les représentations de Montgomery pour y intégrer la multiplication par γ :

$$\begin{cases} \tilde{\mathbf{R}} = 2\gamma M \times \mathbf{R}' \bmod (2d), \\ \tilde{\mathbf{d}} = M \times \mathbf{d} \bmod (2d). \end{cases} \tag{4.69}$$

Alors le calcul de la réduction modulaire donne un résultat de la forme :

$$\mathrm{RedModRNS}\left(\mathcal{B}, \{m_\sigma, \gamma\}, \mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}, 2d\right) = \left(|\mathbf{r} + \mathbf{v_e}|_{m_\sigma}, |\mathbf{r} + \mathbf{v_e}|_{\gamma} \right),$$

où $M = M^{\mathcal{B}}$, $n = |\mathcal{B}|$ et $\mathbf{v_e} \in [\![0, n]\!]^{\mathcal{N}}$ est un vecteur d'erreur dû à l'inexactitude de la conversion Bex_{crt}.

Par suite, il est aisé d'obtenir les résidus dans la base $\{m_\sigma, \gamma\}$ du quotient $\frac{2\gamma\mathbf{c}\mathbf{R}' + \mathbf{d} - \mathbf{r}}{2d}$. Ainsi, lorsque γ vérifie la condition $\gamma \geqslant \gamma_{\mathbf{R},n}$ du Corollaire 4.1, la correction du vecteur $\mathbf{v_e}$ dans le canal m_σ intervient après le calcul dans le canal $\mathbb{Z}/\gamma\mathbb{Z}$ de la quantité suivante :

$$f_{\gamma,a}\left(\frac{2\gamma\mathbf{c}\mathbf{R}' + \mathbf{d} - \mathbf{r}}{2d} \bmod \gamma \right),$$

où a est un paramètre vérifiant les Conditions (4.59) du Corollaire 4.1 (p. 164). La Formule (4.64) de ce même corollaire peut alors être calculée dans le canal $\mathbb{Z}/m_\sigma\mathbb{Z}$, ce qui donne le résultat escompté. Le schéma général de l'approche est résumé par la Figure 4.10.

Bases concernées

Pour toute base $\mathbf{R}$ vérifiant la condition de Babai, la technique est toujours applicable. Il suffit pour cela de choisir $\gamma \geqslant \gamma_{\mathbf{R},n}$. Il est clair que plus $\sigma\rho_{\mathbf{R}}$ sera proche de $\frac{1}{2}$, plus les entiers γ utilisables seront grands. L'efficacité de cette

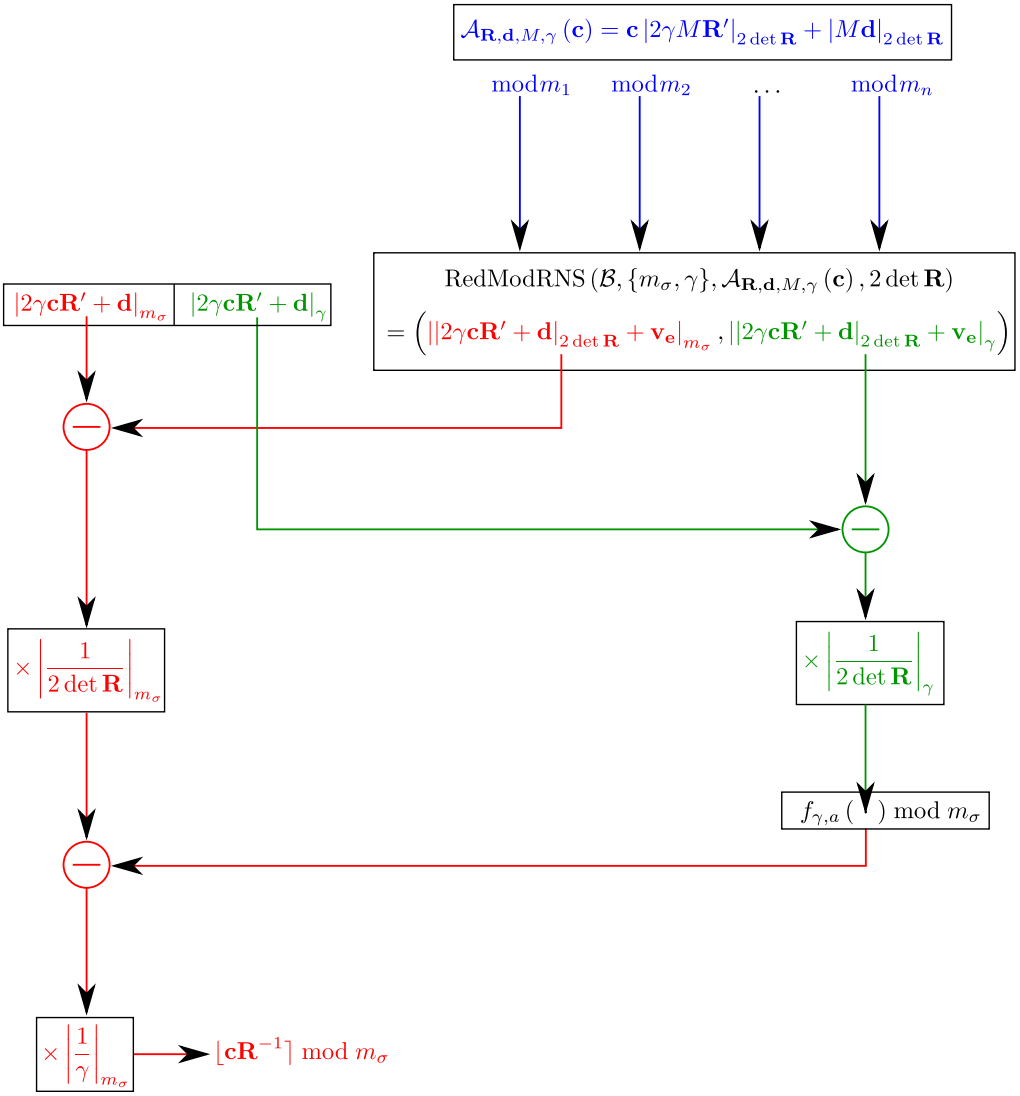


FIGURE 4.10 – *Principe de l'approche proposée pour un round-off RNS (les calculs dans $\mathcal{B}$ (resp. $\{m_\sigma\}$ et $\{\gamma\}$) sont colorés en bleu (resp. rouge et vert)).*

approche est optimale lorsque la taille de γ ne surpasse pas la taille β des moduli de $\mathcal{B}$. En effet, lorsque les calculs dans $\mathbb{Z}/\gamma\mathbb{Z}$ peuvent s'effectuer dans un unique canal, la comparaison nécessaire pour l'évaluation de la fonction $f_{\gamma,a}$ ne requiert alors pas de passer par le MRS. Ainsi, la technique est en particulier optimisée pour les réseaux vérifiant $\gamma_{\mathbf{R},n} \leqslant \beta - 1$, c'est-à-dire vérifiant la condition suivante :

$$0 < \sigma\rho_{\mathbf{R}} \leqslant \frac{1}{2} - \frac{n+2}{2(\beta-1)}. \tag{4.70}$$

Dans le pire cas, γ est de l'ordre du déterminant d de $\mathbf{R}$. En effet, en remarquant que, pour une base $\mathbf{R}$ fixée vérifiant la condition de Babai, $d\rho_{\mathbf{R}}$ est un entier, alors nous pouvons écrire :

$$\epsilon_{\mathbf{R}} = \frac{1}{2} - \sigma\rho_{\mathbf{R}} = \frac{d - d\sigma\rho_{\mathbf{R}}}{2d} \geqslant \frac{1}{2d}.$$

Ainsi, $\gamma_{\mathbf{R},n} = \lceil \frac{n+2}{2\epsilon_{\mathbf{R}}} \rceil \leqslant (n+2)\,d$.

Ce cas extrême correspond à l'Algorithme 22 (p. 154) du CVP en représentation hybride RNS-MRS dépourvu de la seconde réduction via le modulus $\tilde{m}$, et où γ est remplacé par la base RNS $\mathcal{B}'$ dont l'intervalle dynamique doit alors également être de la taille de $(n+2)\,d$.

Il est possible de conserver un γ petit, et ce quelle que soit la base $\mathbf{R}$, sous la condition de réduire l'espace d'états des messages $\mathcal{P}_\sigma$ à $[\![-\sigma+1,\sigma]\!]$ ou $[\![-\sigma,\sigma-1]\!]$ par exemple. Dans ce cas, la Condition (4.63) du Corollaire 4.1 devient :

$$\gamma \geqslant n+1+\lfloor \gamma\,(\sigma-1)\,\rho_{\mathbf{R}} \rceil - \lfloor -\gamma\sigma\rho_{\mathbf{R}} \rceil.$$

Une condition suffisante est alors donnée par :

$$\gamma \geqslant n+2+2\gamma\sigma\rho_{\mathbf{R}} - \gamma\sigma\rho_{\mathbf{R}} \Leftrightarrow \gamma \geqslant \left\lceil \frac{n+2}{1-2\sigma\rho_{\mathbf{R}}+\rho_{\mathbf{R}}} \right\rceil = \left\lceil \frac{n+2}{\epsilon_{\mathbf{R}}+\rho_{\mathbf{R}}} \right\rceil.$$

Or, comme $0 < \epsilon_{\mathbf{R}} < \frac{1}{2}$ et $\rho_{\mathbf{R}} = \frac{1-2\epsilon_{\mathbf{R}}}{2\sigma}$, nous avons alors $\frac{1}{2\sigma} < \epsilon_{\mathbf{R}}+\rho_{\mathbf{R}} < \frac{1}{2}$. Par conséquent, dans ce cas :

$$\gamma_{\mathbf{R},n} = \left\lceil \frac{n+2}{\epsilon_{\mathbf{R}}+\rho_{\mathbf{R}}} \right\rceil \leqslant 2\sigma\,(n+2)\,.$$

Plus généralement, le round-off renvoie le plus proche vecteur lorsqu'il est appliqué à l'ensemble suivant :

$$\mathcal{R} + \left[B_\infty\left(0, \frac{1}{2\rho_{\mathbf{R}}}\right)\mathbf{R}^{-1} \cap \mathbb{Z}^{\mathcal{N}} \right].$$

À γ donné, la technique d'accélération propose un arrondi exact pour un ensemble restreint à :

$$\mathcal{R} + \left[B_\infty\left(0, \frac{1}{2\rho_{\mathbf{R}}}\left(1-\frac{n}{\gamma}\right)\right)\mathbf{R}^{-1} \cap \mathbb{Z}^{\mathcal{N}} \right].$$

Dans le cas plus général d'une base $\mathbf{R}$ résultant d'une réduction LLL d'une matrice tirée uniformément dans $[\![-\mathcal{N},\mathcal{N}]\!]^{\mathcal{N}^2}$, des expérimentations sur un échantillon de 100 matrices et ce pour plusieurs dimensions montrent que la taille espérée des γ obtenus semble la plupart du temps rester inférieure à 2^{12}, et donc inférieurs aux tailles de moduli utilisées en pratique (cf. Table 4.8).

$\mathcal{N}$	128	256	384	512	640	768
σ	3	3	3	3	3	2
max	2003	4775	3637	4059	683	334
min	10	6	6	6	5	4
médiane	66.5	37.5	31	28.5	21	9
écart moyen	193.7	260.9	162.6	193.6	41.6	13.2

TABLE 4.8 – *Quelques statistiques sur des valeurs $\gamma_{\mathbf{R},1}$ pour 100 matrices aléatoires de $[\![-\mathcal{N},\mathcal{N}]\!]^{\mathcal{N}^2}$ LLL-réduites pour différentes dimensions.*

4.3.3 Un algorithme entièrement RNS pour la résolution du CVP

Dans cette partie, la méthode précédente est intégrée dans un algorithme RNS résolvant le problème du plus proche vecteur par le round-off. Autrement dit, l'algorithme présenté par la suite retourne le même résultat que

l'Algorithme 22. Afin de fournir un algorithme optimisé, les précalculs possibles sont détaillés. Une analyse donne une estimation de la taille totale de ces précalculs. Ensuite, l'algorithme principal est détaillé, et sa complexité est étudiée.

Dans la suite, δ peut encore désigner l'entier $2d$.

Du binaire au RNS

Dans l'optique de proposer un algorithme RNS optimisé et utilisable dans un contexte global où les données sont représentées dans un système de numération positionnel standard binaire, il convient de préciser l'impact que peut avoir le changement de représentation vers le RNS du vecteur $\mathbf{c}$ sur le coût du total de l'algorithme pour le CVP.

Le contexte est celui d'une base RNS $\mathcal{B} = \{m_1, \ldots, m_n\}$ ayant pour contrainte une taille de moduli bornée par $\beta = 2^r$, et vérifiant de plus $M > \mathcal{N}c_\infty + 1$. Ainsi, n est estimé être peu ou prou par $\lceil \log_\beta (\mathcal{N}c_\infty + 1) \rceil$. Par la suite, k dénote $\lceil \log_\beta (c_\infty) \rceil$.

Les différentes approches dans l'état-de-l'art se basent sur un compromis temps/mémoire. Les principes généraux sont les suivants.

La méthode la plus directe consiste à utiliser l'écriture binaire d'un entier $x = \sum_{i=1}^{t} x_i 2^i$ qui doit être réduit modulo m. Pour ce faire, il suffit de se doter d'une table mémoire contenant les t valeurs $\left|2^i\right|_m$. La réduction nécessite alors $t-1$ additions modulaires élémentaires, puisque $|c|_m = \left(\sum_{i=1}^{t} x_i \times \left|2^i\right|_m\right) \bmod m$. Cependant, il est à noter qu'une périodicité des valeurs $\left|2^i\right|_m$ liée à l'ordre de 2 dans le groupe multiplicatif $\mathbb{Z}/m\mathbb{Z}^\times$ peut réduire significativement les besoins en mémoire (Premkumar et al. 2006). Appliquant ceci au vecteur $\mathbf{c}$ avec la base RNS $\mathcal{B}$, le coût total est au plus de $nkr\mathcal{N}$ `AMEl`. Mais il est nécessaire de conserver en mémoire nkr β-mots, soit nkr^2 bits.

Une autre approche par blocs se base sur la décomposition des coefficients de $\mathbf{c}$ dans la base β. Pour $i \in [\![1, \mathcal{N}]\!]$, notant $\mathbf{c}_i = (\mathbf{c}_{i,kr-1} \ldots \mathbf{c}_{i,1}\mathbf{c}_{i,0})_2$ l'écriture binaire du i-ième coefficient de $\mathbf{c}$, alors :

$$\mathbf{c}_i = \sum_{j=0}^{kr-1} \mathbf{c}_{i,j} 2^j = \sum_{j=0}^{k-1} \sum_{t=0}^{r-1} \mathbf{c}_{i,jr+t} 2^{jr+t} = \sum_{j=0}^{k-1} \sum_{t=0}^{r-1} \left| \lfloor \frac{\mathbf{c}_i}{\beta^j} \rfloor \right|_\beta \beta^j.$$

De cette manière, en mémorisant toutes les valeurs $\left|x\beta^i\right|_{m_j}$ avec $x \in [\![0, \beta[\![$, la réduction du vecteur $\mathbf{c}$ s'exécute en au plus $nk\mathcal{N}$`AMEl`, au prix de $nkr\beta$ bits stockés.

Le principe de l'approche précédente par blocs amène naturellement à une technique moins gourmande en mémoire, mais plus coûteuse en terme d'opérations élémentaires. En précalculant la matrice suivante :

$$\mathbf{W}_{\beta,k,\mathcal{B}} = \begin{pmatrix} 1 & |\beta|_{m_1} & \ldots & \left|\beta^{k-1}\right|_{m_1} \\ 1 & |\beta|_{m_2} & \ldots & \left|\beta^{k-1}\right|_{m_2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & |\beta|_{m_n} & \ldots & \left|\beta^{k-1}\right|_{m_n} \end{pmatrix} \tag{4.71}$$

alors le calcul des résidus de $\mathbf{c}$ s'effectue simplement par le produit suivant :

$$\begin{pmatrix} \mathbf{c} \bmod m_1 \\ \mathbf{c} \bmod m_2 \\ \vdots \\ \mathbf{c} \bmod m_n \end{pmatrix} = \begin{pmatrix} |\mathbf{c}_1|_{m_1} & |\mathbf{c}_2|_{m_1} & \cdots & |\mathbf{c}_{\mathcal{N}}|_{m_1} \\ |\mathbf{c}_1|_{m_2} & |\mathbf{c}_2|_{m_2} & \cdots & |\mathbf{c}_{\mathcal{N}}|_{m_2} \\ \vdots & \vdots & \vdots & \vdots \\ |\mathbf{c}_1|_{m_n} & |\mathbf{c}_2|_{m_n} & \cdots & |\mathbf{c}_{\mathcal{N}}|_{m_n} \end{pmatrix} \equiv \mathbf{W}_{\beta,k,\mathcal{B}} \times \mathbf{C} \bmod \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix}$$
$$\equiv \mathbf{W}_{\beta,k,\mathcal{B}} \times \begin{pmatrix} |\mathbf{c}_1|_\beta & |\mathbf{c}_2|_\beta & \cdots & |\mathbf{c}_{\mathcal{N}}|_\beta \\ \left|\lfloor \frac{\mathbf{c}_1}{\beta} \rfloor\right|_\beta & \left|\lfloor \frac{\mathbf{c}_2}{\beta} \rfloor\right|_\beta & \cdots & \left|\lfloor \frac{\mathbf{c}_{\mathcal{N}}}{\beta} \rfloor\right|_\beta \\ \vdots & \vdots & \vdots & \vdots \\ \left|\lfloor \frac{\mathbf{c}_1}{\beta^{k-1}} \rfloor\right|_\beta & \left|\lfloor \frac{\mathbf{c}_2}{\beta^{k-1}} \rfloor\right|_\beta & \cdots & \left|\lfloor \frac{\mathbf{c}_{\mathcal{N}}}{\beta^{k-1}} \rfloor\right|_\beta \end{pmatrix} \bmod \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix}. \tag{4.72}$$

La taille de la matrice $\mathbf{W}_{\beta,k,\mathcal{B}}$ est de $n(k-1)r$ bits. Le coût d'une telle conversion est de $nk\mathcal{N}$ `MMEl` $+ n(k-1)\mathcal{N}$ `AMEl`. Si n est choisi de telle manière qu'il divise $\mathcal{N}$, alors la calcul de $\mathbf{W}_{\beta,n,\mathcal{B}} \times \mathbf{C}$ peut se faire par des techniques de multiplication efficace de matrices carrées (Coppersmith et Winograd 1987, Strassen 1969, Williams 2012). Cette considération permet d'évaluer la complexité de la réduction en $\mathcal{O}\left(\mathcal{N}n^{1+\epsilon}\right)$ `MMEl`, au prix d'une complexité spatiale de $n(n-1)r$ bits.

Kawamura et al. (2000) montrent que cette conversion peut se réaliser efficacement en utilisant l'architecture Cox-Rower. En effet, celle-ci se prête idéalement à l'approche par blocs via la décomposition d'un entier en base β. Avec une parallélisation maximale, la réduction d'un coefficient $\mathbf{c}_i$ s'exécute alors en k `ETEl`, et nk `MMEl` $+ n(k-1)$`AMEl`. Soit un total de $k\mathcal{N}$ `ETEl` et $nk\mathcal{N}$ `MMEl` $+ n(k-1)\mathcal{N}$ `AMEl`.

$$\text{Coûts binaire -> RNS : } \begin{cases} \text{précalculs : } n(k-1) \ \beta\text{-mots} \\ nk\mathcal{N} \ \texttt{MMEl}_\beta + n(k-1)\mathcal{N} \ \texttt{AMEl}_\beta \\ k\mathcal{N} \ \texttt{ETEl}_\beta \\ k = \lceil \log_\beta(c_\infty) \rceil \\ n = \lceil \log_\beta(c_\infty \mathcal{N} + 1) \rceil \end{cases} \tag{4.73}$$

Pour résumer, la considération du coût de la transformation en RNS dans le cadre de l'étude de complexité de l'algorithme CVP-RNS proposé ici se révèle plutôt délicate. Pour fixer clairement la situation, il s'agirait de préciser le contexte plus général dans lequel l'utilisation de l'algorithme s'inscrirait. Par exemple, l'hypothèse de réception des résidus de $\mathbf{c}$ peut sembler pertinente si le choix est fait de mener les opérations de chiffrement également en RNS. Dans un système comme proposé par Micciancio (2001), si le déterminant de $\mathbf{R}$ est peu friable, alors sa forme normale de Hermite pourra posséder des coefficients diagonaux de grande taille, et l'utilisation du RNS comme système de numération pour l'ensemble du protocole peut être pertinente.

Précalculs

Comme la technique d'accélération s'accorde avec tout type de conversion de base utilisée pour le calcul de $\left|\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right|_{2^d}$, puisqu'il s'agit alors seulement de rechercher un paramètre γ en conséquence, la conversion utilisée par la suite pour la réduction modulaire RNS est Bex_{crt}.

Les représentations de Montgomery utilisées ont été introduites en (4.69), soit :

$$\begin{cases} \tilde{\mathbf{R}} = 2\gamma M \times \mathbf{R}' \bmod (2d) \\ \tilde{\mathbf{d}} = M \times \mathbf{d} \bmod (2d) \end{cases} \tag{4.74}$$

Remarque 4.4 *La fonction $f_{\gamma,a}$, définie p. 161, sert à corriger le vecteur d'erreur dans le calcul de $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \bmod m_\sigma$. Ainsi, nous pouvons nous restreindre à récupérer seulement le résultat de la comparaison effectuée lors de l'évaluation de $f_{\gamma,a}$, sans effectuer l'éventuelle correction par $-\gamma$ du résidu modulo γ sur lequel la fonction est appliquée, celle-ci se faisant alors directement dans le canal $\mathbb{Z}/m_\sigma\mathbb{Z}$.*

Pour utiliser la remarque précédente, nous introduisons la fonction suivante :

$$\begin{array}{rccl} \tilde{f}_{\gamma,a}: & [\![0,\gamma[\![& \to & \{0,1\} \\ & z & \mapsto & \begin{cases} 1 \text{ si } z \geqslant |-a|_\gamma, \\ 0 \text{ sinon.} \end{cases} \end{array} \tag{4.75}$$

Ainsi, pour tout $z \in [\![0,\gamma[\![$, $f_{\gamma,a}(z) = z - \gamma \times \tilde{f}_{\gamma,a}(z)$. Par conséquent, un calcul du type $f_{\gamma,a}(z) \bmod m_\sigma$, nécessitant une comparaison entre z et $|-a|_\gamma$ et une éventuelle addition dans $\mathbb{Z}/\gamma\mathbb{Z}$, se réduit au calcul de $(z - |\gamma|_{m_\sigma}) \bmod m_\sigma$, nécessitant toujours la même comparaison entre z et $|-a|_\gamma$. Mais la correction s'effectue dans l'anneau plus petit $\mathbb{Z}/m_\sigma\mathbb{Z}$.

Dans $\mathcal{B}$ Comme la conversion de base utilisée pour le calcul de $\left|\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right|_\delta$ est Bex_{crt}, les précalculs sont identiques à (4.30). Soit :

$$\boxed{\forall m_i \in \mathcal{B}, \begin{cases} \tilde{\mathbf{R}}_{m_i} = -\dfrac{1}{\delta M_i} \times \tilde{\mathbf{R}} \bmod m_i, \\ \tilde{\mathbf{d}}_{m_i} = -\dfrac{1}{\delta M_i} \times \tilde{\mathbf{d}} \bmod m_i. \end{cases}} \tag{4.76}$$

Ainsi, le vecteur $\mathbf{q}$ des coefficients de Montgomery est le suivant :

$$\mathbf{q} = \sum_{i=1}^{n} \mathbf{a}^{(m_i)} M_i, \tag{4.77}$$

où $\mathbf{a}^{(m_i)}$ est le vecteur :

$$\mathbf{a}^{(m_i)} = \mathbf{c}\tilde{\mathbf{R}}_{m_i} + \tilde{\mathbf{d}}_{m_i} \bmod m_i. \tag{4.78}$$

Dans $\{\gamma\}$ Le détail des calculs à effectuer dans le canal $\mathbb{Z}/\gamma\mathbb{Z}$ donne :

$$\begin{aligned} \lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil + \mathbf{v_e} \bmod \gamma &= \lfloor \gamma \mathbf{p}\mathbf{R}^{-1} \rceil + \mathbf{v_e} \bmod \gamma \\ &= \frac{2\gamma\mathbf{c}\mathbf{R}' + \mathbf{d} - |2\gamma\mathbf{c}\mathbf{R}' + \mathbf{d}|_\delta}{\delta} + \mathbf{v_e} \bmod \gamma \\ &= \frac{2\gamma\mathbf{c}\mathbf{R}' + \mathbf{d} - \dfrac{\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}} + \mathbf{q}\delta}{M}}{\delta} \bmod \gamma \\ &= \frac{1}{2} - \frac{1}{\delta M}\mathbf{c}\tilde{\mathbf{R}} - \frac{1}{\delta M}\tilde{\mathbf{d}} - \sum_{i=1}^{n} \frac{\mathbf{a}^{(m_i)} M_i}{m_i} \bmod \gamma. \end{aligned}$$

Les précalculs et notations introduits sont :

$$(\gamma)\begin{cases}\tilde{\mathbf{R}}_\gamma = -\dfrac{1}{\delta M} \times \tilde{\mathbf{R}} \bmod \gamma \\ \tilde{\mathbf{d}}_\gamma = \mathbf{v}_{1/2} - \dfrac{1}{\delta M} \times \tilde{\mathbf{d}} \bmod \gamma \\ \tilde{\delta}_\gamma^{(i)} = -\dfrac{1}{m_i} \bmod \gamma,\ \forall m_i \in \mathcal{B} \\ \tilde{\delta}_\gamma^{(0)} = -a \bmod \gamma \end{cases} \tag{4.79}$$

et :

$$\mathbf{a}^{(\gamma)} = \mathbf{c}\tilde{\mathbf{R}}_\gamma + \tilde{\mathbf{d}}_\gamma \bmod \gamma. \tag{4.80}$$

La valeur $\tilde{\delta}_\gamma^{(0)} = -a \bmod \gamma$ sert à l'évaluation de la fonction $\tilde{f}\gamma, a$.

Par conséquent nous avons :

$$\lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil + \mathbf{v_e} \bmod \gamma = \mathbf{a}^{(\gamma)} + \sum_{i=1}^{n} \mathbf{a}^{(m_i)} \tilde{\delta}_\gamma^{(i)} \bmod \gamma. \tag{4.81}$$

Dans $\{m_\sigma\}$ Le canal $\mathbb{Z}/m_\sigma\mathbb{Z}$ reçoit le calcul de $\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}$ via la Formule (4.64), soit :

$$\begin{aligned}
&\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R} \bmod m_\sigma \\
&= \mathbf{c} - \frac{1}{\gamma}\left[\frac{2\gamma \mathbf{c}\mathbf{R}' + \mathbf{d}}{\delta} - \frac{\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}} + \mathbf{q}\delta}{\delta M} - f_{\gamma,a}\left(\lfloor \gamma \mathbf{c}\mathbf{R}^{-1} \rceil + \mathbf{v_e} \bmod \gamma\right)\right] \mathbf{R} \bmod m_\sigma \\
&= \mathbf{c} - \frac{2\gamma}{\gamma\delta}\mathbf{c}\mathbf{R}'\mathbf{R} - \frac{1}{\gamma}\mathbf{v}_{1/2}\mathbf{R} + \frac{1}{\gamma\delta M}\left(\mathbf{c}\tilde{\mathbf{R}} + \tilde{\mathbf{d}}\right)\mathbf{R} \\
&+ \left(\frac{1}{\gamma M}\mathbf{q} + \frac{1}{\gamma} f_{\gamma,a}\left(\mathbf{a}^{(\gamma)} + \sum_{i=1}^{n} \mathbf{a}^{(m_i)} \tilde{\delta}_\gamma^{(i)} \bmod \gamma\right)\right) \mathbf{R} \bmod m_\sigma \\
&= \frac{1}{\gamma\delta M}\mathbf{c}\tilde{\mathbf{R}}\mathbf{R} + \frac{1}{\gamma\delta M}\tilde{\mathbf{d}}\mathbf{R} - \frac{1}{\gamma}\mathbf{v}_{1/2}\mathbf{R} \\
&+ \frac{1}{\gamma}\left(\frac{1}{M}\sum_{i=1}^{n} \mathbf{a}^{(m_i)} M_i + f_{\gamma,a}\left(\mathbf{a}^{(\gamma)} + \sum_{i=1}^{n} \mathbf{a}^{(m_i)} \tilde{\delta}_\gamma^{(i)} \bmod \gamma\right)\right) \mathbf{R} \bmod m_\sigma.
\end{aligned}$$

Les précalculs et notations sont donc :

$$(m_\sigma)\begin{cases}\mathbf{R}_{m_\sigma} = \dfrac{1}{\gamma} \times \mathbf{R} \bmod m_\sigma \\ \tilde{\mathbf{R}}_{m_\sigma} = \dfrac{1}{\gamma\delta M} \times \tilde{\mathbf{R}}\mathbf{R} \bmod m_\sigma \\ \tilde{\mathbf{d}}_{m_\sigma} = \dfrac{1}{\gamma\delta M} \times \tilde{\mathbf{d}}\mathbf{R} - \dfrac{1}{\gamma}\mathbf{v}_{1/2}\mathbf{R} \bmod m_\sigma \\ \tilde{\delta}_{m_\sigma}^{(i)} = \dfrac{1}{m_i} \bmod m_\sigma,\ \forall m_i \in \mathcal{B} \\ \tilde{\delta}_{m_\sigma}^{(0)} = \gamma \bmod m_\sigma \end{cases} \tag{4.82}$$

et :

$$\mathbf{b}^{(m_\sigma)} = \mathbf{c}\tilde{\mathbf{R}}_{m_\sigma} + \tilde{\mathbf{d}}_{m_\sigma} \bmod m_\sigma. \tag{4.83}$$

En introduisant de plus la fonction $\tilde{f}_{\gamma,a}$ déduite de $f_{\gamma,a}$ par $f_{\gamma,a}(x) = x - \gamma \times \tilde{f}_{\gamma,a}(x)$ (cf. (4.75)), il vient alors :

$$\begin{aligned}
&\mathbf{c} - \lfloor \gamma \mathbf{c} \mathbf{R}^{-1} \rceil \mathbf{R} \bmod m_\sigma \\
&= \mathbf{b}^{(m_\sigma)} + \Bigg[\sum_{i=1}^{n} \mathbf{a}^{(m_i)} \tilde{\delta}_{m_\sigma}^{(i)} + \left(\mathbf{a}^{(\gamma)} + \sum_{i=1}^{n} \mathbf{a}^{(m_i)} \tilde{\delta}_{\gamma}^{(i)} \bmod \gamma \right) \\
&- \tilde{\delta}_{m_\sigma}^{(0)} \times \tilde{f}_{\gamma,a} \left(\mathbf{a}^{(\gamma)} + \sum_{i=1}^{n} \mathbf{a}^{(m_i)} \tilde{\delta}_{\gamma}^{(i)} \bmod \gamma \right) \Bigg] \mathbf{R}_{m_\sigma} \bmod m_\sigma.
\end{aligned} \tag{4.84}$$

Algorithme CVP RNS

L'Algorithme 23 retourne $\mathbf{p} \bmod m_\sigma$ en se basant sur les précalculs et formules précédentes.

Le paramètre a de la fonction $f_{\gamma,a}$ satisfait les Conditions (4.59). Ainsi, l'évaluation de $\tilde{f}_{\gamma,a}\left(\mathbf{a}_k^{(\gamma)}\right)$ à l'étape 8 de l'Algorithme 23 est une comparaison de deux entiers de la taille de γ, dont l'un deux, $|a|_\gamma$, est précalculé. Et vu la Remarque 4.3, la comparaison peut s'effectuer aisément si le choix de a est judicieux.

Enfin, nous rappelons que la base $\mathcal{B}$ doit vérifier $M > \mathcal{N} c_\infty + 1$.

Algorithme 23 : CVP_RNS $(\mathbf{c}, \mathbf{R})$

Données : Le vecteur $\mathbf{c}$ exprimé dans la base RNS $\mathcal{B} \cup \{\gamma, m_\sigma\}$. Les valeurs précalculées (4.76), (4.79) et (4.82). Un entier a vérifiant (4.66) (p.165).

Résultat : $\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \times \mathbf{R} \bmod m_\sigma$.

1 **début**
2 **pour** $m \in \mathcal{B} \cup \{\gamma\}$ **faire**
3 $\mathbf{a}^{(m)} \leftarrow \mathbf{c}\tilde{\mathbf{R}}_m + \tilde{\mathbf{d}}_m \bmod m$
4 $\mathbf{b}^{(m_\sigma)} \leftarrow \mathbf{c}\tilde{\mathbf{R}}_{m_\sigma} + \tilde{\mathbf{d}}_{m_\sigma} \bmod m_\sigma$
5 $\mathbf{a}^{(m_\sigma)} \leftarrow (0, \ldots, 0) \in \mathbb{Z}^{\mathcal{N}}$
6 **pour** $m \in \{\gamma, m_\sigma\}$ **faire**
7 $\mathbf{a}^{(m)} \leftarrow \mathbf{a}^{(m)} + \sum_{i=1}^{n} \mathbf{a}^{(m_i)} \times \tilde{\delta}_m^{(i)} \bmod m$
8 $\mathbf{a}^{(m_\sigma)} \leftarrow \mathbf{a}^{(m_\sigma)} + \mathbf{a}^{(\gamma)} - \tilde{\delta}_{m_\sigma}^{(0)} \times \tilde{f}_{\gamma,a}\left(\mathbf{a}^{(\gamma)}\right) \bmod m_\sigma$
9 $\mathbf{p}_{m_\sigma} \leftarrow \mathbf{b}^{(m_\sigma)} + \mathbf{a}^{(m_\sigma)} \mathbf{R}_{m_\sigma} \bmod m_\sigma$
10 **retourner** $\mathbf{p}_{m_\sigma}$

Complexités spatiale et temporelle

À β donné, afin de maximiser l'ensemble des réseaux pour lesquels la technique d'accélération est la plus efficace, la taille de γ est supposée être au plus celle de β. De ce fait, les précalculs (4.76) et (4.79) concernant la base $\mathcal{B} \cup \{\gamma\}$ consistent en $n+1$ matrices $\tilde{\mathbf{R}}_m$ et $n+1$ vecteurs $\tilde{\mathbf{d}}_m$, et $(n+1)$ scalaires $\tilde{\delta}_\gamma^{(i)}$, ce qui représente $(n+1)(\mathcal{N}^2 + \mathcal{N} + 1)$ β-mots. Les précalculs (4.82) requièrent pour leur part une mémorisation de $2\mathcal{N}^2 + \mathcal{N} + n + 1$ m_σ-mots.

base	$\mathcal{B}$	$\{\gamma\}$	$\{m_\sigma\}$
β-mots	$n(\mathcal{N}^2+\mathcal{N})$	$(\mathcal{N}^2+\mathcal{N}+n+1)$	$\times$
m_σ-mots	$\times$	$\times$	$2\mathcal{N}^2+\mathcal{N}+n+1$

TABLE 4.9 – *Coût des précalculs de l'Algorithme 23.*

De même que pour la discussion concernant la complexité spatiale de l'Algorithme 22, nous approximons la valeur de n par $\log_\beta(c_\infty\mathcal{N})$. Ainsi, la complexité spatiale binaire de l'Algorithme 23 est :

$$\begin{aligned}\mathcal{C}_s(\text{CVP\_RNS}) &= \left(\mathcal{N}^2+\mathcal{N}+1\right)\log_2(c_\infty\beta\mathcal{N}) \\ &\quad+\left(2\mathcal{N}^2+\mathcal{N}+\log_\beta(c_\infty\mathcal{N})+1\right)\log_2(2\sigma).\end{aligned} \tag{4.85}$$

Concernant les précalculs pour la transformation du binaire vers le RNS via la conversion matricielle (4.72) ou par l'utilisation d'un Cox-Rower, il s'agit de stocker la matrice $\mathbf{W}_{\beta,k,\mathcal{B}}$ (4.71), soit dans le pire cas $(k-1)n$ β-mot, où nous rappelons que $k=\lceil\log_2(c_\infty)\rceil$. Ainsi, le surcoût en terme de mémoire est environ $\log_2(\mathcal{N}c_\infty)\log_2(c_\infty)$ bits. Il est à noter que ce surcout concerne également l'approche RNS-MRS (Algo. 22).

Le détail des coûts en termes de `MME1` et `AME1` de l'Algorithme 23 est donné dans le Tableau (4.10). Ce coût se résume ainsi à :

$$\begin{aligned}\mathcal{C}_T(\text{CVP\_RNS}) &= \left[2\mathcal{N}^2+\log_\beta(c_\infty\mathcal{N})\,\mathcal{N}\right]\ \texttt{MME1}_{2\sigma} \\ &\quad+\left[2\mathcal{N}^2+\log_\beta(c_\infty\beta\mathcal{N})\,\mathcal{N}\right]\ \texttt{AME1}_{2\sigma} \\ &\quad+\left[\log_\beta(c_\infty\beta\mathcal{N})\,\mathcal{N}^2+\log_\beta(c_\infty\mathcal{N})\,\mathcal{N}\right]\ \texttt{A/MME1}_{\beta}.\end{aligned} \tag{4.86}$$

Étape	$\mathcal{B}$	$\{\gamma\}$	$\{m_\sigma\}$
3	$n\mathcal{N}^2$	$\mathcal{N}^2$	$\times$
4	$\times$	$\times$	$\mathcal{N}^2$
7	$\times$	$n\mathcal{N}$	$n\mathcal{N}$
8	$\times$	$\times$	$\mathcal{N}$ $\texttt{AME1}_{m_\sigma}$
9	$\times$	$\times$	$\mathcal{N}^2$
Total `MME1`	$n\mathcal{N}^2$	$\mathcal{N}^2+n\mathcal{N}$	$2\mathcal{N}^2+n\mathcal{N}$
Total `AME1`	$n\mathcal{N}^2$	$\mathcal{N}^2+n\mathcal{N}$	$2\mathcal{N}^2+(n+1)\mathcal{N}$

TABLE 4.10 – *Nombre de multiplications et additions modulaires élémentaires de l'Algorithme 23.*

Concernant la complexité en terme de nombre d'étapes élémentaires `ETE1`, vu que la base $\mathcal{B}$ est uniquement utilisée lors de l'étape 3, hypothèse est faite d'une parallélisation de la forme $\frac{n}{\alpha}+2$, pour α entier, c'est-à-dire que $\frac{n}{\alpha}$ canaux physiques sont assignés pour les calculs dans $\mathcal{B}$, et 2 canaux sont spécifiquement alloués à la base $\{\gamma, m_\sigma\}$. Ainsi, les étapes 3 et 4 sont réalisées en $\alpha\mathcal{N}^2$ `ETE1`. La boucle `for` de l'étape 6 en nécessite $n\mathcal{N}$ en parallèle sur $\{\gamma, m_\sigma\}$. En supposant que la réalisation de la fonction $f_{\gamma,a}$ est rendue triviale par un choix judicieux du paramètre a, alors les deux étapes finales requièrent $\left(\mathcal{N}^2+\mathcal{N}\right)$ $\texttt{ETE1}_{m_\sigma}$.

Pour résumer, en considérant que $c_\infty \sim d$ à des fins de comparaison avec la complexité (4.49) de l'approche RNS-MRS, le nombre d'étapes permis par le

parallélisme du RNS est le suivant :

$$\mathcal{C}_{\texttt{ETEl}}\,(\text{CVP\_RNS}) = \left[\alpha\mathcal{N}^2 + n\mathcal{N}\right]\ \texttt{ETEl} + \left[\mathcal{N}^2 + \mathcal{N}\right]\ \texttt{ETEl}_{m_\sigma}. \tag{4.87}$$

Exemple 4.7 *Nous reprenons les données de l'Exemple 4.6 (p. 157), et nous les rappelons. Soit* $\mathcal{N} = 4$ *et une base* $\mathbf{R}$ *d'un réseau* $\mathcal{R}$ *de* $\mathbb{Z}^4$ *:*

$$\mathbf{R} = \begin{pmatrix} 1 & -2 & -2 & 15 \\ -17 & 1 & 1 & 1 \\ -3 & 11 & -13 & 6 \\ 2 & 18 & 6 & -2 \end{pmatrix}$$

ainsi que les constantes suivantes :

$$\sigma = 3,\ d = \frac{\delta}{2} = \det\mathbf{R} = 74268,\ \rho_{\mathbf{R}} = \frac{7928}{74268},\ \epsilon_{\mathbf{R}} = \frac{1}{2} - \sigma\rho_{\mathbf{R}} = \frac{2225}{12378}.$$

La base RNS $\mathcal{B}$ *doit vérifier* $M > 4 \times \max\{\|\mathbf{c}\|_\infty \mid \mathbf{c} \in \mathcal{C}\} + 1$*. L'espace des chiffrés* $\mathcal{C}$ *est supposé réduit modulo* d*. Ainsi,* $\mathcal{B}$ *doit vérifier* $M > 2\delta + 1 = 297073$ *et* M *premier avec* δ*. Soit donc* $\mathcal{B} = \{m_1 = 19, m_2 = 23, m_3 = 29, m_4 = 31\}$ *(ainsi* $M = 392863$ *et* $n = 4$*), et* $m_\sigma = 7 \geqslant 2\sigma + 1$*.*

Par suite, il vient :

$$\gamma_{\mathbf{R},4} = \lceil\frac{n+2}{2\epsilon_{\mathbf{R}}}\rceil = 16.$$

Soit alors $\gamma = 17$*, entier premier avec* M *et* δ*. a peut être choisi (cf. (4.66)) dans l'intervalle* $[\![n - \lfloor -\gamma\sigma\rho_{\mathbf{R}}\rfloor, \gamma - 1 - \lfloor\gamma\sigma\rho_{\mathbf{R}}\rfloor]\!] = [\![9, 11]\!]$*. Soit par exemple* $a = 10$*.*

Les représentations de Montgomery (4.74), qui servent uniquement aux précalculs et ne sont pas stockées en mémoire, sont :

$$\tilde{\mathbf{R}} = \begin{pmatrix} 27456 & 82248 & 143688 & 9696 \\ 74264 & 120284 & 96904 & 90886 \\ 146968 & 139684 & 109688 & 90074 \\ 96984 & 98496 & 8064 & 132408 \end{pmatrix},\ \tilde{\mathbf{d}} = \begin{pmatrix} 74268 & 74268 & 74268 & 74268 \end{pmatrix}.$$

Les précalculs (4.76) pour $\mathcal{B}$ *sont les suivants :*

$$\tilde{\mathbf{R}}_{m_1} = \begin{pmatrix} 7 & 17 & 13 & 4 \\ 8 & 3 & 9 & 6 \\ 2 & 10 & 7 & 3 \\ 18 & 0 & 18 & 17 \end{pmatrix},\ \tilde{\mathbf{R}}_{m_2} = \begin{pmatrix} 14 & 0 & 22 & 8 \\ 7 & 14 & 19 & 8 \\ 20 & 19 & 13 & 9 \\ 1 & 15 & 21 & 7 \end{pmatrix},$$

$$\tilde{\mathbf{R}}_{m_3} = \begin{pmatrix} 10 & 15 & 10 & 23 \\ 3 & 28 & 20 & 0 \\ 14 & 17 & 23 & 0 \\ 1 & 16 & 22 & 21 \end{pmatrix},\ \tilde{\mathbf{R}}_{m_4} = \begin{pmatrix} 24 & 19 & 30 & 23 \\ 4 & 9 & 11 & 2 \\ 1 & 11 & 7 & 4 \\ 5 & 28 & 9 & 8 \end{pmatrix},$$

$$\tilde{\mathbf{d}}_{m_1} = \begin{pmatrix} 17 & 17 & 17 & 17 \end{pmatrix},\ \tilde{\mathbf{d}}_{m_2} = \begin{pmatrix} 13 & 13 & 13 & 13 \end{pmatrix},$$

$$\tilde{\mathbf{d}}_{m_3} = \begin{pmatrix} 18 & 18 & 18 & 18 \end{pmatrix},\ \tilde{\mathbf{d}}_{m_4} = \begin{pmatrix} 13 & 13 & 13 & 13 \end{pmatrix}.$$

Les précalculs (4.79) pour γ *sont :*

$$\tilde{\mathbf{R}}_\gamma = \begin{pmatrix} 8 & 16 & 15 & 14 \\ 13 & 4 & 15 & 15 \\ 7 & 11 & 15 & 13 \\ 9 & 1 & 14 & 11 \end{pmatrix},\ \tilde{\mathbf{d}}_\gamma = \begin{pmatrix} 3 & 3 & 3 & 3 \end{pmatrix},$$

$$\tilde{\delta}_\gamma^{(0)} = 7,\ \tilde{\delta}_\gamma^{(1)} = 8,\ \tilde{\delta}_\gamma^{(2)} = 14,\ \tilde{\delta}_\gamma^{(3)} = 7,\ \tilde{\delta}_\gamma^{(4)} = 6.$$

Enfin, les précalculs (4.82) pour m_σ sont :

$$\mathbf{R}_{m_\sigma} = \begin{pmatrix} 5 & 4 & 4 & 5 \\ 6 & 5 & 5 & 5 \\ 6 & 6 & 5 & 2 \\ 3 & 6 & 2 & 4 \end{pmatrix},\ \tilde{\mathbf{R}}_{m_\sigma} = \begin{pmatrix} 5 & 2 & 5 & 5 \\ 0 & 3 & 5 & 3 \\ 2 & 3 & 0 & 2 \\ 2 & 5 & 3 & 5 \end{pmatrix},\ \tilde{\mathbf{d}}_{m_\sigma} = \begin{pmatrix} 2 & 0 & 3 & 3 \end{pmatrix},$$
$$\tilde{\delta}^{(0)}_{m_\sigma} = 3,\ \tilde{\delta}^{(1)}_{m_\sigma} = 4,\ \tilde{\delta}^{(2)}_{m_\sigma} = 1,\ \tilde{\delta}^{(3)}_{m_\sigma} = 5,\ \tilde{\delta}^{(4)}_{m_\sigma} = 3.$$

Nous réutilisons également le vecteur $\mathbf{c}$ *de l'Exemple 4.6, à savoir*

$$\mathbf{c} = \begin{pmatrix} 22143 & 357 & 7328 & 17985 \end{pmatrix} = \left[\begin{pmatrix} -3 & 2 & 1 & -2 \end{pmatrix} + \begin{pmatrix} 1171 & 3230 & -397 & 213 \end{pmatrix} \mathbf{R} \right] \bmod d.$$

Les résidus de $\mathbf{c}$ *dans* $\mathcal{B} \cup \{\gamma, m_\sigma\}$ *sont :*

$$\mathbf{c}_{m_1} = \begin{pmatrix} 8 & 15 & 13 & 11 \end{pmatrix},\ \mathbf{c}_{m_2} = \begin{pmatrix} 17 & 12 & 14 & 22 \end{pmatrix},$$
$$\mathbf{c}_{m_3} = \begin{pmatrix} 16 & 9 & 20 & 5 \end{pmatrix},\ \mathbf{c}_{m_4} = \begin{pmatrix} 9 & 16 & 12 & 5 \end{pmatrix},$$
$$\mathbf{c}_{\gamma} = \begin{pmatrix} 9 & 0 & 1 & 16 \end{pmatrix},\ \mathbf{c}_{m_\sigma} = \begin{pmatrix} 2 & 0 & 6 & 2 \end{pmatrix}.$$

Toutes les données sont maintenant prêtes.
La ***première étape*** *de L'Algorithme 23 donne les calculs suivants :*

$$\mathbf{a}^{(m_1)} = \begin{pmatrix} 18 & 5 & 13 & 4 \end{pmatrix},\ \mathbf{a}^{(m_2)} = \begin{pmatrix} 16 & 18 & 17 & 19 \end{pmatrix},$$
$$\mathbf{a}^{(m_3)} = \begin{pmatrix} 26 & 2 & 0 & 27 \end{pmatrix},\ \mathbf{a}^{(m_4)} = \begin{pmatrix} 20 & 11 & 30 & 30 \end{pmatrix},$$
$$\mathbf{a}^{(\gamma)} = \begin{pmatrix} 5 & 4 & 3 & 12 \end{pmatrix},\ \mathbf{b}^{(m_\sigma)} = \begin{pmatrix} 0 & 4 & 5 & 0 \end{pmatrix}.$$

La ***deuxième étape*** *est la conversion de base de* $\mathcal{B}$ *vers* $\{\gamma, m_\sigma\}$*, ce qui donne :*

$$\mathbf{a}^{(\gamma)} = \mathbf{a}^{(\gamma)} + \sum_{i=1}^{4} \tilde{\delta}^{(i)}_{\gamma} \times \mathbf{a}^{(m_i)} \bmod \gamma = \begin{pmatrix} 12 & 2 & 15 & 16 \end{pmatrix},$$
$$\mathbf{a}^{(m_\sigma)} = \sum_{i=1}^{4} \tilde{\delta}^{(i)}_{m_\sigma} \times \mathbf{a}^{(m_i)} \bmod m_\sigma = \begin{pmatrix} 6 & 4 & 5 & 6 \end{pmatrix}.$$

La ***troisième étape*** *consiste à comparer chaque coefficient de* $\mathbf{a}^{(\gamma)}$ *avec* $\tilde{\delta}^{(0)}_{\gamma} = 7$*, ce qui nous donne* $\tilde{f}_{\gamma,a}\left(\mathbf{a}^{(\gamma)}\right) = \begin{pmatrix} 1 & 0 & 1 & 1 \end{pmatrix}$.
La ***quatrième étape*** *de calcul est :*

$$\mathbf{a}^{(m_\sigma)} = \mathbf{a}^{(m_\sigma)} + \mathbf{a}^{(\gamma)} - \tilde{\delta}^{(i)}_{m_\sigma} \times \tilde{f}_{\gamma,a}\left(\mathbf{a}^{(\gamma)}\right) \bmod m_\sigma = \begin{pmatrix} 1 & 6 & 3 & 5 \end{pmatrix}.$$

Enfin, la ***dernière étape*** *est donnée par :*

$$\mathbf{p}_{m_\sigma} = \mathbf{b}^{(m_\sigma)} + \mathbf{a}^{(m_\sigma)} \mathbf{R}_{m_\sigma} \bmod m_\sigma = \begin{pmatrix} 4 & 2 & 1 & 5 \end{pmatrix}.$$

Finalement,

$$\mathbf{p}_{m_\sigma} \operatorname{modc} m_\sigma = \begin{pmatrix} -3 & 2 & 1 & -2 \end{pmatrix} = \mathbf{p}.$$

Comparaison avec l'approche hybride RNS-MRS

Nous récapitulons dans cette partie les différences de coût entre l'Algorithme 22 du CVP RNS-MRS, et l'Algorithme 23 du CVP RNS, avec une parallélisation maximale pour ce dernier (*i.e.* $\alpha = 1$ dans (4.87)).

Les deux algorithmes ont en commun une base RNS principale $\mathcal{B}$ de n moduli de taille β vérifiant $n = \lceil \log_\beta (\mathcal{N} c_\infty + 1) \rceil$, ainsi qu'un modulus $m_\sigma \geqslant 2\sigma + 1$. Le premier algorithme requiert en plus une seconde base $\mathcal{B}'$ de ℓ β-moduli avec $\ell = \lceil \log_\beta (4d) \rceil$ et un modulus $\tilde{m} \geqslant n$. Le second n'utilise qu'un β-modulus supplémentaire γ.

Outre les gains nets récapitulés dans le Tableau 4.11, l'approche purement RNS autorise une grande flexibilité en terme de parallélisation, justement due à l'absence de conversion vers le MRS.

	Surcoût $\mathcal{C}_{CVP-RNSMRS} - \mathcal{C}_{CVP-RNS}$
bits	$\log_2(4d)\left[\mathcal{N}^2 + \mathcal{N} + \log_2(c_\infty \beta \mathcal{N})\right] - \log_2(\beta)$
$\texttt{A/MMEl}_\beta$	$\ell\mathcal{N}^2 + \ell\left(n + \frac{\ell+1}{2}\right)\mathcal{N}$ $\sim \log_\beta(4d)\left(\mathcal{N}^2 + \log_\beta\left(2c_\infty d^{\frac{1}{2}}\beta^{\frac{1}{2}}\mathcal{N}\right)\mathcal{N}\right)$
$\texttt{MMEl}_{m_\sigma}$	$\mathcal{N}$
$\texttt{AMEl}_{m_\sigma}$	0
ETEl	$\left[\mathcal{N}^2 + (\ell+4)\mathcal{N}\right]$ ETEl $- \mathcal{N}$ $\texttt{ETEl}_{m_\sigma}$ $\sim \left[\mathcal{N}^2 + (\log_\beta(4d) + 4)\mathcal{N}\right]$ ETEl $- \mathcal{N}$ $\texttt{ETEl}_{m_\sigma}$

TABLE 4.11 – *Coûts supplémentaires de l'approche RNS-MRS (Algorithme 22) par rapport à la méthode entièrement RNS (Algorithme 23) pour la résolution du CVP.*

Pour illustrer plus clairement les bénéfices de la méthode RNS par rapport à l'approche RNS-MRS, nous considérons que toute base $\mathbf{R}$ est issue d'un tirage dans $[\![-\mathcal{N}, \mathcal{N}]\!]^{\mathcal{N}^2}$ (cf. suggestion de Micciancio). Cette hypothèse permet de se donner un majorant de d via la borne de Hadamard : $d \leqslant \mathcal{N}^{\frac{3}{2}\mathcal{N}}$. De plus, les coefficients d'un chiffré $\mathbf{c}$ sont supposés bornés par d (*i.e.* $c_\infty \leqslant d$). Ces constatations permettent donc de fixer les contraintes sur les tailles des bases RNS.

Dans les deux approches, la base RNS principale $\mathcal{B}$ doit vérifier la même contrainte : $M > c_\infty \mathcal{N} + 1$. Par suite, le paramètre n est identique dans les deux cas. Puis, la base auxiliaire $\mathcal{B}'$ de l'approche mixte RNS-MRS vérifie $M' > 4d$. Les données n et ℓ exploitées dans les Figures 4.11 et 4.12 représentent donc le nombre suffisant de moduli premiers de taille β pour que $\mathcal{B}$ et $\mathcal{B}'$ vérifient :

$$\begin{cases} \prod_{i=1}^{n} m_i > \mathcal{N}^{\frac{3}{2}\mathcal{N}+1} + 1 \\ \prod_{j=1}^{\ell} m'_j > 4\mathcal{N}^{\frac{3}{2}\mathcal{N}} \end{cases} \tag{4.88}$$

Enfin, nous supposons que $\gamma_{\mathbf{R},n} < \beta$.

Une fois les paramètres n et ℓ ainsi obtenus, la Figure 4.11 montre le ratio des coûts en mémoire en termes de β-mots (cf. Tableaux (4.6) et (4.9) des

précalculs des deux approches, pour $\sigma = 3$:

$$\frac{\mathcal{C}_S(RNSMRS)}{\mathcal{C}_S(RNS)} = \frac{(2\mathcal{N}^2+\mathcal{N}+n+1)\log_\beta(2\sigma)+(\mathcal{N}^2+\mathcal{N}+1)(n+\ell+1)+n\ell-1}{(2\mathcal{N}^2+\mathcal{N}+n+1)\log_\beta(2\sigma)+(\mathcal{N}^2+\mathcal{N}+1)(n+1)}. \tag{4.89}$$

Le comportement singulier de la courbe rouge dédiée à la plus petite taille de mot $\beta = 2^{16}$ est un effet de bord de la représentation en termes de β-mots. Cela est dû à une inhomogénéité des tailles des $n+\ell$ moduli de l'approche RNS-MRS plus marquée que pour les seuls n moduli de l'approche RNS. Avec $\mathcal{N}$ croissant et β petit, les Conditions 4.88 épuisent plus rapidement le nombre de moduli premiers dont la taille est au plus proche de celle de β. Ainsi, le nombre de moduli $n+\ell$ augmente en même temps que leur taille binaire décroît. Autrement dit et *a contrario*, plus β est grand, plus la représentation en termes de β-mots reflète fidèlement la taille binaire réelle des précalculs. De cette même constatation nous pouvons déduire que, du fait que le rapport des contraintes (4.88) sur les tailles de M et M' converge vers 1 selon $\mathcal{N}$, n et ℓ sont alors d'autant plus proches. Ainsi, le nombre de moduli nécessaire pour la première approche est environ le double de celui de la seconde méthode, d'où un ratio proche de 2, ce que reflète la Figure 4.11.

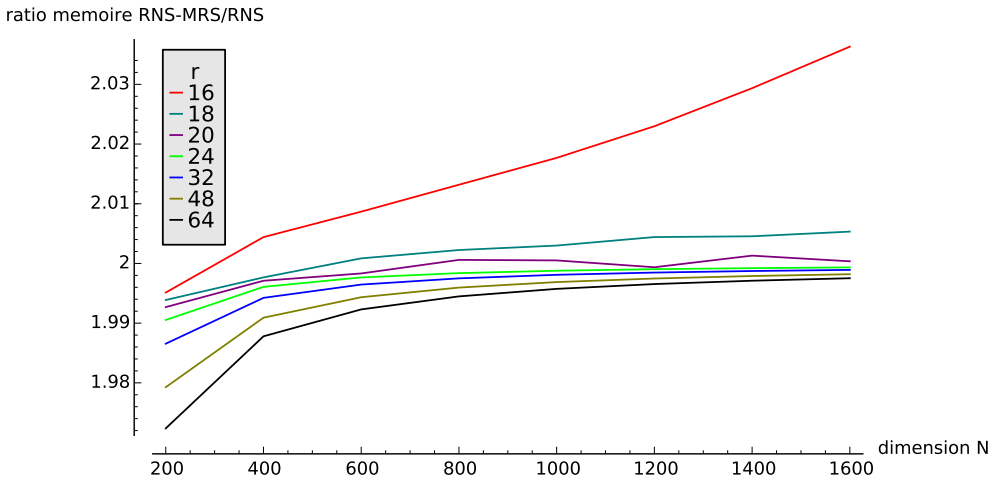


FIGURE 4.11 – *Ratio RNS-MRS/RNS des coûts mémoire pour différentes dimensions $\mathcal{N}$ et tailles $\beta = 2^r$.*

La Figure 4.12 illustre le ratio des coûts en multiplications $\texttt{MMEl}_\beta$ donnés dans les Tableaux 4.7 et 4.10, soit :

$$\frac{\mathcal{C}_{\texttt{MMEl}_\beta}(RNSMRS)}{\mathcal{C}_{\texttt{MMEl}_\beta}(RNS)} = \frac{(n+\ell+1)\mathcal{N}^2+(n\ell+\ell\frac{\ell+1}{2}+n)\mathcal{N}}{(n+1)\mathcal{N}^2+n\mathcal{N}}. \tag{4.90}$$

Les différences observées entre les différents β tiennent notamment au fait que pour une dimension $\mathcal{N}$ donnée, lorsque β croît les paramètres n et ℓ diminuent. De ce fait, ceci donne moins de poids au terme $\frac{(n\ell+\ell\frac{\ell+1}{2}+n)\mathcal{N}}{(n+1)\mathcal{N}^2+n\mathcal{N}}$ par rapport à $\frac{(n+\ell+1)\mathcal{N}^2}{(n+1)\mathcal{N}^2+n\mathcal{N}}$ qui est d'autant plus proche de 2, à $n \sim \ell$ fixés, que $\mathcal{N}$ est grand.

Enfin, une comparaison pertinente pour le RNS est celle du nombre d'étapes parallèles nécessaire en supposant un même nombre de canaux physiques. Une parallélisation totale de l'approche RNS-MRS mobilise $n+\ell+2$ canaux,

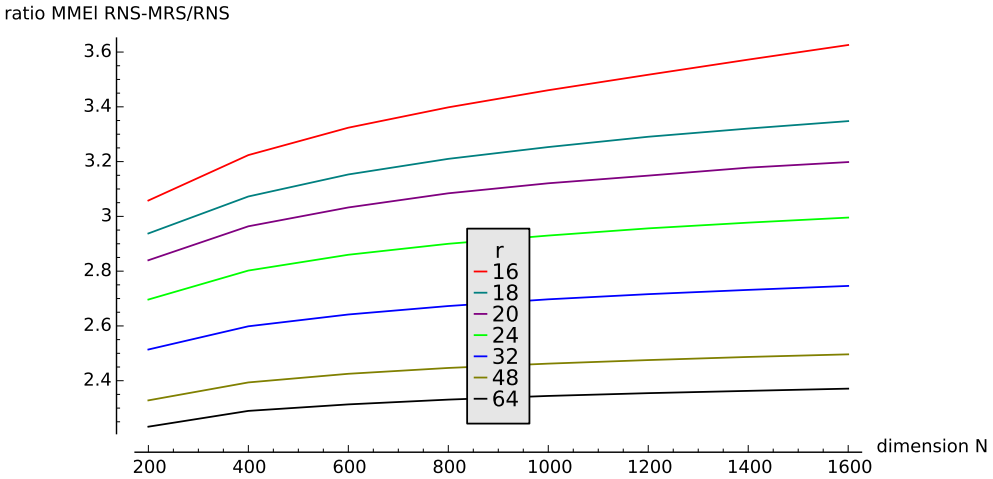


FIGURE 4.12 – *Ratio RNS-MRS/RNS des coûts en* $\texttt{MMEl}_\beta$ *pour différentes dimensions* $\mathcal{N}$ *et tailles* $\beta = 2^r$.

contre $n+2$ pour la méthode RNS. Pour simplifier, nous pouvons, pour ce faire, considérer que les nombres de moduli de $\mathcal{B}$ et $\mathcal{B}'$ de l'approche RNS-MRS sont comparables. Comme les opérations les plus coûteuses sont celles des canaux de taille β (par rapport au canal $\mathbb{Z}/m_\sigma\mathbb{Z}$), nous montrons dans la Figure 4.13 le ratio suivant :

$$\frac{\mathcal{C}_{\texttt{ETEl}_\beta}(RNSMRS)}{\mathcal{C}_{\texttt{ETEl}_\beta}(RNS)} = \frac{2\mathcal{N}^2 + (n+\ell+4)\mathcal{N}}{\mathcal{N}^2 + n\mathcal{N}} \tag{4.91}$$

où il est supposé que la surface disponible est identique pour les deux approches. En l'occurrence, le terme $2\mathcal{N}^2$ du numérateur provient du découpage de l'étape 3 de la Table 4.7 en deux étapes successives.

Moyennant les effets de bord expliqués précédemment concernant les plus petits β, ce rapport converge vers 2. Cependant il est de plus nécessaire de souligner que l'approche entièrement RNS offre l'avantage d'une plus grande flexibilité au niveau du degré de parallélisation grâce à l'absence de conversion en MRS.

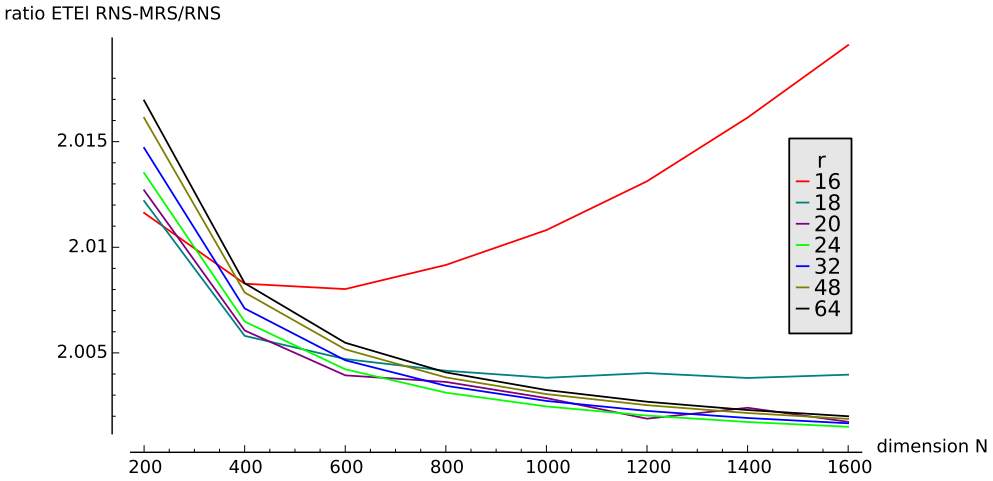


FIGURE 4.13 – *Ratio RNS-MRS/RNS du nombre d'étapes élémentaires* $\texttt{ETEl}_\beta$ *dans les canaux de taille* β *pour différentes dimensions* $\mathcal{N}$ *et tailles* $\beta = 2^r$.

En conclusion, l'Algorithme 23 requérant moitié moins de moduli que l'Algorithme 22, l'efficacité s'en trouve globalement dédoublée.

Comparaison avec une approche standard

Donner une complexité asymptotique pour un algorithme RNS est particulièrement ardu, puisque lorsque la taille des données à représenter augmente, non seulement le nombre de moduli nécessaire croît, mais la taille β de ceux-ci doit varier également. Dans la littérature, pour comparer le coût d'une approche RNS avec une approche standard, il est usuel de donner le coût de cette approche standard en termes de multiplications de β-mots.

Nous fournissons donc dans cette partie une tentative de comparaison avec un calcul de $\mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}$ dans le système de numération positionnelle standard en base β. Pour ce faire, nous considérons que la matrice $\mathbf{R}^{-1}$ est précalculée. Plus précisément, il nous faut dans un premier temps estimer la précision suffisante à laquelle représenter les coefficients de cette matrice permettant de calculer exactement le round-off. D'une manière analogue à celle adoptée par Goldreich et al. (1997), nous écrivons $\mathbf{R}^{-1} = \widehat{\mathbf{R}} - \mathbf{E}$, où $\left|\mathbf{E}_{i,j}\right| \leqslant \beta^{-t}$ et $\beta^{-t} < \left|\widehat{\mathbf{R}}_{i,j}\right| < 1$. $\widehat{\mathbf{R}}$ est la matrice conservée en mémoire.

En notant $\mathbf{p} = \mathbf{c} - \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil \mathbf{R}$, alors la précision t doit être assez grande pour avoir :

$$\lfloor \mathbf{c}\widehat{\mathbf{R}} \rceil = \lfloor \mathbf{c}\mathbf{R}^{-1} + \mathbf{c}\mathbf{E} \rceil = \lfloor \mathbf{c}\mathbf{R}^{-1} \rceil.$$

Pour ce faire, il suffit donc que $\left\| \mathbf{p}\mathbf{R}^{-1} + \mathbf{c}\mathbf{E} \right\|_\infty < \frac{1}{2}$. Or, nous avons les inégalités suivantes :

$$\begin{aligned} \left\| \mathbf{p}\mathbf{R}^{-1} + \mathbf{c}\mathbf{E} \right\|_\infty &\leqslant \sigma\rho_{\mathbf{R}} + c_\infty \left\| \mathbf{E} \right\|_\infty \\ &\leqslant \sigma\rho_{\mathbf{R}} + c_\infty \mathcal{N} \beta^{-t}. \end{aligned}$$

L'algorithme du CVP RNS étant utilisable de manière optimale si nous avons la condition $0 < \sigma\rho_{\mathbf{R}} \leqslant \frac{1}{2} - \frac{n+2}{2(\beta-1)}$ (cf. p. 167), nous utilisons donc cette condition dans le cadre de la comparaison. Ainsi, nous déduisons qu'il suffit que t vérifie $\beta^t \geqslant \frac{2(\beta-1)c_\infty \mathcal{N}}{n+2}$. Pour simplifier l'écriture, nous avons la condition suffisante suivante, où nous approximons n par $\log_\beta(\mathcal{N}c_\infty)$:

$$\beta^t \geqslant \frac{2\beta c_\infty \mathcal{N}}{n} \Leftrightarrow t \geqslant \frac{1}{r} + 1 + n - \log_\beta(n). \tag{4.92}$$

Ensuite, nous avons déjà noté (cf. Tableau 4.4) que, comme n désigne un nombre d'entiers inférieurs à β, $\log_\beta(n) < 1$. Par soucis de clarté, nous relâchons alors légèrement la condition sur t en fixant $t = n$. En conséquence, le produit vecteur-matrice $\mathbf{c}\widehat{\mathbf{R}}$ consiste en au plus $\mathcal{N}^2$ $\texttt{Mul}_\beta\left(n, n - \log_\beta(\mathcal{N})\right)$. Chaque coefficient du vecteur résultat contient alors au plus $\log_\beta(\mathcal{N}) + n + n - \log_\beta(\mathcal{N}) = 2n$ mots.

Pour évaluer le coût du produit $\lfloor \mathbf{c}\widehat{\mathbf{R}} \rceil \mathbf{R}$, nous notons

$$R_\infty = \max_{1 \leqslant i,j \leqslant \mathcal{N}} \left|\mathbf{R}_{i,j}\right|. \tag{4.93}$$

Par suite, ce dernier produit nécessite au plus $\mathcal{N}^2$ $\texttt{Mul}_\beta\left(2n, \log_\beta(R_\infty)\right)$.

Coût	Standard	RNS
spatial ($\times \log(\beta)$ bits)	$\mathcal{N}^2 \left(n + \log_\beta(R_\infty)\right)$	$\left(\mathcal{N}^2 + \mathcal{N} + 1\right)(n+1)$ $+ \left(2\mathcal{N}^2 + \mathcal{N} + n + 1\right) \log_\beta(2\sigma)$
mult.	$\mathcal{N}^2$ $\texttt{Mul}_\beta\left(n, n - \log_\beta(\mathcal{N})\right)$ $+\mathcal{N}^2$ $\texttt{Mul}_\beta\left(2n, \log_\beta(R_\infty)\right)$	$((n+1)\mathcal{N}^2 + n\mathcal{N})$ $\texttt{MMEl}_\beta$ $+(2\mathcal{N}^2 + n\mathcal{N})$ $\texttt{MMEl}_{2\sigma}$

TABLE 4.12 – *Comparatif du coût spatial et du coût en multiplications des approches standard et RNS pour le CVP.*

Le Tableau 4.12 résume l'analyse précédente.

Comme σ est négligeable par rapport à β, le coût du CVP RNS se résume essentiellement à celui de ses multiplications élémentaires dans les canaux de $\mathcal{B} \cup \{\gamma\}$. Mais ceux-ci sont uniquement utilisés pour calculer l'arrondi $\lfloor \mathbf{c}\mathbf{R}^{-1} \rceil$. Le produit du vecteur obtenu à l'issue de cette étape par $\mathbf{R}$ intervient dans $\mathbb{Z}/m_\sigma\mathbb{Z}$.

Nous proposons donc de mesurer l'efficacité de l'approche RNS par rapport à une méthode standard en comparant le coût multiplicatif de l'étape de l'arrondi dans le pire cas, à savoir en considérant la borne de Hadamard sur le déterminant de $\mathbf{R}$ comme taille limite pour c_∞. Ainsi, la base RNS $\mathcal{B}$ est supposée vérifier la Condition (4.88). De plus, comme $\log_\beta(n) < 1$, nous pouvons de plus affirmer que $\log_\beta(\mathcal{N}) < 1$. Ainsi, le calcul de l'arrondi requiert donc $((n+1)\mathcal{N}^2 + n\mathcal{N})$ $\texttt{MMEl}_\beta$ pour la méthode RNS contre $\mathcal{N}^2$ $\texttt{Mul}_\beta(n, n)$ pour une approche multi-précision.

Ensuite, une $\texttt{MMEl}_\beta$ représentant une multiplication modulaire $|x_i \times y_i|_{m_i}$ peut dans tous les cas s'effectuer via l'Algorithme 7 de réduction de Montgomery classique. Il suffit en effet de réduit modulo m_i le produit $x_i \times y_i \times |\beta|_{m_i}$, où $x_i y_i |\beta|_{m_i} < m_i \beta^2$, afin d'éliminer le facteur de Montgomery. Ainsi, vu le coût (1.29) (p. 32) de cette réduction, une $\texttt{MMEl}_\beta$ se réalise en au plus 6 $\texttt{EMul}_\beta$. Ce coût peut être largement optimisé dans le cas de moduli pseudo Mersenne (cf. (1.7), p. 19), néanmoins nous nous contenterons de cette majoration.

r	18	20	22	24	26	28	30	32
$\mathcal{N}$	74	133	239	454	849	1536	2760	5079

TABLE 4.13 – *Dimensions limites pour lesquelles tous les β-moduli d'une base $\mathcal{B}$ vérifiant $M > \mathcal{N}^{\frac{3}{2}\mathcal{N}+1} + 1$ peuvent être premiers de type pseudo Mersenne, pour différents $r = \log_2(\beta)$.*

Pour comparer les approches, nous proposons donc d'étudier le rapport (4.94) du nombre de multiplications $\texttt{EMul}_\beta = \texttt{Mul}_\beta(1,1)$. Ainsi, $\texttt{Mul}_\beta(n, n)$ est estimé par n^{1+c} $\texttt{EMul}_\beta$, où c dépend de l'algorithme de multiplication en base β. Les Figures 4.14, 4.15 et 4.16 illustrent donc le ratio suivant :

$$\frac{n^{1+c}\mathcal{N}^2}{6((n+1)\mathcal{N}^2 + n\mathcal{N})} \tag{4.94}$$

pour respectivement $c = 1$ (multiplication quadratique), $c \sim 0,585$ (approche Karatsuba) et $c \sim 0,465$ (approche Toom-Cook).

L'approche RNS proposée pour le round-off promet ainsi des performances compétitives, couplées avec l'avantage d'une parallélisation naturelle et d'une arithmétique entière simple-précision.

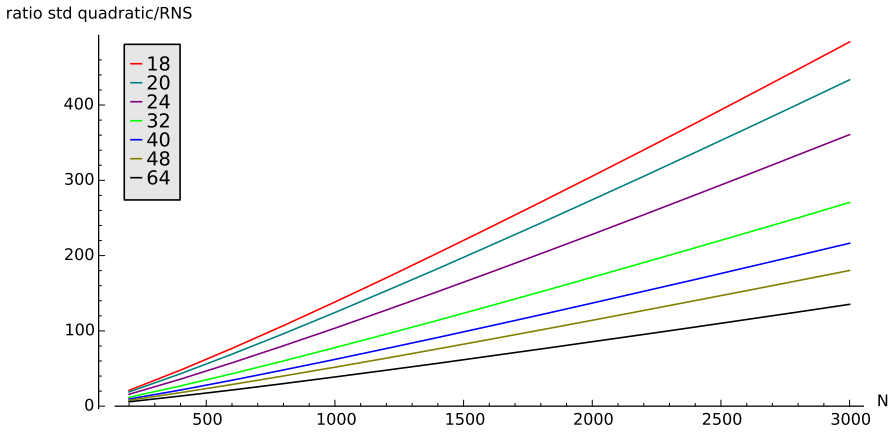


FIGURE 4.14 – *Ratio Standard quadratique/RNS du nombre de multiplications* $\texttt{EMul}_\beta$ *pour le calcul de* $\lfloor \mathbf{cR}^{-1} \rceil$.

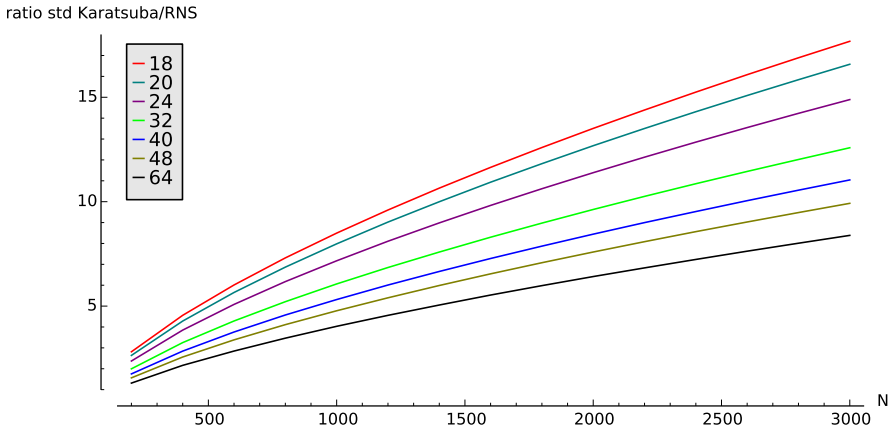


FIGURE 4.15 – *Ratio Standard Karatsuba/RNS du nombre de multiplications* $\texttt{EMul}_\beta$ *pour le calcul de* $\lfloor \mathbf{cR}^{-1} \rceil$.

Cependant, il faut souligner que dans la comparaison précédente, la phase de conversion du vecteur $\mathbf{c}$ en RNS n'est pas intégrée. Nous rappelons (cf. p. 170) que cette transformation requiert un coût supplémentaire de :

$$n \log_\beta(c_\infty)\mathcal{N} \sim n^2\mathcal{N} \ \texttt{MMEl}_\beta.$$

Même en supposant une approche efficace de la transformation binaire vers RNS par des produits de matrices optimisés, cette transformation fait alors perdre la linéarité en n du coût en nombre de multiplications. Cependant, les gains en performances estimés restent du même ordre de grandeur. Les Figures 4.17, 4.18 et 4.19 modélisent le rapport (4.95) suivant :

$$\frac{n^{1+c}\mathcal{N}^2}{6((n+1)\mathcal{N}^2 + n(n+1)\mathcal{N})} \tag{4.95}$$

Enfin, il faut noter qu'une telle comparaison ne fait pas apparaître les avantages de la parallélisation. En considérant la base $\mathcal{B}$ scindée sur $\frac{n}{\alpha}$ canaux, la transformation binaire-RNS nécessite au plus $\alpha n\mathcal{N}$ $\texttt{ETEl}_\beta$ (cf. (4.73, p. 170). Ainsi, le temps du calcul complet du vecteur le plus proche, intégrant la conversion en RNS du chiffré, est proportionnel à :

$$\left[\alpha\mathcal{N}^2 + (\alpha+1)n\mathcal{N}\right] \ \texttt{ETEl}_\beta + \left[\mathcal{N}^2 + \mathcal{N}\right] \ \texttt{ETEl}_{m_\sigma}. \tag{4.96}$$

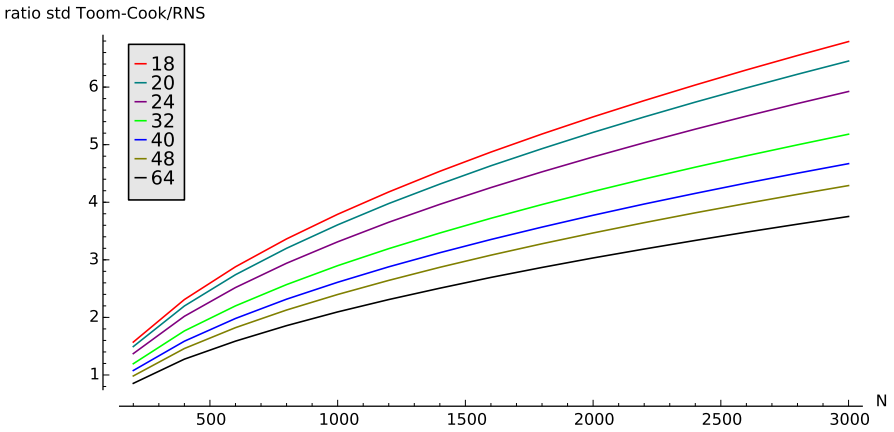


FIGURE 4.16 – *Ratio Standard Toom-Cook/RNS du nombre de multiplications* $\texttt{EMul}_\beta$ *pour le calcul de* $\lfloor \mathbf{cR}^{-1} \rceil$.

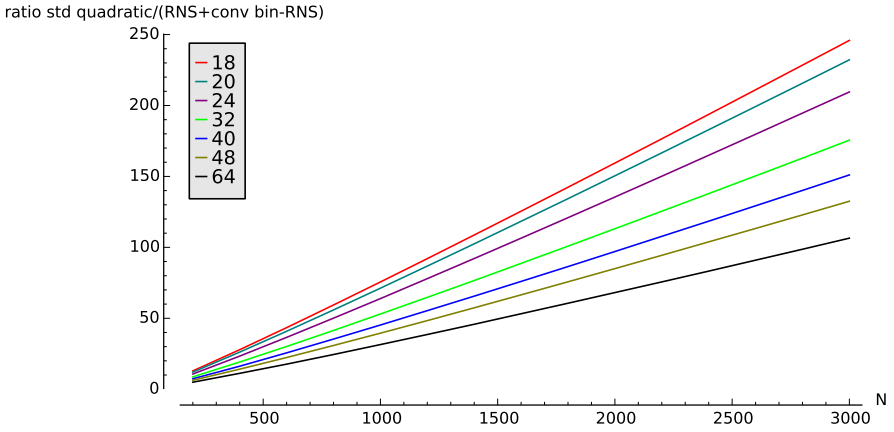


FIGURE 4.17 – *Idem Fig. 4.14 avec intégration du coût de conversion binaire vers RNS.*

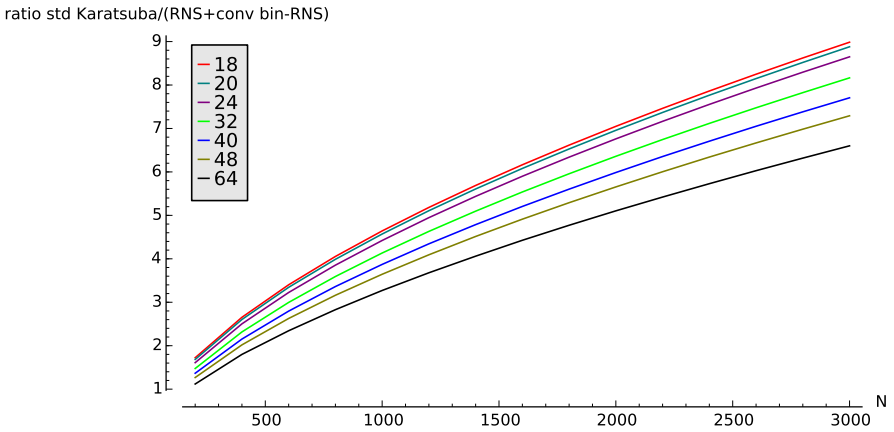


FIGURE 4.18 – *Idem Fig. 4.15 avec intégration du coût de conversion binaire vers RNS.*

Vers une implantation effective

De premiers travaux de Nabil Merkiche (Bajard, Eynard, Merkiche, et Plantard 2015) concernant l'étude de faisabilité d'implantation de l'Algorithme 23 sur FPGA (Virtex-5 et Kintex-7) ont été menés. Le principal problème relevé repose sur la taille des précalculs, lesquels limitent à de faibles dimensions ($\mathcal{N} \sim 64$), non suffisantes pour des applications cryptographique) une implan-

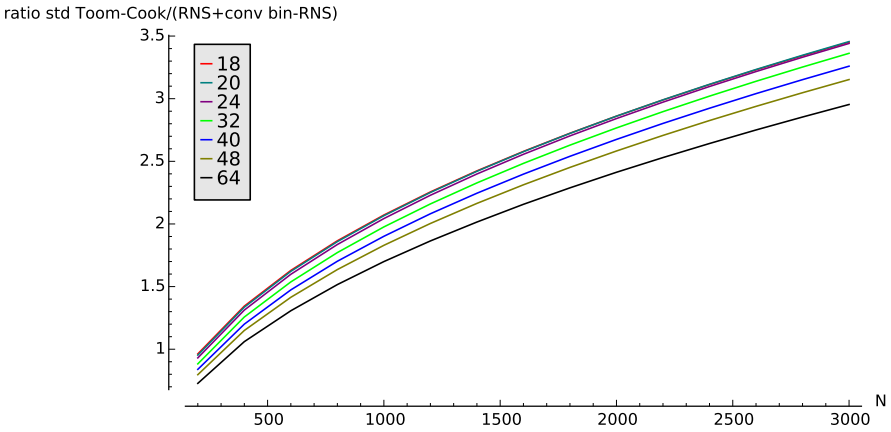


FIGURE 4.19 – *Idem Fig. 4.16 avec intégration du coût de conversion binaire vers RNS.*

tation telle quelle sur une unique plateforme FPGA. Néanmoins, la structure même de l'algorithme CVP-RNS est hautement parallélisable, principalement à cause de l'absence de conversion MRS. De ce fait, l'hypothèse d'une implantation sur un cluster de FPGA par exemple reste tout à fait réaliste sans pour autant impacter notablement les performances espérées.

Conclusion

Dans ce chapitre, nous avons proposé une optimisation d'une opération de base utilisée en cryptographie basée sur les réseaux, nommément l'algorithme du round-off de Babai. Afin de pouvoir utiliser les capacités d'accélération des RNS, il a été nécessaire d'employer une formulation adaptée à une arithmétique RNS. Le problème qui est alors apparu est celui de devoir calculer une réduction modulaire exacte. Or, la réduction modulaire RNS, basée sur la réduction de Montgomery, ne permet pas toujours d'obtenir un résultat complètement réduit, notamment lorsqu'une conversion de base efficace est utilisée.

Dans un premiers temps, ce problème a été résolu en recourant à une représentation hybride RNS-MRS autorisant l'emploi de comparaisons et, ainsi, de retrouver le résultat exact de la réduction. Cependant, de par le caractère séquentiel de la conversion RNS vers MRS, cette méthode demeure peu flexible et efficace.

Dans un second temps, une technique d'accélération a été présentée. Celle-ci découle d'une approche différente du problème considéré. En ne se focalisant plus uniquement sur la réduction modulaire mais sur l'intégralité de la reformulation du round-off, une nouvelle stratégie a permis de développer un algorithme entièrement RNS. Ceci a pour conséquence notamment de diviser grossièrement le nombre de moduli par deux, et d'offrir une très grande flexibilité en termes de parallélisation. De plus, de premières analyses de complexité montrent le potentiel d'efficacité de cette nouvelle approche face à un calcul standard en multi-précision notamment.

Conclusion générale

Le sujet de cette thèse se situe à l'intersection des domaines de la cryptographie et de l'arithmétique des ordinateurs. L'objectif poursuivi a été de contribuer à l'amélioration d'opérateurs centraux de la cryptographie asymétrique en utilisant des propriétés liées au choix de la représentation des entiers et de l'arithmétique associée. Plus précisément, les systèmes de représentation des nombres par les restes (RNS, residue number systems) ont été exploités d'une part pour améliorer l'arithmétique dans les corps finis dans un contexte d'attaques par fautes, d'autre part pour accélérer l'opération de base en cryptographie fondée sur les réseaux que constitue le round-off de Babai.

L'association des RNS et de la cryptographie asymétrique est source de nombreux résultats depuis maintenant plusieurs années. Le choix des RNS apporte des bénéfices certains aux primitives cryptographiques asymétriques. Non seulement ils permettent d'accélérer les calculs, mais ils sont aussi pourvoyeurs de solutions originales à des problèmes spécifiques à la cryptographie, comme le fait de fournir une arithmétique résistante aux fuites. La protection des primitives cryptographiques contre les attaques par injection de faute est un enjeu fondamental tant l'état de l'art a démontré à de multiples reprises la redoutable efficacité de ce genre d'attaque. Or, la nature même des RNS permet de se doter très simplement d'une capacité de détection d'erreurs via l'utilisation de redondance. Cependant, ce point de convergence entre besoins de la cryptographie et propriétés des RNS se heurte à une apparente difficile compatibilité entre les principes de fonctionnement des RNS redondants et la structure de la réduction modulaire en RNS. L'ensemble de ces constatations a donc ouvert un premier axe de recherche fondé sur les questionnements suivants :

- Est-il possible d'étendre le champ d'application des RNS redondants à la réduction modulaire ?
- Si oui, dans quelle mesure pouvons-nous garantir un moindre impact par rapport aux performances de l'état de l'art concernant l'arithmétique RNS dans les corps finis ?

Le chapitre 2 de ce mémoire propose une solution à ces problèmes. Un nouvel algorithme de réduction modulaire en RNS redondant a été présenté. Une preuve formelle garantit la détection de toute faute sur un résidu. La procédure classique de détection de faute des RNS redondants a été intégrée de manière optimale à la réduction modulaire RNS, de manière à ce qu'elle s'appuie sur une conversion de base faisant initialement partie de la structure même de l'algorithme de réduction modulaire RNS. De plus, l'ajout des canaux redondants n'entraîne aucune augmentation du temps d'exécution puisque l'intégration

de la redondance dans le flux de calcul principal bénéficie des caractéristiques de parallélisation des RNS. Le modèle de faute a également été étendu pour intégrer des spécificités liées à une implantation sur une architecture de type Cox-Rower. Le surcout en surface engendré par cette nouvelle contre-mesure se limite sensiblement à celui du Rower dédié au canal redondant.

Le troisième chapitre s'est concentré sur l'étude de la généralisation de cette nouvelle approche à un contexte plus large de fautes multiples. Nous avons ainsi montré comment l'adaptation à ce modèle de faute élargi s'effectue très simplement par l'ajout de canaux redondants supplémentaires. L'intégration de ceux-ci dans le schéma de calcul demeure encore optimale. Enfin, nous avons montré qu'au-delà de la réduction modulaire sur les entiers, ce nouvel algorithme s'accorde aussi parfaitement avec une arithmétique RNS dans les corps finis non premiers.

Finalement, ce nouveau résultat contribue à renforcer le rôle des RNS en tant qu'arithmétique adaptée à la cryptographie asymétrique, en achevant de concilier RNS redondants et réduction modulaire.

Le second axe de recherche exploré dans cette thèse, et développé dans le quatrième chapitre, concerne l'optimisation arithmétique d'une opération importante en cryptographie basée sur les réseaux, le round-off. Celle-ci permet de calculer un vecteur d'un réseau proche d'un point donné $\mathbf{c}$ de l'espace. La première étape a consisté en une reformulation complète du contexte adaptée à une arithmétique sur les entiers. Cette approche permet de réduire une partie des calculs dans un corps dont la taille est proportionnelle à la distance entre $\mathbf{c}$ et le réseau. Or, celle-ci s'avère faible dans le contexte des cryptosystèmes de type GGH. Par conséquent, l'arithmétique modulaire permet une accélération évidente des calculs. Cependant, le problème central soulevé par l'étape de reformulation est l'apparition d'une réduction modulaire, délicate à traiter en RNS. Nos efforts se sont alors particulièrement concentrés sur ce point précis. Un premier travail a permis d'aboutir à un algorithme du round-off utilisant une représentation mixte des nombres en RNS et MRS (mixed radix system). Ensuite, une propriété géométrique de la base du réseau utilisée dans la round-off a été exploitée pour introduire une technique d'accélération. Lorsque celle-ci est applicable à la base donnée, elle autorise alors un calcul complet du vecteur proche par round-off via une représentation entièrement RNS des nombres. La complexité de l'algorithme qui résulte de cette approche indique des performances prometteuses face à une approche standard multi-précision. De plus, le caractère hautement parallélisable de l'algorithme proposé autorise une importante flexibilité, ce qui permet d'envisager des implantations futures compétitives.

Perspectives

Contrairement aux méthodes par infection, la solution de protection contre les injections de faute proposée se base sur une approche par détection qui présente la faiblesse intrinsèque de pouvoir être contrée par une attaque par faute bien ciblée. Or, les RNS peuvent potentiellement apporter des solutions efficaces pour une approche complémentaire de protection par infection. Les propriétés d'aléa offertes par ces systèmes ont par exemple déjà été exploitées par Bajard et al. (2004) pour un masquage des données, et par Bajard et

Hördegen (2009) pour créer un générateur pseudo-aléatoire. Les perspectives de protection des primitives cryptographiques au niveau arithmétique restent donc nombreuses. De plus, l'approche arithmétique ne se limite pas aux seuls RNS, et peut être élargie à d'autres systèmes de représentation, comme les systèmes modulaires adaptés de Plantard (2005), qui disposent d'une arithmétique efficace et de propriétés de redondance. Les axes de recherche à ce niveau sont donc encore nombreux.

La suite logique des résultats concernant l'accélération des cryptosystèmes de type GGH par le biais des RNS passe bien sûr par l'implantation logicielle et matérielle de nos propositions. Si les problèmes de taille des clefs restent toujours présents, la flexibilité permise par notre approche en terme de parallélisation autorise une grande latitude puisque la gestion des données précalculées se répartit également indépendamment entre les canaux RNS. Ainsi, des solutions multiples et variées sont envisageables : CPU multi-core, GPU, clusters de FPGA, etc. Enfin, si la recherche en cryptographie basée sur les réseaux est un domaine actuellement très actif, les problèmes de complexité spatiale et temporelle demeurent un point critique pour la plupart des différentes techniques développées dans ce domaine. Nul doute que l'approche arithmétique de ces problématiques est une perspective qui mérite d'être explorée en profondeur.

Annexes

A

A.1 De la détection des fautes multiples dans le cas très général

Nous traitons le cas des RNS redondants pour la détection des fautes multiples dans le cas très général où aucune hypothèse *a priori* n'est faite à propos des moduli redondants, pour les corps finis $\mathbb{F}_p$ et $\mathbb{F}_{p^s}$. Nous donnons donc dans ce contexte les formulations et preuves de théorèmes généraux dont les Théorèmes 3.1 (p. 90), 3.2 (p. 91), 3.5 (p. 110) et 3.6 (p. 111) sont des cas particuliers.

A.1.1 Dans un corps fini $\mathbb{F}_p$

Nous nous plaçons dans le contexte de la Section 3.1. Les données sont une base RNS $\mathcal{B} = \{m_1, \ldots, m_n\}$ et un modulus redondant M_R. Les Propositions 3.1 et 3.2 concernant les effets d'une faute multiple sur l'entier représenté par les résidus erronés permettent de montrer le théorème suivant.

Théorème A.1 *Pour détecter toute d-faute, $d \leqslant n$, alors il est nécessaire que M_R vérifie la condition suivante :*

$$\forall\, (i_1, \ldots, i_d) \subset [\![1, n]\!], \quad M_R \geqslant \prod_{i \in \mathcal{I}_d} m_i \times \left(M_R \wedge M_{\mathcal{I}_d}\right). \tag{A.1}$$

Démonstration. Le début de la preuve est identique à celui de la preuve du Théorème 3.1. En en reprenant les notations, nous avons ainsi que M_R divise $a_{\mathcal{I}_d} M_{\mathcal{I}_d}$. Par suite, $\frac{M_R}{M_R \wedge M_{\mathcal{I}_d}}$ divise $a_{\mathcal{I}_d}$, et donc $\frac{M_R}{M_R \wedge M_{\mathcal{I}_d}} < \prod\limits_{i \in \mathcal{I}_d} m_i$, ce qui achève la preuve. □

Théorème fondamental de détection pratique d'erreurs multiples

Deux lemmes préliminaires aideront à la preuve du théorème principal. Le premier est un résultat général d'arithmétique qui sert pour la preuve du second lemme.

Lemme A.1 *Pour tout triplet d'entiers $(a, b, c) \in \mathbb{Z}^3$, $(ab) \wedge c$ divise $(a \wedge c) \times (b \wedge c)$.*

Démonstration. Sans perte de généralité, a, b et c sont supposés positifs. Les décompositions uniques en produit de nombres premiers de a et b sont notées $a = \prod\limits_{p \in \mathcal{P}} p^{u_p}$, $b = \prod\limits_{p \in \mathcal{P}} p^{v_p}$. Ainsi,

$$\begin{cases} (ab) \wedge c = \left(\prod\limits_{p \in \mathcal{P}} p^{u_p + v_p}\right) \wedge c = \prod\limits_{p \in \mathcal{P}} \left(p^{u_p + v_p} \wedge c\right), \\ (a \wedge c) \times (b \wedge c) = \prod\limits_{p \in \mathcal{P}} \left(p^{u_p} \wedge c\right) \times \left(p^{v_p} \wedge c\right). \end{cases} \tag{A.2}$$

Il suffit donc de montrer que pour tout $(u, v) \in \mathbb{N}^2$ et tout $p \in \mathcal{P}$, $(p^{u+v} \wedge c)$ divise $(p^u \wedge c) \times (p^v \wedge c)$. $u + v + 1$ cas sont possibles : si, pour $i \in [\![0, u + v]\!]$, p^i divise c et p^{i+1} ne divise pas c. Sans perte de généralité, il est supposé que $u \leqslant v$. Alors :

- si $i \in [\![0, u]\!]$, $(p^{u+v} \wedge c) = p^i$, $(p^u \wedge c) \times (p^v \wedge c) = p^{2i}$,

- si $i \in [\![u+1, v]\!]$, $(p^{u+v} \wedge c) = p^i$, $(p^u \wedge c) \times (p^v \wedge c) = p^{u+i}$,
- si $i \in [\![v+1, u+v]\!]$, $(p^{u+v} \wedge c) = p^i$, $(p^u \wedge c) \times (p^v \wedge c) = p^{u+v}$.

Par conséquent $(p^{u+v} \wedge c)$ divise bien $(p^u \wedge c) \times (p^v \wedge c)$. Par les Équations (A.2), le résultat attendu vient alors immédiatement. □

Lemme A.2 *Soit $\mathcal{B} = \{m_1, \ldots, m_n\}$ une base RNS et $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$ un ensemble de k entiers, avec $k \leqslant n$, vérifiant les hypothèses suivantes :*

$$\forall (i,j) \in [\![1,k]\!] \times [\![1,n]\!],\ m_{R,i} \geqslant m_j \times \left(m_{R,i} \wedge M_j\right), \tag{A.3}$$

et si $k \geqslant 2$,

$$\begin{cases} \forall (z_1, z_2, i_1, i_2) \in [\![1,k]\!]^2 \times [\![1,n]\!]^2, \\ \dfrac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \geqslant m_{i_1} m_{i_2} \times \left(\dfrac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \wedge M_{i_1 i_2}\right). \end{cases} \tag{A.4}$$

Alors pour tout $d \in [\![2,k]\!]$ et tous ensembles de d indices $\mathcal{I}_d \subset [\![1,n]\!]$ et $\mathcal{Z}_d \subset [\![1,k]\!]$,

$$\frac{\prod\limits_{z \in \mathcal{Z}_d} m_{R,z}}{\operatorname{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_d\}} \geqslant \prod_{i \in \mathcal{I}_d} m_i \times \left(\frac{\prod\limits_{z \in \mathcal{Z}_d} m_{R,z}}{\operatorname{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_d\}} \wedge M_{\mathcal{I}_d}\right) \tag{A.5}$$

où $M_{\mathcal{I}_d}$ désigne le produit $\prod\limits_{i \in [\![1,n]\!] \setminus \mathcal{I}_d} m_i = \dfrac{M}{m_{i_1} \ldots m_{i_d}}$.

Démonstration. La preuve se fait par récurrence sur d. Les cas $d = 1$ et $d = 2$ sont simplement les Hypothèses (A.3) et (A.4) respectivement. Soit donc $d \in [\![3, k-1]\!]$. La Propriété (A.5) est supposée vérifiée pour tout entier $t \in [\![1, d]\!]$. Il s'agit donc de la prouver pour $d+1$.

Soit deux ensembles de $d+1$ indices $\mathcal{I}_{d+1} = \{i_1, \ldots, i_{d+1}\}$ et $\mathcal{Z}_{d+1} = \{z_1, \ldots, z_{d+1}\}$. Afin d'alléger les écritures, les notations suivantes sont introduites :

$$\begin{cases} \mathcal{I}_d = \mathcal{I}_{d+1} \backslash \{i_{d+1}\} \\ \mathcal{Z}_d = \mathcal{Z}_{d+1} \backslash \{z_{d+1}\} \\ p_d = \operatorname{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_d\} \\ p_{d+1} = \operatorname{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_{d+1}\} \\ R_d = \prod\limits_{z \in \mathcal{Z}_d} m_{R,z} \\ R_{d+1} = \prod\limits_{z \in \mathcal{Z}_{d+1}} m_{R,z} \end{cases} \tag{A.6}$$

Il s'agit donc de montrer l'inégalité suivante :

$$\frac{R_{d+1}}{p_{d+1}} \geqslant \prod_{i \in \mathcal{I}_{d+1}} m_i \times \left(\frac{R_{d+1}}{p_{d+1}} \wedge M_{\mathcal{I}_{d+1}}\right). \tag{A.7}$$

En appliquant l'hypothèse de récurrence (A.5) avec les indices $\mathcal{I}_d$ et $\mathcal{Z}_d$, il vient :

$$R_d \geqslant p_d \times \prod_{i \in \mathcal{I}_d} m_i \times \left(\frac{R_d}{p_d} \wedge M_{\mathcal{I}_d}\right). \tag{A.8}$$

De même, l'hypothèse (A.3) utilisée avec $m_{i_{d+1}}$ et $m_{R,z_{d+1}}$ donne :

$$m_{R,z_{d+1}} \geqslant m_{i_{d+1}} \times \left(m_{R,z_{d+1}} \wedge M_{i_{d+1}}\right). \tag{A.9}$$

En rassemblant les Inégalités (A.8) et (A.9), il vient l'inégalité suivante :

$$R_{d+1} \geqslant p_d \times \prod_{i \in \mathcal{I}_{d+1}} m_i \times \left(\frac{R_d}{p_d} \wedge M_{\mathcal{I}_d}\right) \times \left(m_{R,z_{d+1}} \wedge M_{i_{d+1}}\right). \tag{A.10}$$

Les prochaines étapes vont consister à démontrer l'inégalité suivante :

$$p_d \times \left(\frac{R_d}{p_d} \wedge M_{\mathcal{I}_d}\right) \times \left(m_{R,z_{d+1}} \wedge M_{i_{d+1}}\right) \geqslant p_{d+1} \times \left(\frac{R_{d+1}}{p_{d+1}} \wedge M_{\mathcal{I}_{d+1}}\right). \tag{A.11}$$

Une fois ceci fait, l'Inégalité (A.7) résulte simplement de la combinaison des Inégalités (A.10) et (A.11).

Comme p_{d+1} divise p_d, il existe un entier a tel que $a \times p_{d+1} = p_d$. En utilisant le résultat du Lemme A.1 et le fait que p_{d+1} divise R_d, il vient :

$$\begin{aligned} p_{d+1} \times \left(\frac{R_d}{p_{d+1}} \wedge M_{\mathcal{I}_d}\right) &= p_{d+1} \times \left(\frac{a \times R_d}{p_d} \wedge M_{\mathcal{I}_d}\right) \\ &\leqslant p_{d+1} \times \left(a \wedge M_{\mathcal{I}_d}\right) \times \left(\frac{R_d}{p_d} \wedge M_{\mathcal{I}_d}\right) \\ &\leqslant p_{d+1} \times a \times \left(\frac{R_d}{p_d} \wedge M_{\mathcal{I}_d}\right) \\ &= p_d \times \left(\frac{R_d}{p_d} \wedge M_{\mathcal{I}_d}\right). \end{aligned} \tag{A.12}$$

De plus, en utilisant le Lemme A.1 précédent, l'Inégalité (A.12) et le fait que $M_{\mathcal{I}_{d+1}} < M_{i_{d+1}}$, alors :

$$\begin{aligned} p_{d+1} \times \left(\frac{R_{d+1}}{p_{d+1}} \wedge M_{\mathcal{I}_{d+1}}\right) &\leqslant p_{d+1} \times \left(\frac{R_d}{p_{d+1}} \wedge M_{\mathcal{I}_{d+1}}\right) \times \left(m_{R,z_{d+1}} \wedge M_{\mathcal{I}_{d+1}}\right) \\ &\leqslant p_d \times \left(\frac{R_d}{p_d} \wedge M_{\mathcal{I}_d}\right) \times \left(m_{R,z_{d+1}} \wedge M_{i_{d+1}}\right). \end{aligned} \tag{A.13}$$

L'Inégalité (A.11) est ainsi montrée, et la preuve du lemme achevée. □

Nous pouvons désormais énoncer et prouver le théorème général donnant les conditions sur l'ensemble de moduli redondants $\mathcal{B}_R$ pour la détection des fautes multiples basée sur la procédure DetectMultErr.

Théorème A.2 *Soit $\mathcal{B} = \{m_1, \ldots, m_n\}$ une base RNS et $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$ k entiers avec $k \in [\![2, n]\!]$. Soit $x \in [\![0, M[\![$ donné par ses résidus $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R})$.*

Quelle que soit la conversion utilisée dans la procédure de détection DetectMultErr, pour tout $d \in [\![k+1, n+k]\!]$ il existe une d-faute non détectable.

1. *L'ensemble de toutes les d-fautes affectant $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R})$ pour tout $d \leqslant k$ est détectable par la Procédure TestCoherence ou la Procédure DetectMultErr couplée avec la conversion Bex_{mrs} si, et seulement si,*

$$\begin{cases} \forall (i,j) \in [\![1,k]\!] \times [\![1,n]\!],\ m_{R,i} \geqslant m_j \times \left(m_{R,i} \wedge M_j\right), \\ \forall\, (z_1, z_2, i_1, i_2) \in [\![1,k]\!]^2 \times [\![1,n]\!]^2, \\ \dfrac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \geqslant m_{i_1} m_{i_2} \times \left(\dfrac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \wedge M_{i_1 i_2}\right). \end{cases} \tag{A.14}$$

2. *Si un modulus supplémentaire m_{sk} est adjoint à la base $\mathcal{B}$ et* Bex_{sk} *est utilisé par la Procédure DetectMultErr, une condition suffisante est donnée par :*

$$\begin{cases} \forall (i,j) \in [\![1,k]\!] \times [\![1,n]\!], \\ m_{R,i} \geqslant m_j \times \left(m_{R,i} \wedge m_{sk} M_j\right) \text{ et } m_{R,i} \geqslant m_{sk} \times (m_{R,i} \wedge M), \\ \forall\, (z_1,z_2,i_1,i_2) \in [\![1,k]\!]^2 \times [\![1,n]\!]^2, \\ \dfrac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \geqslant m_{i_1} m_{i_2} \times \left(\dfrac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \wedge m_{sk} M_{i_1 i_2}\right), \\ \dfrac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \geqslant m_{i_1} m_{sk} \times \left(\dfrac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \wedge M_{i_1}\right). \end{cases} \tag{A.15}$$

3. *Si toute valeur x dont les résidus sont intègres est supposée être dans l'intervalle $[\![0, (1-\alpha_{kw})\, M[\![$ et si* Bex_{kwc} *est utilisé pour la Procédure DetectMultErr, alors l'ensemble de toutes les d-fautes avec $d \leqslant k$ est détectable si les Conditions (A.14) du cas* Bex_{mrs} *sont vérifiées.*

Démonstration. Soit $(\overline{\boldsymbol{x}}_{\mathcal{B}}, \overline{\boldsymbol{x}}_{\mathcal{B}_R})$ les résidus issus de ceux de x affectés par une d-faute, et $\overline{x} = \varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) \in [\![0, M[\![$. Nous rappelons que le test de cohérence est la vérification de l'égalité $\text{Bex}(\overline{\boldsymbol{x}}_{\mathcal{B}}) \stackrel{?}{=} \overline{\boldsymbol{x}}_{\mathcal{B}_R}$ où Bex est une conversion de la base $\mathcal{B}$ vers la base $\mathcal{B}_R$.

Pour $d \in [\![k+1, n+k]\!]$, il est possible de construire une d-faute non détectable. Les résidus $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R})$ sont par hypothèse ceux d'un entier x quelconque de l'intervalle dynamique de $\mathcal{B}$. Soit une faute affectant $d-k$ des résidus principaux $\boldsymbol{x}_{\mathcal{B}}$. Notant $\overline{x} = \text{Bex}\,(\mathcal{B}, \mathcal{B}_R, \overline{\boldsymbol{x}}_{\mathcal{B}}) \in [\![0, M[\![$, il suffit de considérer la faute modifiant les résidus redondants de la manière suivante : $\overline{x}_{R,i} = |\overline{x}|_{m_{R,i}}$ pour tout $i \in [\![1,k]\!]$. La capacité de détection du RNS redondant est donc inférieure ou égale à k.

1. • Preuve de la suffisance.

   La suffisance peut être montrée par récurrence sur $d \in [\![1,k]\!]$. Le cas $d = 1$ est prouvé par le Théorème 2.3. Soit $d \in [\![1, k-1]\!]$. Par hypothèse, pour tout $t \in [\![1,d]\!]$, toute t-faute affectant $(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}_R})$ est détectée par le test de cohérence.

   Soit alors une $(d+1)$-faute affectant d_n résidus principaux et d_k résidus redondants. Ainsi $d_n + d_k = d+1$, $d_n \leqslant n$ et $d_k \leqslant k$. Il est clair que si $d_n \leqslant d$, alors l'hypothèse de récurrence sur les t-fautes avec $t \leqslant d$ appliquée avec l'ensemble des résidus redondants privé d'un résidu redondant erroné permet de conclure à la détection.

   Il reste donc à considérer le cas où $d_n = d+1$. Montrons qu'une telle d-faute est nécessairement détectée sous les Hypothèses 3.3. Les résidus principaux erronés sont supposés être indexés par $\mathcal{I}_{d+1}$. Alors l'Équation (3.1) de la Proposition 3.1 p(. 89) donne l'existence d'un entier $a_{\mathcal{I}_{d+1}}$ tel que $0 < \left|a_{\mathcal{I}_{d+1}}\right| < \prod_{i \in \mathcal{I}_{d+1}} m_i$ et vérifiant $\overline{x} - x = a_{\mathcal{I}_{d+1}} M_{\mathcal{I}_{d+1}} \neq 0$. Comme $\overline{x} = \varphi_{\mathcal{B}_R}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = \text{Bex}_{mrs}(\overline{\boldsymbol{x}}_{\mathcal{B}})$, la non-détection d'une telle faute par la procédure DetectMultErr impliquerait que $\overline{x} - x \equiv 0 \bmod m_{R,z}$ pour tout $m_{R,z} \in \mathcal{B}_R$, et donc en particulier :

$$\overline{x} - x \equiv 0 \bmod \frac{\prod_{z \in \mathcal{Z}_{d+1}} m_{R,z}}{\text{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_{d+1}\}}$$

pour tout ensemble de $d+1$ indices $\mathcal{Z}_{d+1} \subseteq [\![1,k]\!]$. Nous reprenons pour la suite les notations (A.6) (p. 194), à savoir notamment $p_{d+1} = \text{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_{d+1}\}$ et $R_{d+1} = \prod_{z\in\mathcal{Z}_{d+1}} m_{R,z}$. Par conséquent, il existerait au moins un ensemble de $d+1$ indices $\mathcal{Z}_{d+1}$ pour lesquels nous avons :

$$\frac{\dfrac{R_{d+1}}{p_{d+1}}}{\dfrac{R_{d+1}}{p_{d+1}} \wedge M_{\mathcal{I}_{d+1}}} \text{ divise } a_{\mathcal{I}_{d+1}}$$

et donc :

$$\frac{R_{d+1}}{p_{d+1}} < \prod_{i\in\mathcal{I}_{d+1}} m_i \times \left(\frac{R_{d+1}}{p_{d+1}} \wedge M_{\mathcal{I}_{d+1}}\right).$$

Or, par le Lemme A.2, cela contredirait l'Hypothèse (3.3).

- Prouvons la nécessité par contraposition, avec construction de contre-exemples.
  Soit $(z,i) \in [\![1,k]\!] \times [\![1,n]\!]$ tel que $m_{R,z} < m_i \times (m_{R,z} \wedge M_i)$, et $x = 0$. Alors il est possible de construire un ensemble de résidus erroné $(\overline{\boldsymbol{x}}_{\mathcal{B}}, \overline{\boldsymbol{x}}_{\mathcal{B}_R})$ contenant au plus k fautes et qui passera avec succès le test de cohérence. Soit $\overline{x}_t = 0$ pour tout $t \in [\![1,n]\!]\backslash\{i\}$, et $\overline{x}_i = \left|\dfrac{m_{R,z}}{m_{R,z} \wedge M_i} \times M_i\right|_{m_i}$. Alors en notant toujours $\overline{x}$ l'entier $\overline{x} = \varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}})$ de $[\![0, M[\![$ et en notant $a_i = \left|\overline{x}_i M_i^{-1}\right|_{m_i}$, nous avons :

$$\begin{aligned}
\overline{x} &= a_i M_i \\
&= \left|\frac{m_{R,z}}{m_{R,i} \wedge M_i} \times M_i \times M_i^{-1}\right|_{m_i} \times M_i \\
&= \frac{m_{R,z}}{m_{R,z} \wedge M_i} \times M_i \\
&= m_{R,z} \times \frac{M_i}{m_{R,z} \wedge M_i}.
\end{aligned}$$

  Nous supposons alors que les résidus redondants erronés sont $\overline{x}_{R,z} = 0$ et $\overline{x}_{R,t} = |a_i M_i|_{m_{R,t}}$ pour tout $t \neq z$. Par conséquent,

$$\varphi_{\mathcal{B}_R} \circ \varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = \text{Bex}_{mrs}(\mathcal{B}, \mathcal{B}_R, \overline{\boldsymbol{x}}_{\mathcal{B}}) = \overline{\boldsymbol{x}}_{\mathcal{B}_R}.$$

  Ceci implique que le test de cohérence ne détecte pas la faute multiple donnée.
  Soit maintenant un quadruplet d'indices $(z_1, z_2, i_1, i_2) \in [\![1,k]\!]^2 \times [\![1,n]\!]^2$ pour lequel :

$$\frac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} < m_{i_1} m_{i_2} \times \left(\frac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \wedge M_{i_1 i_2}\right).$$

  Notons $R = \frac{m_{R,z_1} m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}}$. Il est possible de construire une d-faute non détectée avec $d \leqslant k$ affectant la valeur $x = 0$. Pour ce faire, cette faute

affecte les résidus $\boldsymbol{x}_{\mathcal{B}}$ d'indices i_1 et i_2, et les résidus redondants $\boldsymbol{x}_{\mathcal{B}_R}$ d'indices $[\![1,k]\!]\backslash\{z_1,z_2\}$ de la manière suivante :

$$\begin{cases} \overline{x}_{i_1} = \left|\dfrac{R \times M_{i_1 i_2}}{R \wedge M_{i_1 i_2}}\right|_{m_{i_1}} \\ \overline{x}_{i_2} = \left|\dfrac{R \times M_{i_1 i_2}}{R \wedge M_{i_1 i_2}}\right|_{m_{i_1}} \\ \forall z \in [\![1,k]\!]\backslash\{z_1,z_2\},\ \overline{x}_{R,z} = \left|\dfrac{R \times M_{i_1 i_2}}{R \wedge M_{i_1 i_2}}\right|_{m_{R,z}} \end{cases}$$

Par hypothèse, l'entier $\frac{R}{R \wedge M_{i_1 i_2}}$ est dans l'intervalle $]\!]0, m_{i_1} m_{i_2}[\![$. Cet entier est donc complètement représentable par ses résidus dans la base RNS à deux moduli $\{m_{i_1}, m_{i_2}\}$, et donc, par la Formule (1.15) de la preuve constructive du théorème des restes chinois, à savoir dans le cas présent :

$$\frac{R}{R \wedge M_{i_1 i_2}} = \left|\frac{R \times m_{i_2}^{-1}}{R \wedge M_{i_1 i_2}}\right|_{m_{i_1}} m_{i_2} + \left|\frac{R \times m_{i_1}^{-1}}{R \wedge M_{i_1 i_2}}\right|_{m_{i_2}} m_{i_1} - \delta m_{i_1} m_{i_2} \quad \text{(A.16)}$$

où $\delta \in \{0,1\}$ permet la réduction dans $]\!]0, m_{i_1} m_{i_2}[\![$. Ainsi, en notant $\overline{x} = \varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = \text{Bex}_{mrs}(\mathcal{B}, \mathcal{B}_R, \overline{\boldsymbol{x}}_{\mathcal{B}})$, alors par l'Équation (A.16), et comme par hypothèse $M_{i_1 i_2} \times \frac{R}{R \wedge M_{i_1 i_2}} < M$, nous avons :

$$\begin{aligned} \overline{x} &= \left(\left|\frac{R \times m_{i_2}^{-1}}{R \wedge M_{i_1 i_2}}\right|_{m_{i_1}} M_{i_1} + \left|\frac{R \times m_{i_1}^{-1}}{R \wedge M_{i_1 i_2}}\right|_{m_{i_2}} M_{i_2}\right) \bmod M \\ &= \left[M_{i_1 i_2}\left(\left|\frac{R \times m_{i_2}^{-1}}{R \wedge M_{i_1 i_2}}\right|_{m_{i_1}} m_{i_2} + \left|\frac{R \times m_{i_1}^{-1}}{R \wedge M_{i_1 i_2}}\right|_{m_{i_2}} m_{i_1}\right)\right] \bmod M \\ &= \left[M_{i_1 i_2}\left(\frac{R}{R \wedge M_{i_1 i_2}} + \delta m_{i_1} m_{i_2}\right)\right] \bmod M \\ &= \left(\frac{R \times M_{i_1 i_2}}{R \wedge M_{i_1 i_2}}\right) \bmod M \\ &= R \times \frac{M_{i_1 i_2}}{R \wedge M_{i_1 i_2}}. \end{aligned}$$

Par conséquent, nous avons en particulier $\overline{x} \equiv 0 \bmod m_{R,z_1}$ et $\overline{x} \equiv 0 \bmod m_{R,z_2}$. Et de par la forme donnée des erreurs sur les autres résidus redondants de $x = 0$, il vient donc :

$$\varphi_{\mathcal{B}_R}(\overline{x}) = \varphi_{\mathcal{B}_R} \circ \varphi_{\mathcal{B}}^{-1}(\overline{\boldsymbol{x}}_{\mathcal{B}}) = \overline{\boldsymbol{x}}_{\mathcal{B}_R}.$$

La faute n'est donc pas détectée.

2. La suffisance de la condition se prouve une fois de plus par récurrence. Le cas $d = 1$ est donné par le Théorème 2.3. Pour les mêmes raisons que précédemment, seul le cas où $d \in [\![2,k]\!]$ fautes affectant les résidus principaux reste à vérifier en détails.

Dans un premier temps, le résidu $\overline{x}_{sk}$ utilisé pour calculer $\kappa_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}})$ est supposé intègre. Nous avons donc :

$$\overline{x} = x + a_{\mathcal{I}_d} M_{\mathcal{I}_d} \in [\![0, M[\![$$

pour des indices $\mathcal{I}_d = \{i_1, \ldots, i_d\} \subset [\![1, n]\!]$ et un entier $0 < \left|a_{\mathcal{I}_d}\right| < \prod_{i \in \mathcal{I}_d} m_i$ (cf. Prop. 3.1). Alors la conversion donne :

$$\mathrm{Bex}_{sk}(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}, x_{sk}) = \overline{x} + bM$$

où b est un entier tel que $|b| < m_{sk}$ et vérifiant également :

$$b \equiv -a_{\mathcal{I}_d} M_{\mathcal{I}_d} \bmod m_{sk}.$$

En particulier, ceci implique que $\left|\mathrm{Bex}_{sk}\left(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}, x_{sk}\right) - x\right|_{m_{sk}} = 0$. Il faut souligner que $\overline{x} + bM \neq x$. En effet, il suffit de voir que, par hypothèse, $\overline{x} + bM \not\equiv x \bmod \prod_{i \in \mathcal{I}_d} m_i$. Finalement nous avons :

$$0 < \left|\mathrm{Bex}_{sk}\left(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}, x_{sk}\right) - x\right| = \left|a_{\mathcal{I}_d} M_{\mathcal{I}_d} + bM\right| = c m_{sk} M_{\mathcal{I}_d}$$

avec c un entier tel que $0 < c < \prod_{i \in \mathcal{I}_d} m_i$. Si une telle faute n'est pas détectée, cela signifierait que $c m_{sk} M_{\mathcal{I}_d} \equiv 0 \bmod m_{R,z}$ pour tout $m_{R,z} \in \mathcal{B}_R$, et donc en particulier, pour tout ensemble de d indices $\mathcal{Z}_d \subseteq [\![1, k]\!]$,

$$c m_{sk} M_{\mathcal{I}_d} \equiv 0 \bmod \frac{R_d}{p_d}.$$

Ainsi, pour un ensemble $\mathcal{Z}_d$ fixé, c étant non nul il viendrait alors :

$$\frac{\dfrac{R_p}{d_p}}{\dfrac{R_p}{d_p} \wedge m_{sk} M_{\mathcal{I}_d}} \text{ divise } c \Rightarrow \frac{R_p}{d_p} < \prod_{i \in \mathcal{I}_d} m_i \times \left(\frac{R_p}{d_p} \wedge m_{sk} M_{\mathcal{I}_d}\right).$$

Le Lemme A.2 impliquerait dans ce cas la contradiction de l'Hypothèse (3.4).

Dans un deuxième temps, le résidu x_{sk} est supposé être erroné. Soit donc $\overline{x}_{sk} \neq x_{sk}$, et $\mathcal{I}_{d-1} = \{i_1, \ldots, i_{d-1}\}$ les autres indices des résidus principaux erronés. Ceci implique que nous pouvons écrire :

$$\overline{x} = \mathrm{Bex}_{sk}(\mathcal{B} \cup \{m_{sk}\}, \overline{\boldsymbol{x}}_{\mathcal{B}}, \overline{x}_{sk}) = x + a_{\mathcal{I}_{d-1}} M_{\mathcal{I}_{d-1}} - bM$$

avec $0 < \left|a_{\mathcal{I}_{d-1}}\right| < \prod_{i \in \mathcal{I}_{d-1}} m_i$ et $0 \leqslant b \leqslant m_{sk} - 1$. De plus, nous avons alors :

$$\left|a_{\mathcal{I}_{d-1}} - b \times \prod_{i \in \mathcal{I}_{d-1}} m_i\right| < m_{sk} \times \prod_{i \in \mathcal{I}_{d-1}} m_i.$$

Mais nous avons aussi $a_{\mathcal{I}_{d-1}} - b \times \prod_{i \in \mathcal{I}_{d-1}} m_i \neq 0$. En effet, comme par exemple $x_{sk} \not\equiv \overline{x}_{sk} \equiv x_{sk} + a_{\mathcal{I}_{d-1}} - b \times \prod_{i \in \mathcal{I}_{d-1}} m_i \bmod m_{sk}$, alors $a_{\mathcal{I}_{d-1}} - b \times$

$\prod_{i\in\mathcal{I}_{d-1}} m_i \not\equiv 0 \bmod m_{sk}$. Le Lemme A.2 appliqué avec les Hypothèses (3.4) du théorème que nous sommes en train de prouver implique que pour tout ensemble de d indices $\mathcal{Z}_d \subseteq [\![1,k]\!]$:

$$\frac{R_d}{p_d} \geqslant m_{sk} \prod_{i\in\mathcal{I}_{d-1}} m_i \times \left(\frac{R_d}{p_d} \wedge M_{\mathcal{I}_{d-1}}\right).$$

Ainsi, de la même manière que pour le cas où x_{sk} était supposé intègre, cela implique que :

$$a_{\mathcal{I}_{d-1}} - b \times \prod_{i\in\mathcal{I}_{d-1}} m_i \not\equiv 0 \bmod \frac{R_d}{p_d}.$$

D'où la détection de la faute multiple.

3. Comme vu précédemment dans le cas Bex_{mrs}, $\overline{x} = x + a_{\mathcal{I}_d} M_{\mathcal{I}_d}$ avec $0 < \left|a_{\mathcal{I}_d}\right| < \prod_{i\in\mathcal{I}_d} m_i$. Il vient alors :

$$\text{Bex}_{kwc}(\mathcal{B}, \overline{\boldsymbol{x}}_{\mathcal{B}}) - x = \overline{x} - \delta M - x = a_{\mathcal{I}_d} M_{\mathcal{I}_d} - \delta M$$

avec $\delta \in \{0,1\}$. De plus, vu la Remarque 1.9 (p. 29), et comme par hypothèse $x \in [\![0, (1-\alpha_{kw})\, M[\![$, les implications suivantes sont vérifiées :

$$\delta = 1 \Rightarrow \overline{x} = \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{x}}_{\mathcal{B}}\right) \in [\![(1-\alpha_{kw})\, M, M[\![\Rightarrow \overline{x} > x \Rightarrow a_{\mathcal{I}_d} > 0.$$

Ainsi, pour tout ensemble de d indices $\mathcal{Z}_d \subseteq [\![1,k]\!]$, par le Lemme A.2 il vient :

$$0 < \left| a_{\mathcal{I}_d} - \delta \times \prod_{i\in\mathcal{I}_d} m_i \right| < \prod_{i\in\mathcal{I}_d} m_i \leqslant \frac{\dfrac{R_d}{p_d}}{\dfrac{R_d}{p_d} \wedge M_{\mathcal{I}_d}}.$$

Ceci empêche la divisibilité de $\text{Bex}_{kwc}(\mathcal{B}, \overline{\boldsymbol{x}}_{\mathcal{B}}) - x$ par $\dfrac{R_d}{p_d}$, ce qui garantit la détection et achève la preuve du théorème.

□

Le Théorème A.2 fournit des conditions très générales sur une redondance qui n'est pas considérée *a priori* première à la base RNS principale, et qui est de plus constituée de moduli redondants non supposés premiers entre eux deux à deux.

Lorsque ceux-ci sont supposés être premiers entre eux deux à deux, alors il suffit qu'ils vérifient les conditions $m_{R,i} > m_j \times \left(m_{R,i} \wedge M_j\right)$ pour tout doublet $(i,j) \in [\![1,k]\!] \times [\![1,n]\!]$. Cette affirmation est justifiée par le lemme suivant, qui permet alors de simplifier les Hypothèses (3.3) du Théorème A.2.

Lemme A.3 *Soit $\mathcal{B} = \{m_1, \ldots, m_n\}$ une base RNS, et $\mathcal{B}_R = \{m_{R,1}, \ldots, m_{R,k}\}$ une base RNS redondante telle que $k \leqslant n$ et pour tout $(i,j) \in [\![1,k]\!] \times [\![1,n]\!]$, $m_{R,i} \geqslant m_j \times \left(m_{R,i} \wedge M_j\right)$. Alors pour tout $d \leqslant k$ et tous ensembles de d indices $\mathcal{I}_d \subset [\![1,n]\!]$ et $\mathcal{Z}_d \subset [\![1,k]\!]$,*

$$\prod_{z\in\mathcal{Z}_d} m_{R,z} \geqslant \prod_{i\in\mathcal{I}_d} m_i \times \left(\left(\prod_{z\in\mathcal{Z}_d} m_{R,z}\right) \wedge M_{\mathcal{I}_d}\right).$$

Démonstration. Pour tous ensemble d'indices $1 \leqslant i_1 < \ldots < i_d \leqslant n$ et $1 \leqslant z_1 < \ldots < z_d \leqslant k$, par hypothèse $\prod_{i=1}^{d} m_{R,z_i} \geqslant \prod_{j=1}^{d} m_{i_j} \times \prod_{j=1}^{d} \left(m_{R,z_j} \wedge M_{i_j}\right)$. De plus, pour tout $j \in [\![1,d]\!]$, $m_{R,z_j} \wedge M_{i_j} \geqslant m_{R,z_j} \wedge M_{\mathcal{I}_d}$, et par le Lemme A.1 :

$$\prod_{z \in \mathcal{Z}_d} \left(m_{R,z} \wedge M_{\mathcal{I}_d}\right) \geqslant \left(\prod_{z \in \mathcal{Z}_d} m_{R,z}\right) \wedge M_{\mathcal{I}_d}.$$

Le résultat attendu vient immédiatement. □

A.1.2 Dans un corps fini $\mathbb{F}_{p^s} \simeq \mathbb{F}_p[X]/N(X)\mathbb{F}_p[X]$

Nous généralisons certains résultats de la Section 3.2. Les données sont une base RNS de polynômes $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ ($M(X) = \prod_{i=1}^{n} m_i(X)$) vérifiant $\sum_{i=1}^{n} d_i \geqslant s = \deg(N(X))$ et un ensemble de polynômes $\mathcal{B}_R = \{m_{R,1}(X), \ldots, m_{R,k}(X)\}$ ($M_R(X) = \prod_{z=1}^{k} m_{R,z}(X)$).

Généralisation du Théorème 3.5

Théorème A.3 *Soit $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ un ensemble de n polynômes de $\mathbb{F}_p[X]$ premiers entre eux deux à deux, $d_i = \deg(m_i)$ pour tout $i \in [\![1,n]\!]$ et $d = \sum_{i=1}^{n} d_i$. Soit $M_R \in \mathbb{F}_p[X]$ un polynôme, et k un entier dans $[\![1,n]\!]$. Alors pour tout $A \in \mathbb{F}_p[X]_d$, tout entier $t \in [\![1,k]\!]$ et toute t-faute affectant $\varphi_{\mathcal{B}}(A) = \boldsymbol{A}_{\mathcal{B}}$,*

$$A_R \neq \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) \bmod M_R(X)$$

si, et seulement si,

$$\deg\left(\frac{M_R(X)}{M_R(X) \wedge \prod_{j \notin \mathcal{I}_k} m_j(X)}\right) \geqslant \max\{\sum_{i \in \mathcal{I}_k} d_i \mid \mathcal{I}_k [\![1,n]\!],\ |\mathcal{I}_k| = k\}.$$

Démonstration.

- Nécessité : la preuve est faite par contraposition. Soit un ensemble de k indices $\mathcal{I}_k$ tel que $\deg\left(\frac{M_R(X)}{M_R(X) \wedge \prod_{i \notin \mathcal{I}_k} m_i(X)}\right) < \sum_{i \in \mathcal{I}_k} d_i$. Soit $E(X)$ le polynôme suivant :

$$E(X) = \frac{M_R(X)}{M_R(X) \wedge \prod_{i \notin \mathcal{I}_k} m_i(X)}.$$

En particulier, $\deg(E) < d = \sum_{i=1}^{n} d_i$. Ainsi par la Proposition 3.4, il existe un entier $t \in [\![1,k]\!]$ et une t-faute d'indices $\mathcal{I}_t \subseteq \mathcal{I}_k$ telle que, si elle affecte

les résidus de A,

$$\begin{aligned}\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) - A(X) &= E(X) \times \prod_{i \notin \mathcal{I}_k} m_i(X) \\ &= \frac{M_R(X)}{M_R(X) \wedge \prod\limits_{i \notin \mathcal{I}_k} m_i(X)} \prod_{i \notin \mathcal{I}_k} m_i(X) \\ &= M_R(X) \frac{\prod\limits_{i \notin \mathcal{I}_k} m_i(X)}{M_R(X) \wedge \prod\limits_{i \notin \mathcal{I}_k} m_i(X)}.\end{aligned} \tag{A.17}$$

Par conséquent, l'Équation (A.17) est nulle modulo $M_R(X)$, ce qui achève la preuve de la nécessité.

- Suffisance : soit une t-faute non nulle d'indices $\mathcal{I}_t$, avec $t \in [\![1, k]\!]$, affectant $A \in \mathbb{F}_p[X]_d$. Par la Proposition 3.5,

$$\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) - A(X) = E(X) \prod_{i \notin \mathcal{I}_t} m_i(X)$$

avec $\deg(E) < \sum\limits_{i \in \mathcal{I}_t} d_i$. Ainsi, nous pouvons écrire :

$$\left|\varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}_{\mathcal{B}}\right) - A(X)\right|_{M_R(X)} = 0 \Leftrightarrow \left|E(X) \prod_{i \notin \mathcal{I}_t} m_i(X)\right|_{M_R(X)} = 0.$$

Or, la deuxième équation de l'équivalence précédente implique que $\frac{M_R(X)}{M_R(X) \wedge \prod\limits_{i \notin \mathcal{I}_t} m_i(X)}$ divise $E(X)$ et donc :

$$\deg\left(\frac{M_R(X)}{M_R(X) \wedge \prod\limits_{i \notin \mathcal{I}_t} m_i(X)}\right) < \sum_{i \in \mathcal{I}_t} d_i.$$

De plus, comme les éléments de $\mathcal{B}$ sont premiers entre eux deux à deux, alors pour tout ensemble de k indices $\mathcal{I}_k$ contenant $\mathcal{I}_t$,

$$\left(M_R(X) \wedge \prod_{i \notin \mathcal{I}_t} m_i(X)\right) = \left(M_R(X) \wedge \prod_{i \notin \mathcal{I}_k} m_i(X)\right) \times \left(M_R(X) \wedge \prod_{i \in \mathcal{I}_k \setminus \mathcal{I}_t} m_i(X)\right).$$

Par conséquent,

$$\begin{aligned}\deg\left(\frac{M_R(X)}{M_R(X) \wedge \prod\limits_{i \notin \mathcal{I}_k} m_i(X)}\right) &< \sum_{i \in \mathcal{I}_t} d_i + \deg\left(M_R(X) \wedge \prod_{i \notin \mathcal{I}_k} m_i(X)\right) \\ &< \sum_{i \in \mathcal{I}_t} d_i + \sum_{i \in \mathcal{I}_k \setminus \mathcal{I}_t} d_i \\ &= \sum_{i \in \mathcal{I}_k} d_i.\end{aligned}$$

La preuve est achevée.

□

Généralisation du Théorème 3.5

Lemme A.4 *Supposons que pour tout* $(z,i) \in [\![1,k]\!] \times [\![1,n]\!]$,

$$\deg\left(\frac{m_{R,z}(X)}{m_{R,z}(X) \wedge \prod\limits_{j\in[\![1,n]\!]\setminus\{i\}} m_j(X)}\right) \geqslant d_i$$

et que de plus pour tout quadruplet $(z_1,z_2,i_1,i_2) \in [\![1,k]\!]^2 \times [\![1,n]\!]^2$,

$$\deg\left(\frac{m_{R,z_1}(X)m_{R,z_2}(X)}{m_{R,z_1}(X) \wedge m_{R,z_2}(X)}\right) \geqslant d_{i_1} + d_{i_2} + \deg\left(\frac{m_{R,z_1}(X)m_{R,z_2}(X)}{m_{R,z_1}(X) \wedge m_{R,z_2}(X)} \wedge \prod_{j\in[\![1,n]\!]\setminus\{i_1,i_2\}} m_j(X)\right).$$

Soit un entier $\ell \leqslant k$, *ainsi que* $\mathcal{Z}_\ell \subseteq [\![1,k]\!]$ *et* $\mathcal{I}_\ell \subseteq [\![1,n]\!]$ *deux ensembles de* ℓ *indices. Si nous notons* $R_\ell(X) = \prod\limits_{z\in\mathcal{Z}_\ell} m_{R,z}(X)$ *et* $p_\ell(X) = \operatorname{pgcd}\{m_{R,z}(X) \mid z \in \mathcal{Z}_\ell\}$, *alors :*

$$\deg\left(\frac{R_\ell(X)}{p_\ell(X)}\right) \geqslant \sum_{i\in\mathcal{I}_\ell} d_i + \deg\left(\frac{R_\ell(X)}{p_\ell(X)} \wedge \prod_{j\in[\![1,n]\!]\setminus\mathcal{I}_\ell} m_j(X)\right).$$

Démonstration. La preuve suit exactement le même schéma que celle du Lemme A.2 et se fait donc par récurrence sur ℓ. De plus, le Lemme A.1 a son équivalent pour l'arithmétique sur les polynômes en se basant sur l'unicité de la décomposition en facteurs irréductibles unitaires. Autrement dit, pour tous polynômes $A(X)$, $B(X)$ et $C(X)$, $(A(X)B(X) \wedge C(X))$ divise $(A(X) \wedge C(X)) \times (B(X) \wedge C(X))$.

Les cas $\ell = 1$ et $\ell = 2$ sont simplement les hypothèses du présent lemme. Soit donc $\ell \in [\![3,k-1]\!]$, et deux ensembles de $\ell+1$ indices $\mathcal{I}_{\ell+1} = \mathcal{I}_\ell \cup \{i_{\ell+1}\} \subset [\![1,n]\!]$ et $\mathcal{Z}_{\ell+1} = \mathcal{Z}_\ell \cup \{z_{\ell+1}\} \subset [\![1,k]\!]$. Les notations suivantes sont introduites, où $\mathcal{I} \subset [\![1,n]\!]$ est un ensemble quelconque d'indices :

$$\begin{cases} R_{\ell+1}(X) = R_\ell(X) \times m_{R,z_{\ell+1}}(X), \\ p_{\ell+1}(X) = p_\ell(X) \wedge m_{R,z_{\ell+1}}(X), \\ M_{\mathcal{I}}(X) = \prod\limits_{j\in[\![1,n]\!]\setminus\mathcal{I}} m_j(X). \end{cases}$$

Il s'agit de montrer que :

$$\deg\left(\frac{R_{\ell+1}(X)}{p_{\ell+1}(X)}\right) \geqslant \sum_{i\in\mathcal{I}_{\ell+1}} d_i + \deg\left(\frac{R_{\ell+1}(X)}{p_{\ell+1}(X)} \wedge M_{\mathcal{I}_{\ell+1}}(X)\right). \tag{A.18}$$

L'hypothèse de récurrence appliquée à $\mathcal{I}_\ell$ et $\mathcal{Z}_\ell$ donne :

$$\deg\left(\frac{R_\ell(X)}{p_\ell(X)}\right) \geqslant \sum_{i\in\mathcal{I}_\ell} d_i + \deg\left(\frac{R_\ell(X)}{p_\ell(X)} \wedge M_{\mathcal{I}_\ell}(X)\right). \tag{A.19}$$

De plus, par hypothèse,

$$\deg\left(m_{R,z_{\ell+1}}(X)\right) \geqslant d_{i_{\ell+1}} + \deg\left(m_{R,z}(X) \wedge M_{i_{\ell+1}}(X)\right). \tag{A.20}$$

En combinant les Inégalités (A.19) et (A.20), il vient donc :

$$\begin{aligned}\deg\left(R_{\ell+1}(X)\right) &= \deg\left(R_\ell(X)\right) + \deg\left(m_{R,z_{\ell+1}}(X)\right)\\ &\geqslant \sum_{i\in\mathcal{I}_{\ell+1}} d_i + \deg\left(\frac{R_\ell(X)}{p_\ell(X)} \wedge M_{\mathcal{I}_\ell}(X)\right) + \deg\left(m_{R,z}(X) \wedge M_{i_{\ell+1}}(X)\right)\\ &\quad + \deg\left(p_\ell(X)\right).\end{aligned} \tag{A.21}$$

Il suffit donc de montrer que $p_\ell(X)\left(\frac{R_\ell(X)}{p_\ell(X)} \wedge M_{\mathcal{I}_\ell}(X)\right)\left(m_{R,z}(X) \wedge M_{i_{\ell+1}}(X)\right)$ est divisible par $p_{\ell+1}(X)\left(\frac{R_{\ell+1}(X)}{p_{\ell+1}(X)} \wedge M_{\mathcal{I}_{\ell+1}}(X)\right)$.

Comme $p_{\ell+1}(X)$ divise $p_\ell(X)$, il existe un polynôme $a(X)$ tel que $p_\ell(X) = a(X)p_{\ell+1}(X)$. Ainsi, de par l'équivalent du Lemme A.1 pour les polynômes :

$$\begin{aligned}p_{\ell+1}\left(\frac{R_\ell}{p_{\ell+1}} \wedge M_{\mathcal{I}_\ell}\right) &= p_{\ell+1}\left(\frac{aR_\ell}{p_\ell} \wedge M_{\mathcal{I}_\ell}\right)\\ &\quad \text{divise } p_{\ell+1}\left(a \wedge M_{\mathcal{I}_\ell}\right)\left(\frac{R_\ell}{p_\ell} \wedge M_{\mathcal{I}_\ell}\right)\\ &\quad \text{qui divise } ap_{\ell+1}\left(\frac{R_\ell}{p_\ell} \wedge M_{\mathcal{I}_\ell}\right) = p_\ell\left(\frac{R_\ell}{p_\ell} \wedge M_{\mathcal{I}_\ell}\right).\end{aligned}$$

Par conséquent, comme $\left(m_{R,z_{\ell+1}} \wedge M_{\mathcal{I}_{\ell+1}}\right)$ divise $\left(m_{R,z_{\ell+1}} \wedge M_{i_{\ell+1}}\right)$ et $\left(\frac{R_\ell}{p_{\ell+1}} \wedge M_{\mathcal{I}_{\ell+1}}\right)$ divise $\left(\frac{R_\ell}{p_{\ell+1}} \wedge M_{\mathcal{I}_\ell}\right)$:

$$\begin{aligned}p_{\ell+1}\left(\frac{R_{\ell+1}}{p_{\ell+1}} \wedge M_{\mathcal{I}_{\ell+1}}\right) &= p_{\ell+1}\left(\frac{R_\ell m_{R,z_{\ell+1}}}{p_{\ell+1}} \wedge M_{\mathcal{I}_{\ell+1}}\right)\\ &\quad \text{divise } p_{\ell+1}\left(\frac{R_\ell}{p_{\ell+1}} \wedge M_{\mathcal{I}_{\ell+1}}\right)\left(m_{R,z_{\ell+1}} \wedge M_{\mathcal{I}_{\ell+1}}\right)\\ &\quad \text{qui divise } p_{\ell+1}\left(\frac{R_\ell}{p_{\ell+1}} \wedge M_{\mathcal{I}_\ell}\right)\left(m_{R,z_{\ell+1}} \wedge M_{i_{\ell+1}}\right)\\ &\quad \text{qui divise } p_\ell\left(\frac{R_\ell}{p_\ell} \wedge M_{\mathcal{I}_\ell}\right)\left(m_{R,z_{\ell+1}} \wedge M_{i_{\ell+1}}\right).\end{aligned} \tag{A.22}$$

L'Inégalité (A.18) se déduit ainsi directement de (A.21) et (A.22). □

Théorème A.4 *Soit $\mathcal{B} = \{m_1(X), \ldots, m_n(X)\}$ n polynômes premiers entre eux deux à deux et $\mathcal{B}_R = \{m_{R,1}(X), \ldots, m_{R,k}(X)\}$ k polynômes de $\mathbb{F}_p[X]$ avec $k \leqslant n$. Pour tout $\ell \in [\![k+1, n+k]\!]$, il existe une ℓ-faute non détectable. De plus, la Procédure* `DetectMultErrExt` *détecte toutes les ℓ-fautes sur tout ensemble de résidus $(\boldsymbol{A}_\mathcal{B}, \boldsymbol{A}_R)$, pour $A \in \mathbb{F}_p[X]_d$, et ceci pour tout $\ell \leqslant k$ si, et seulement si,*

$$\begin{cases}\forall\,(z,i) \in [\![1,k]\!] \times [\![1,n]\!],\ \deg\left(\dfrac{m_{R,z}}{m_{R,z} \wedge M_i}\right) \geqslant d_i,\\ \forall\,(z_1,z_2,i_1,i_2) \in [\![1,k]\!]^2 \times [\![1,n]\!]^2,\\ \deg\left(\dfrac{m_{R,z_1}m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}}\right) \geqslant d_{i_1} + d_{i_2} + \deg\left(\dfrac{m_{R,z_1}m_{R,z_2}}{m_{R,z_1} \wedge m_{R,z_2}} \wedge M_i\right).\end{cases} \tag{A.23}$$

Démonstration. Si $\ell \geqslant k+1$, il est possible de construire une ℓ-faute non détectée par la Procédure `DetectMultErrExt`. Soit $A(X) = 0$ et une ℓ-faute

affectant $\ell - k$ résidus de A d'indices $\mathcal{I}$ ainsi que les k résidus redondants $(A_{R,z}(X))_{z\in[\![1,k]\!]}$. Soit $E_i(X) \in \mathbb{F}_p[X]_{d_i}$ l'erreur sur $A_i(X)$ et E défini par $E(X) = \varphi_{\mathcal{B}}^{-1}\left(\overline{\boldsymbol{A}}\right) - A(X)$. Par la Proposition 3.4,

$$E(X) = E_{\mathcal{I}} \times \prod_{i\notin\mathcal{I}}^{n} m_i(X)^{-1}.$$

Les autres résidus de l'erreur sont alors définis par $E_{R,z}(X) = |E(X)|_{m_{R,z}(X)}$ pour tout $z \in [\![1,k]\!]$. Une telle erreur n'est donc pas détectée.

- Prouvons maintenant par contraposition la nécessité de la condition du théorème :

  Supposons tout d'abord l'existence d'un doublet $(z,i) \in [\![1,k]\!] \times [\![1,n]\!]$ tel que :

  $$\deg\left(\frac{m_{R,z}(X)}{m_{R,z}(X)\wedge M_i(X)}\right) < d_i.$$

  Soit la k-faute sur $A(X) = 0$ donnée par les valeurs suivantes :

  $$\begin{cases} E_i(X) = \left|\dfrac{m_{R,z}(X)}{m_{R,z}(X)\wedge M_i(X)}\right|_{m_i(X)} = \dfrac{m_{R,z}(X)}{m_{R,z}(X)\wedge M_i(X)}, \\ \forall j \in [\![1,k]\!]\backslash\{z\},\ E_{R,j}(X) = \left|m_{R,z}\times\dfrac{M_i(X)}{m_{R,z}(X)\wedge M_i(X)}\right|_{m_{R,j}(X)}. \end{cases}$$

  En particulier, cela implique :

  $$\begin{aligned} \varphi_{\mathcal{B}}^{-1}\left(\boldsymbol{A}_{\mathcal{B}}\right) &= \left|\frac{m_{R,z}(X)}{m_{R,z}(X)\wedge M_i(X)}\right|_{m_i(X)} \times M_i(X), \\ &= m_{R,z} \times \frac{M_i(X)}{m_{R,z}(X)\wedge M_i(X)}. \end{aligned}$$

  Cette faute vérifie $\mathrm{Bex}\left(\mathcal{B},\mathcal{B}_R,\overline{\boldsymbol{A}}_{\mathcal{B}}\right) = \overline{\boldsymbol{A}}_R$ et n'est donc pas détectée.

  Supposons maintenant l'existence d'un quadruplet $(z_1,z_2,i_1,i_2) \in [\![1,k]\!]^2 \times [\![1,n]\!]^2$ tel que :

  $$\deg\left(R_2(X)\right) < d_{i_1} + d_{i_2} + \deg\left(R_2(X)\wedge M_{i_1i_2}(X)\right),$$

  où $R_2(X) = \dfrac{m_{R,z_1}(X)m_{R,z_2}(X)}{m_{R,z_1}(X)\wedge m_{R,z_2}(X)}$. Soit la k-faute sur $A(X) = 0$ donnée par les valeurs suivantes :

  $$\begin{cases} \forall j \in \{1,2\},\ E_{i_j} = \left|\dfrac{R_2M_{i_1i_2}}{R_2\wedge M_{i_1i_2}}\right|_{m_{i_j}}, \\ \forall z \in [\![1,k]\!]\backslash\{z\},\ E_{R,z} = \left|\dfrac{R_2M_{i_1i_2}}{R_2\wedge M_{i_1i_2}}\right|_{m_{R,z}}. \end{cases}$$

  Par hypothèse, le polynôme $\dfrac{R_2(X)}{R_2(X)\wedge M_{i_1i_2}(X)}$ est de degré strictement inférieur à $d_{i_1} + d_{i_2}$. Il est donc complètement représentable dans la base

$\{m_{i_1}(X), m_{i_2}(X)\}$ de l'espace vectoriel $\mathbb{F}_p[X]_{d_{i_1}+d_{i_2}}$. Plus précisément,

$$\begin{aligned}\frac{R_2}{R_2 \wedge M_{i_1 i_2}} &= \varphi^{-1}_{\{m_{i_1}, m_{i_2}\}}\left(\left(\left|\frac{R_2}{R_2 \wedge M_{i_1 i_2}}\right|_{m_{i_1}}, \left|\frac{R_2}{R_2 \wedge M_{i_1 i_2}}\right|_{m_{i_2}}\right)\right) \\ &= \left|\frac{R_2 m_{i_2}^{-1}}{R_2 \wedge M_{i_1 i_2}}\right|_{m_{i_1}} \times m_{i_2} + \left|\frac{R_2 m_{i_1}^{-1}}{R_2 \wedge M_{i_1 i_2}}\right|_{m_{i_2}} \times m_{i_1}.\end{aligned} \tag{A.24}$$

Par conséquent, par la Proposition 3.4 et l'Équation (A.24) il vient :

$$\begin{aligned}\varphi_{\mathcal{B}}^{-1}\left(\overline{A}_{\mathcal{B}}\right) &= \left|E_{i_1} M_{i_1}^{-1}\right|_{m_{i_1}} \times M_{i_1} + \left|E_{i_2} M_{i_2}^{-1}\right|_{m_{i_2}} \times M_{i_2} \\ &= \left(\left|\frac{R_2 m_{i_2}^{-1}}{R_2 \wedge M_{i_1 i_2}}\right|_{m_{i_1}} \times m_{i_2} + \left|E_{i_2} M_{i_2}^{-1}\right|_{m_{i_2}} \times m_{i_1}\right) \times M_{i_1 i_2} \\ &= \frac{R_2}{R_2 \wedge M_{i_1 i_2}} \times M_{i_1 i_2} \\ &= R_2 \times \frac{M_{i_1 i_2}}{R_2 \wedge M_{i_1 i_2}}.\end{aligned}$$

Ainsi, $\varphi_{\mathcal{B}_R} \circ \varphi_{\mathcal{B}}^{-1}\left(\overline{A}_{\mathcal{B}}\right) = \overline{A}_{\mathcal{B}_R}$ et la faute n'est ainsi pas détectée par la Procédure DetectMultErrExt.

- Prouvons la suffisance par contraposition. Pour ce faire, supposons qu'il existe un entier $\ell \leqslant k$ pour lequel il existe une ℓ-faute sur un polynôme $A \in \mathbb{F}_p[X]_d$ qui n'est pas détectée. Cette faute est présumée affecter ℓ_P résidus principaux d'indices $\mathcal{I}_{\ell_P}$ ainsi que ℓ_R résidus redondants. En particulier, $\ell_P + \ell_R = \ell \leqslant k$ et il y a donc au moins $\ell_P \leqslant k - \ell_R$ résidus redondants intègres. Soit ℓ_P d'entre eux d'indices $\mathcal{Z}_{\ell_P}$. Par la Proposition 3.4 qui précise la forme du polynôme donné par des résidus erronés, nous pouvons écrire :

$$\varphi_{\mathcal{B}}^{-1}\left(\overline{A}_{\mathcal{B}}\right) - A(X) = E_{\mathcal{I}_{\ell_P}}(X) \times \prod_{i \notin \mathcal{I}_{\ell_P}} m_i(X)$$

où $\deg\left(E_{\mathcal{I}_{\ell_P}}\right) < \sum_{i \in \mathcal{I}_{\ell_P}} d_i$. Par hypothèse, la faute n'est pas détectée, ce qui implique en particulier que :

$$\frac{\prod_{z \in \mathcal{Z}_{\ell_P}} m_{R,z}}{\operatorname{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_{\ell_P}\}} \text{ divise } E_{\mathcal{I}_{\ell_P}}(X) \times \left(\frac{\prod_{z \in \mathcal{Z}_{\ell_P}} m_{R,z}}{\operatorname{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_{\ell_P}\}} \wedge M_{\mathcal{I}_{\ell_P}}\right).$$

Il vient alors :

$$\deg\left(\frac{\prod_{z \in \mathcal{Z}_{\ell_P}} m_{R,z}}{\operatorname{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_{\ell_P}\}}\right) - \deg\left(\frac{\prod_{z \in \mathcal{Z}_{\ell_P}} m_{R,z}}{\operatorname{pgcd}\{m_{R,z} \mid z \in \mathcal{Z}_{\ell_P}\}} \wedge M_{\mathcal{I}_{\ell_P}}\right) < \sum_{i \in \mathcal{I}_{\ell_P}} d_i.$$

Or, par le Lemme A.4, ceci contredit les Hypothèses (A.23).

La preuve est achevée.

□

Bibliographie

"Recommended elliptic curves for federal government use", 1999. Available at http://csrc.nist.gov/encryption. (Cité page 37.)

D. Agrawal, B. Archambeault, J. R. Rao, et P. Rohatgi. The EM Side—Channel(s). Dans B. S. Kaliski, Ç. K. Koç, et C. Paar, éditeurs, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 de *Lecture Notes in Computer Science*, pages 29–45. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-00409-7. URL http://dx.doi.org/10.1007/3-540-36400-5_4. (Cité pages 3 et 47.)

M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). Dans *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM. ISBN 0-89791-785-5. URL http://doi.acm.org/10.1145/237814.237838. (Cité pages 3, 125 et 128.)

C. Aumüller, P. Bier, W. Fischer, P. Hofreiter, et J.-P. Seifert. Fault Attacks on RSA with CRT : Concrete Results and Practical Countermeasures. Dans BurtonS. Kaliski, Çetin K. Koç, et Christof Paar, éditeurs, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 de *Lecture Notes in Computer Science*, pages 260–275. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-00409-7. URL http://dx.doi.org/10.1007/3-540-36400-5_20. (Cité page 49.)

L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1) :1–13, 1986. ISSN 0209-9683. URL http://dx.doi.org/10.1007/BF02579403. (Cité pages 128 et 135.)

J.-C. Bajard, L.-S. Didier, et P. Kornerup. Modular multiplication and base extensions in residue number systems. Dans *Computer Arithmetic, 2001. Proceedings. 15th IEEE Symposium on*, pages 59–65, 2001. (Cité pages 4, 29, 33 et 36.)

J.-C. Bajard, L.-S. Didier, et J.-M. Muller. A New Euclidean Division Algorithm for Residue Number Systems. *Journal of VLSI signal processing systems for signal, image and video technology*, 19(2) :167–178, 1998. ISSN 0922-5773. (Cité page 17.)

J.-C. Bajard, S. Duquesne, et M. Ercegovac. Combining leak-resistant arithmetic for elliptic curves defined over $\mathbb{F}_p$. *Publications Mathématiques de Besançon. Algèbre et Théorie des Nombres*, pages 67–87, 2013a. ISSN ISSN : 1958-7236. URL http://pmb.univ-fcomte.fr/2013/Bajard_co.pdf. (Cité pages 4 et 32.)

J.-C. Bajard, S. Duquesne, M. Ercegovac, et N. Meloni. Residue systems efficiency for modular products summation : application to elliptic curves cryptography, 2006a. URL http://dx.doi.org/10.1117/12.679541. (Cité page 32.)

J.-C. Bajard, J. Eynard, et F. Gandino. Fault Detection in RNS Montgomery Modular Multiplication. Dans *Computer Arithmetic (ARITH), 2013 21st IEEE Symposium on*, pages 119–126, April 2013b. (Cité pages 5 et 83.)

J.-C. Bajard, J. Eynard, N. Merkiche, et T. Plantard. Babai round-off CVP method in RNS : Application to lattice based cryptographic protocols. Dans *Integrated Circuits (ISIC), 2014 14th International Symposium on*, pages 440–443, Dec 2014. (Cité pages 5 et 6.)

J.-C. Bajard, J. Eynard, N. Merkiche, et T. Plantard. (to appear) RNS Arithmetic Approach in Lattice-based Cryptography - Accelerating the "Rounding-off" Core Procedure. Dans *Computer Arithmetic (ARITH), 2013 21st IEEE Symposium on*, June 2015. (Cité pages 5, 6 et 183.)

J.-C. Bajard et H. Hördegen. Pseudo-random generator based on Chinese Remainder Theorem, 2009. URL http://dx.doi.org/10.1117/12.827023. (Cité page 188.)

J.-C. Bajard et L. Imbert. A full RNS implementation of RSA. *Computers, IEEE Transactions on*, 53(6) :769–774, June 2004. ISSN 0018-9340. (Cité pages 4 et 30.)

J.-C. Bajard, L. Imbert, et G. A. Jullien. Parallel Montgomery multiplication in $GF(2k)$ using trinomial residue arithmetic. Dans *Computer Arithmetic, 2005. ARITH-17 2005. 17th IEEE Symposium on*, pages 164–171, June 2005. (Cité page 38.)

J.-C. Bajard, L. Imbert, P.-Y. Liardet, et Y. Teglia. Leak Resistant Arithmetic. Dans M. Joye et J.-J. Quisquater, éditeurs, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 de *Lecture Notes in Computer Science*, pages 62–75. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-22666-6. URL http://dx.doi.org/10.1007/978-3-540-28632-5_5. (Cité pages ii, 4, 44, 45, 47, 61, 74 et 188.)

J.-C. Bajard, L. Imbert, et C. Negre. Arithmetic Operations in Finite Fields of Medium Prime Characteristic Using the Lagrange Representation. *Computers, IEEE Transactions on*, 55(9) :1167–1177, Sept 2006b. ISSN 0018-9340. (Cité pages 40 et 42.)

J.-C. Bajard, L. Imbert, et T. Plantard. Improving Euclidean division and modular reduction for some classes of divisors. Dans *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 2218–2221 Vol.2, Nov 2003. (Cité page 17.)

J.-C. Bajard, M. Kaihara, et T. Plantard. Selected RNS Bases for Modular Multiplication. Dans *Computer Arithmetic, 2009. ARITH 2009. 19th IEEE Symposium on*, pages 25–32, June 2009. (Cité pages 19 et 23.)

J.-C. Bajard et N. Merkiche. Double Level Montgomery Cox-Rower Architecture, New Bounds. Dans *13th Smart Card Research and Advanced Application Conference, Paris, France*, 2014. (Cité pages 4 et 19.)

A. Barenghi, G. Bertoni, A. Palomba, et R. Susella. A novel fault attack against ECDSA. Dans *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, pages 161–166, June 2011. (Cité page 4.)

A. Barenghi, L. Breveglieri, I. Koren, et D. Naccache. Fault Injection Attacks on Cryptographic Devices : Theory, Practice, and Countermeasures. *Proceedings of the IEEE*, 100(11) :3056–3076, Nov 2012. ISSN 0018-9219. (Cité pages 4, 75 et 76.)

P. Barrett. Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor. Dans *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 de *Lecture Notes in Computer Science*, pages 311–323. Springer, 1986. (Cité page 19.)

F. Barsi et P. Maestrini. Error Correcting Properties of Redundant Residue Number Systems. *Computers, IEEE Transactions on*, C-22(3) :307–315, March 1973. ISSN 0018-9340. (Cité page 52.)

K. Bigou. *Theoretical Study and Hardware Implementation of Arithmetical Units in Residue Number System (RNS) for Elliptic Curve Cryptography (ECC)*. PhD thesis, Université de Rennes 1, 2014. (Cité page 31.)

K. Bigou et A. Tisserand. RNS modular multiplication through reduced base extensions. Dans *Application-specific Systems, Architectures and Processors (ASAP), 2014 IEEE 25th International Conference on*, pages 57–62, June 2014. (Cité page 4.)

D. Boneh, R. A. DeMillo, et R. J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. Dans Walter Fumy, éditeur, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 de *Lecture Notes in Computer Science*, pages 37–51. Springer Berlin Heidelberg, 1997. ISBN 978-3-540-62975-7. URL http://dx.doi.org/10.1007/3-540-69053-0_4. (Cité pages 4 et 48.)

D. Brumley et D. Boneh. Remote Timing Attacks Are Practical. Dans *Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12*, SSYM'03, pages 1–1, Berkeley, CA, USA, 2003. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1251353.1251354. (Cité pages 3 et 47.)

P. W. Cheney. *An investigation of residue number theory for digital systems*. PhD thesis, Stanford University, 1962. (Cité page 49.)

R. C. C. Cheung, S. Duquesne, J. Fan, N. Guillermin, I. Verbauwhede, et G. X. Yao. FPGA Implementation of Pairings Using Residue Number System and Lazy Reduction. Dans *Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'11, pages 421–441, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23950-2. URL http://dl.acm.org/citation.cfm?id=2044928.2044966. (Cité pages 4, 31 et 32.)

M. Ciet, M. Neve, E. Peeters, et J.-J. Quisquater. Parallel FPGA implementation of RSA with residue number systems - can side-channel threats be avoided ? Dans *Circuits and Systems, 2003 IEEE 46th Midwest Symposium on*, volume 2, pages 806–810 Vol. 2, Dec 2003. (Cité page 31.)

R. Conway et J. Nelson. Improved RNS FIR filter architectures. *Circuits and Systems II : Express Briefs, IEEE Transactions on*, 51(1) :26–28, Jan 2004. ISSN 1549-7747. (Cité page 9.)

D. Coppersmith et S. Winograd. Matrix Multiplication via Arithmetic Progressions. Dans *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 1–6, New York, NY, USA, 1987. ACM. ISBN 0-89791-221-7. URL http://doi.acm.org/10.1145/28395.28396. (Cité page 170.)

J.-S. Coron. Resistance Against Differential Power Analysis For Elliptic Curve Cryptosystems. Dans Çetin K. Koç et Christof Paar, éditeurs, *Cryptographic Hardware and Embedded Systems*, volume 1717 de *Lecture Notes in Computer Science*, pages 292–302. Springer Berlin Heidelberg, 1999. ISBN 978-3-540-66646-2. URL http://dx.doi.org/10.1007/3-540-48059-5_25. (Cité pages 3 et 47.)

W. Diffie et M. E. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6) :644–654, Nov 1976. ISSN 0018-9448. (Cité page 1.)

I. Dinur, G. Kindler, R. Raz, et S. Safra. Approximating cvp to within almost-polynomial factors is np-hard. *Combinatorica*, 23(2) :205–243, Avril 2003. ISSN 0209-9683. URL http://dx.doi.org/10.1007/s00493-003-0019-y. (Cité page 128.)

S. Duquesne et N. Guillermin. A FPGA pairing implementation using the Residue Number System. Cryptology ePrint Archive, 2011. (Cité pages 29 et 32.)

T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4) :469–472, Jul 1985. ISSN 0018-9448. (Cité page 2.)

P. Erdös. A Theorem of Sylvester and Schur. *Journal of the London Mathematical Society*, s1-9(4) :282–288, 1934. URL http://jlms.oxfordjournals.org/content/s1-9/4/282.short. (Cité page 18.)

M. Etzel et W. Jenkins. Redundant residue number systems for error detection and correction in digital filters. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(5) :538–545, Oct 1980. ISSN 0096-3518. (Cité pages 52 et 55.)

C. F. de Barros et L. Menasché Schechter. GGH may not be dead after all. Dans *Proceedings of XXXV Congresso Nacional de Matemática Aplicada e Computacional (CNMAC 2014)*, sept. 2014. (Cité page 134.)

PUB FIPS. 46-3 : Data encryption standard (des). *National Institute of Standards and Technology*, 1999. (Cité page 1.)

PUB FIPS. 197 : Advanced encryption standard (aes). *National Institute of Standards and Technology*, November 2001. (Cité page 1.)

PUB FIPS. 186-4 : Digital signatures standards (dss). *National Institute of Standards and Technology*, July 2013. (Cité page 2.)

P.-A. Fouque, N. Guillermin, D. Leresteux, M. Tibouchi, et J.-C. Zapalowicz. Attacking RSA–CRT Signatures with Faults on Montgomery Multiplication. Dans Emmanuel Prouff et Patrick Schaumont, éditeurs, *Cryptographic Hardware and Embedded Systems – CHES 2012*, volume 7428 de *Lecture Notes in Computer Science*, pages 447–462. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33026-1. URL http://dx.doi.org/10.1007/978-3-642-33027-8_26. (Cité page 49.)

D. Freeman, M. Scott, et E. Teske. A taxonomy of pairing-friendly elliptic curves. Cryptology ePrint Archive, Report 2006/372, 2006. http://eprint.iacr.org/. (Cité page 37.)

W. L. Freking et K. K. Parhi. Low-power FIR digital filters using residue arithmetic. Dans *Signals, Systems amp ; Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, volume 1, pages 739–743 vol.1, Nov 1997. (Cité page 9.)

D. Gallaher, F. E. Petry, et P. Srinivasan. The digit parallel method for fast RNS to weighted number system conversion for specific moduli $(2^k - 1, 2^k, 2^k + 1)$. *Circuits and Systems II : Analog and Digital Signal Processing, IEEE Transactions on*, 44(1) :53–57, Jan 1997. ISSN 1057-7130. (Cité page 19.)

F. Gandino, F. Lamberti, P. Montuschi, et J.-C. Bajard. A General Approach for Improving RNS Montgomery Exponentiation Using Pre-processing. Dans *Computer Arithmetic (ARITH), 2011 20th IEEE Symposium on*, pages 195–204, July 2011. (Cité page 35.)

F. Gandino, F. Lamberti, G. Paravati, J.-C. Bajard, et P. Montuschi. An Algorithmic and Architectural Study on Montgomery Exponentiation in RNS. *IEEE Trans. Computers*, 61(8) :1071–1083, 2012. (Cité page 35.)

K. Gandolfi, C. Mourtel, et F. Olivier. Electromagnetic Analysis : Concrete Results. Dans Ç. K. Koç, D. Naccache, et C. Paar, éditeurs, *Cryptographic Hardware and Embedded Systems — CHES 2001*, volume 2162 de *Lecture Notes in Computer Science*, pages 251–261. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-42521-2. URL http://dx.doi.org/10.1007/3-540-44709-1_21. (Cité pages 3 et 47.)

H. L. Garner. The Residue Number System. Dans *Papers Presented at the the March 3-5, 1959, Western Joint Computer Conference*, IRE-AIEE-ACM '59 (Western), pages 146–153, New York, NY, USA, 1959. ACM. URL http://doi.acm.org/10.1145/1457838.1457864. (Cité page 9.)

K. A. Gbolagade et S. D. Cotofana. An $\mathcal{O}(n)$ Residue Number System to Mixed Radix Conversion technique. Dans *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 521–524, May 2009. (Cité page 20.)

C. Gentry. Fully Homomorphic Encryption Using Ideal Lattices. Dans *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-506-2. URL http://doi.acm.org/10.1145/1536414.1536440. (Cité pages 3 et 125.)

O. Goldreich, S. Goldwasser, et S. Halevi. Public-key cryptosystems from lattice reduction problems. Dans Jr. Kaliski, BurtonS., éditeur, *Advances in Cryptology — CRYPTO '97*, volume 1294 de *Lecture Notes in Computer Science*, pages 112–131. Springer Berlin Heidelberg, 1997. ISBN 978-3-540-63384-6. URL http://dx.doi.org/10.1007/BFb0052231. (Cité pages 3, 125, 129, 132, 133, 134, 136, 138, 156 et 180.)

O. Goldreich, D. Micciancio, S. Safra, et J.-P. Seifert. Approximating Shortest Lattice Vectors is Not Harder Than Approximating Closet Lattice Vectors. *Inf. Process. Lett.*, 71(2) :55–61, Juillet 1999. ISSN 0020-0190. URL http://dx.doi.org/10.1016/S0020-0190(99)00083-6. (Cité page 128.)

L. Goubin. A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. Dans YvoG. Desmedt, éditeur, *Public Key Cryptography — PKC 2003*, volume 2567 de *Lecture Notes in Computer Science*, pages 199–211. Springer Berlin Heidelberg, 2002. ISBN 978-3-540-00324-3. URL http://dx.doi.org/10.1007/3-540-36288-6_15. (Cité pages 3 et 47.)

S. Gueron. Enhanced Montgomery Multiplication. Dans *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 de *Lecture Notes in Computer Science*, pages 46–56. Springer, 2002. (Cité page 34.)

N. Guillermin. A High Speed Coprocessor for Elliptic Curve Scalar Multiplications over $\mathbb{F}_p$. Dans Stefan Mangard et François-Xavier Standaert, éditeurs, *Cryptographic Hardware and Embedded Systems, CHES 2010*, volume 6225 de *Lecture Notes in Computer Science*, pages 48–64. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15030-2. URL http://dx.doi.org/10.1007/978-3-642-15031-9_4. (Cité pages 4, 29, 31 et 36.)

N. Guillermin. A coprocessor for secure and high speed modular arithmetic. Cryptology ePrint Archive, Report 2011/354, 2011. http://eprint.iacr.org/. (Cité pages ii, 29, 45, 48, 77, 83, 84 et 85.)

N. Guillermin. *Implémentation matérielle de coprocesseurs haute performance pour la cryptographie asymétrique*. PhD thesis, Université de Rennes 1, 2012. (Cité page 31.)

S. Guilley, L. Sauvage, J.-L. Danger, et N. Selmane. Fault Injection Resilience. Dans *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2010 Workshop on*, pages 51–65, Aug 2010. (Cité page 77.)

G. Hachez et J.-J. Quisquater. Montgomery Exponentiation with no Final Subtractions : Improved Results. Dans Ç. K. Koc et C. Paar, éditeurs, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 de *Lecture Notes in Computer Science*, pages 293–301. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-41455-1. URL http://dx.doi.org/10.1007/3-540-44499-8_23. (Cité page 34.)

M. A. Hitz et E. Kaltofen. Integer division in residue number systems. *Computers, IEEE Transactions on*, 44(8) :983–989, Aug 1995. ISSN 0018-9340. (Cité page 17.)

J. Hoffstein, J. Pipher, et J. H. Silverman. NTRU : A ring-based public key cryptosystem. Dans JoeP. Buhler, éditeur, *Algorithmic Number Theory*, volume 1423 de *Lecture Notes in Computer Science*, pages 267–288. Springer Berlin Heidelberg, 1998. ISBN 978-3-540-64657-0. URL http://dx.doi.org/10.1007/BFb0054868. (Cité page 3.)

C. Huang et F. J. Taylor. High speed DFT's using residue numbers. Dans *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80.*, volume 5, pages 238–242, Apr 1980. (Cité page 9.)

L. H. Ing. The history of the Chinese remainder theorem. *Mathematical Medley*, 30(1) :54–62, 2003. ISSN 0217-2976. (Cité page 9.)

G. A. Jullien J.-C. Bajard, L. Imbert et H. C. Williams. A CRT-Based Montgomery Multiplication for Finite Fields of Small Characteristic. Dans *IMACS'05 : World Congress : Scientific Computation, Applied Mathematics and Simulation*, July 2005. (Cité page 38.)

W. Jenkins et B. Leon. The use of residue number systems in the design of finite impulse response digital filters. *Circuits and Systems, IEEE Transactions on*, 24(4) :191–201, Apr 1977. ISSN 0098-4094. (Cité page 9.)

M. Joye et S.-M. Yen. The Montgomery Powering Ladder. Dans B. S. Kaliski, Ç. K. Koç, et C. Paar, éditeurs, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 de *Lecture Notes in Computer Science*, pages 291–302. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-00409-7. URL http://dx.doi.org/10.1007/3-540-36400-5_22. (Cité page 47.)

S. Kangsheng. Historical development of the Chinese remainder theorem. *Archive for History of Exact Sciences*, 38(4) :285–305, 1988. ISSN 0003-9519. URL http://dx.doi.org/10.1007/BF00357063. (Cité page 9.)

S. Kawamura, M. Koike, F. Sano, et A. Shimbo. Cox-Rower Architecture for Fast Parallel Montgomery Multiplication. Dans Bart Preneel, éditeur, *EUROCRYPT*, volume 1807 de *Lecture Notes in Computer Science*, pages 523–538. Springer, 2000. (Cité pages 4, 26, 27, 29, 30, 36, 65, 76, 78, 84, 85 et 170.)

A. Kerckhoffs. La cryptographie militaire. *Journal des Sciences Militaires*, pages 161–191, 1883. (Cité page 1.)

Ç. K. Koç et T. Acar. Montgomery Multiplication in $GF(2^k)$. *Designs, Codes and Cryptography*, 14(1) :57–69, 1998. ISSN 0925-1022. (Cité page 42.)

N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177) : 203–209, Janvier 1987. ISSN 0025-5718. (Cité pages 2 et 49.)

Ç. K. Koç. High-Speed RSA Implementation. Rapport technique, RSA Laboratories, November 1994. URL ftp://ftp.rsasecurity.com/pub/pdfs/tr201.pdf. (Cité page 48.)

P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. Dans Neal Koblitz, éditeur, *Advances in Cryptology — CRYPTO '96*, volume 1109 de *Lecture Notes in Computer Science*, pages 104–113. Springer Berlin Heidelberg, 1996. ISBN 978-3-540-61512-5. URL http://dx.doi.org/10.1007/3-540-68697-5_9. (Cité pages 3 et 47.)

P. C. Kocher, J. Jaffe, et B. Jun. Differential Power Analysis. Dans *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397, London, UK, UK, 1999. Springer-Verlag. ISBN 3-540-66347-9. URL http://dl.acm.org/citation.cfm?id=646764.703989. (Cité pages 3 et 47.)

P. C. Kocher, J. Jaffe, B. Jun, et P. Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering*, 1(1) :5–27, 2011. ISSN 2190-8508. URL http://dx.doi.org/10.1007/s13389-011-0006-y. (Cité pages 3 et 47.)

H. Krishna, K.-Y. Lin, et Jenn-Dong Sun. A coding theory approach to error control in redundant residue number systems. I. Theory and single error correction. *Circuits and Systems II : Analog and Digital Signal Processing, IEEE Transactions on*, 39(1) :8–17, Jan 1992. ISSN 1057-7130. (Cité pages 52 et 55.)

L. Y. Lam et T. S. Ang. *Fleeting footsteps : tracing the conception of arithmetic and algebra in ancient China* . River Edge, N.J. ; Singapore : World Scientific, rev. ed édition, 2004. ISBN 9812386963. Revised edition includes an edited text of Lam Lay Yong's plenary lecture, 'Ancient Chinese mathematics and its influence on world mathematics', delivered at the International Congress of Mathematicians, Beijing, 2002. (Cité page 9.)

A. K. Lenstra, Jr. Lenstra, H. W., et L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4) :515–534, 1982. ISSN 0025-5831. URL http://dx.doi.org/10.1007/BF01457454. (Cité page 128.)

D. Mandelbaum. Error Correction in Residue Arithmetic. *Computers, IEEE Transactions on*, C-21(6) :538–545, June 1972. ISSN 0018-9340. (Cité pages 52 et 55.)

P. M. Matutino, H. Pettenghi, R. Chaves, et L. Sousa. RNS Arithmetic Units for Modulo $\{2^n \pm k\}$. Dans *Digital System Design (DSD), 2012 15th Euromicro Conference on*, pages 795–802, Sept 2012. (Cité page 19.)

S. Medoš et S. Boztaş. Montgomery Residue Representation Fault-Tolerant Computation in $GF(2k)$. Dans *Information Security and Privacy, 13th Australasian Conference, ACISP 2008, Wollongong, Australia, July 7-9, 2008, Proceedings*, pages 419–432, 2008. URL http://dx.doi.org/10.1007/978-3-540-70500-0_31. (Cité pages ii, v, 87, 116 et 119.)

D. Micciancio. Improving Lattice Based Cryptosystems Using the Hermite Normal Form. Dans JosephH. Silverman, éditeur, *Cryptography and Lattices*, volume 2146 de *Lecture Notes in Computer Science*, pages 126–145. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-42488-8. URL http://dx.doi.org/10.1007/3-540-44670-2_11. (Cité pages 126, 133, 134, 139, 155, 170 et 177.)

V. S. Miller. Use of Elliptic Curves in Cryptography. Dans HughC. Williams, éditeur, *Advances in Cryptology — CRYPTO '85 Proceedings*, volume 218 de *Lecture Notes in Computer Science*, pages 417–426. Springer Berlin Heidelberg, 1986. ISBN 978-3-540-16463-0. URL http://dx.doi.org/10.1007/3-540-39799-X_31. (Cité pages 2 et 49.)

P.V. Ananda Mohan. RNS-To-Binary Converter for a New Three-Moduli Set $\{2^{n+1}-1, 2^n, 2^n-1\}$. *Circuits and Systems II : Express Briefs, IEEE Transactions on*, 54(9) :775–779, Sept 2007. ISSN 1549-7747. (Cité page 19.)

P.V. Ananda Mohan. New reverse converters for the moduli set. *AEU - International Journal of Electronics and Communications*, 62(9) :643 – 658, 2008. ISSN 1434-8411. URL http://www.sciencedirect.com/science/article/pii/S1434841107001586. (Cité page 19.)

P. L. Montgomery. Modular Multiplication without Trial Division. *Mathematics of Computation*, 44(170) :519–521, 1985. (Cité pages 19 et 31.)

P. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97. Dans Michael Wiener, éditeur, *Advances in Cryptology — CRYPTO' 99*, volume 1666 de *Lecture Notes in Computer Science*, pages 288–304. Springer Berlin Heidelberg, 1999. ISBN 978-3-540-66347-8. URL http://dx.doi.org/10.1007/3-540-48405-1_18. (Cité pages 123 et 133.)

P. Nguyen et O. Regev. Learning a Parallelepiped : Cryptanalysis of GGH and NTRU Signatures. Dans Serge Vaudenay, éditeur, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 de *Lecture Notes in Computer Science*, pages 271–288. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-34546-6. URL http://dx.doi.org/10.1007/11761679_17. (Cité page 123.)

P. Nguyen et J. Stern. The Two Faces of Lattices in Cryptology. Dans J. H. Silverman, éditeur, *Cryptography and Lattices*, volume 2146 de *Lecture Notes in Computer Science*, pages 146–180. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-42488-8. URL http://dx.doi.org/10.1007/3-540-44670-2_12. (Cité page 125.)

H. Nozaki, M. Motoyama, A. Shimbo, et S. Kawamura. Implementation of RSA Algorithm Based on RNS Montgomery Multiplication. Dans Ç. K. Koç, D. Naccache, et C. Paar, éditeurs, *Cryptographic Hardware and Embedded Systems — CHES 2001*, volume 2162 de *Lecture Notes in Computer Science*, pages 364–376. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-42521-2. URL http://dx.doi.org/10.1007/3-540-44709-1_30. (Cité pages 29, 31, 72 et 81.)

A. M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. Dans Thomas Beth, Norbert Cot, et Ingemar Ingemarsson, éditeurs, *Advances in Cryptology*, volume 209 de *Lecture Notes in Computer Science*, pages 224–314. Springer Berlin Heidelberg, 1985. ISBN 978-3-540-16076-2. URL http://dx.doi.org/10.1007/3-540-39757-4_20. (Cité page 2.)

M. Otto. *Fault Attacks and Countermeasures*. PhD thesis, Universität Paderborn, 2004. (Cité page 76.)

D. Page et F. Vercauteren. A Fault Attack on Pairing-Based Cryptography. *Computers, IEEE Transactions on*, 55(9) :1075–1080, Sept 2006. ISSN 0018-9340. (Cité page 49.)

G. Perin, L. Imbert, L. Torres, et P. Maurine. Practical Analysis of RSA Countermeasures Against Side-Channel Electromagnetic Attacks. Dans A. Francillon et P. Rohatgi, éditeurs, *Smart Card Research and Advanced Applications*, Lecture Notes in Computer Science, pages 200–215. Springer International Publishing, 2014. ISBN 978-3-319-08301-8. URL http://dx.doi.org/10.1007/978-3-319-08302-5_14. (Cité page 48.)

T. Plantard. *Arithmétique modulaire pour la cryptographie*. Theses, Université Montpellier II - Sciences et Techniques du Languedoc, Décembre 2005. URL https://tel.archives-ouvertes.fr/tel-00112121. (Cité page 189.)

T. Plantard, M. Rose, et W. Susilo. Improvement of Lattice-Based Cryptography Using CRT. Dans Alexander Sergienko, Saverio Pascazio, et Paolo Villoresi, éditeurs, *Quantum Communication and Quantum Networking*, volume 36 de *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 275–282. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-11730-5. URL http://dx.doi.org/10.1007/978-3-642-11731-2_34. (Cité pages 134, 135 et 136.)

K. C. Posch et R. Posch. Modulo reduction in residue number systems. *Parallel and Distributed Systems, IEEE Transactions on*, 6(5) :449–454, May 1995. ISSN 1045-9219. (Cité page 26.)

A. B. Premkumar, E. L. Ang, et E. M. Lai. Improved memoryless RNS forward converter based on the periodicity of residues. *Circuits and Systems II : Express Briefs, IEEE Transactions on*, 53(2) :133–137, Feb 2006. ISSN 1549-7747. (Cité page 169.)

R. L. Rivest, A. Shamir, et L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2) :120–126, Février 1978. ISSN 0001-0782. URL http://doi.acm.org/10.1145/359340.359342. (Cité page 2.)

M. Rose, T. Plantard, et W. Susilo. Improving BDD Cryptosystems in General Lattices. Dans Feng Bao et Jian Weng, éditeurs, *Information Security Practice and Experience*, volume 6672 de *Lecture Notes in Computer Science*, pages 152–167. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-21030-3. URL http://dx.doi.org/10.1007/978-3-642-21031-0_12. (Cité page 134.)

J. B. Rosser et L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6(1) :64–94, 03 1962. URL http://projecteuclid.org/euclid.ijm/1255631807. (Cité page 18.)

C.-P. Schnorr. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Theor. Comput. Sci.*, 53 :201–224, 1987. URL http://dblp.uni-trier.de/db/journals/tcs/tcs53.html#Schnorr87. (Cité page 128.)

A. Shamir. Method and apparatus for protecting public key schemes from timing and fault attacks, November 1999. URL https://www.google.com/patents/US5991415. US Patent 5,991,415. (Cité page 49.)

A.P. Shenoy et R. Kumaresan. Fast base extension using a redundant modulus in . *Computers, IEEE Transactions on*, 38(2) :292–297, Feb 1989. ISSN 0018-9340. (Cité page 24.)

M.-H. Sheu, S.-H. Lin, C. Chen, et S.-W. Yang. An efficient VLSI design for a residue to binary converter for general balance moduli $(2^n - 3, 2^n + 1, 2^n - 1, 2^n + 3)$. *Circuits and Systems II : Express Briefs, IEEE Transactions on*, 51(3) : 152–155, March 2004. ISSN 1549-7747. (Cité page 19.)

P. W. Shor. Algorithms for quantum computation : discrete logarithms and factoring. Dans *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134, Nov 1994. (Cité pages 2 et 125.)

M. A. Soderstrand, W. K. Jenkins, G. A. Jullien, et F. J. Taylor, éditeurs. *Residue Number System Arithmetic : Modern Applications in Digital Signal Processing*. IEEE Press, Piscataway, NJ, USA, 1986. ISBN 0-87942-205-X. (Cité page 9.)

SP. 800-56a : Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography. *National Institute of Standards and Technology*, March 2007. (Cité page 2.)

T. Stouraitis et V. Paliouras. Considering the alternatives in low-power design. *Circuits and Devices Magazine, IEEE*, 17(4) :22–29, Jul 2001. ISSN 8755-3996. (Cité page 9.)

V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13 (4) :354–356, 1969. ISSN 0029-599X. URL http://dx.doi.org/10.1007/BF02165411. (Cité page 170.)

N. S. Szabo et R. I. Tanaka. *Residue Arithmetic and its application to Computer Technology*. McGraw-Hill, 1967. (Cité pages 20 et 22.)

F. J. Taylor. An RNS discrete Fourier transform implementation. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(8) :1386–1394, Aug 1990. ISSN 0096-3518. (Cité page 9.)

B.-D. Tseng, G. A. Jullien, et W. C. Miller. Implementation of FFT Structures Using the Residue Number System. *Computers, IEEE Transactions on*, C-28 (11) :831–845, Nov 1979. ISSN 0018-9340. (Cité page 9.)

P. van Emde-Boas. *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*. Report. Department of Mathematics. University of Amsterdam. Department, Univ., 1981. URL http://books.google.fr/books?id=tCQiHQAACAAJ. (Cité page 128.)

D. Vigilant. RSA with CRT : A New Cost-Effective Solution to Thwart Fault Attacks. Dans Elisabeth Oswald et Pankaj Rohatgi, éditeurs, *Cryptographic Hardware and Embedded Systems – CHES 2008*, volume 5154 de *Lecture Notes in Computer Science*, pages 130–145. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85052-6. URL http://dx.doi.org/10.1007/978-3-540-85053-3_9. (Cité page 49.)

C. D. Walter. Montgomery exponentiation needs no final subtractions. *Electronics Letters*, 35(21) :1831–1832, Oct 1999. ISSN 0013-5194. (Cité page 34.)

R. W. Watson et C. W. Hastings. Self-checked computation using residue arithmetic. *Proceedings of the IEEE*, 54(12) :1920–1931, Dec 1966. ISSN 0018-9219. (Cité pages 49 et 55.)

C. Whelan et M. Scott. The Importance of the Final Exponentiation in Pairings When Considering Fault Attacks. Dans Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, et Takeshi Okamoto, éditeurs, *Pairing-Based Cryptography – Pairing 2007*, volume 4575 de *Lecture Notes in Computer Science*, pages 225–246. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-73488-8. URL http://dx.doi.org/10.1007/978-3-540-73489-5_12. (Cité page 49.)

V. V. Williams. Multiplying Matrices Faster Than Coppersmith-Winograd. Dans *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 887–898, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1245-5. URL http://doi.acm.org/10.1145/2213977.2214056. (Cité page 170.)

G. X. Yao, J. Fan, R. C. C. Cheung, et I. Verbauwhede. Faster Pairing Coprocessor Architecture. Dans Michel Abdalla et Tanja Lange, éditeurs, *Pairing-Based Cryptography, Pairing 2012*, volume 7708 de *Lecture Notes in Computer Science*, pages 160–176. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36333-7. URL http://dx.doi.org/10.1007/978-3-642-36334-4_10. (Cité page 29.)

S.S.-S. Yau et Yu-Cheng L. Error Correction in Redundant Residue Number Systems. *Computers, IEEE Transactions on*, C-22(1) :5–11, Jan 1973. ISSN 0018-9340. (Cité pages 52 et 55.)

S.-M. Yen, S. Kim, S. Lim, et S.-J. Moon. RSA speedup with Chinese remainder theorem immune against hardware fault cryptanalysis. *Computers, IEEE Transactions on*, 52(4) :461–472, April 2003. ISSN 0018-9340. (Cité page 77.)

M. Yoshino et N. Kunihiro. Improving GGH cryptosystem for large error vector. Dans *Information Theory and its Applications (ISITA), 2012 International Symposium on*, pages 416–420, Oct 2012. (Cité page 134.)

Ce document a été préparé à l'aide du logiciel de composition typographique LaTeX 2ε.

Titre Approche arithmétique RNS de la cryptographie asymétrique

Résumé Cette thèse se situe à l'intersection de la cryptographie et de l'arithmétique des ordinateurs. Elle traite de l'amélioration de primitives cryptographiques asymétriques en termes d'accélération des calculs et de protection face aux attaques par fautes par le biais particulier de l'utilisation des systèmes de représentation des nombres par les restes (RNS). Afin de contribuer à la sécurisation de la multiplication modulaire, opération centrale en cryptographie asymétrique, un nouvel algorithme de réduction modulaire doté d'une capacité de détection de faute est présenté. Une preuve formelle garantit la détection des fautes sur un ou plusieurs résidus pouvant apparaître au cours d'une réduction. De plus, le principe de cet algorithme est généralisé au cas d'une arithmétique dans un corps fini non premier $\mathbb{F}_{p^s}$. Ensuite, les RNS sont exploités dans le domaine de la cryptographie sur les réseaux euclidiens. L'objectif est d'importer dans ce domaine certains avantages des systèmes de représentation par les restes dont l'intérêt a déjà été montré pour une arithmétique sur $\mathbb{F}_p$ notamment. Le premier résultat obtenu est une version en représentation hybride RNS-MRS de l'algorithme du « round-off » de Babai. Puis, une technique d'accélération est introduite, permettant d'aboutir dans certains cas à un algorithme entièrement RNS pour le calcul d'un vecteur proche.

Mots-clés Cryptographie asymétrique, arithmétique des ordinateurs, systèmes de représentation par les restes, RNS, conversions de base, faute, détection, RNS redondant, architecture Cox-Rower, cryptographie basée sur les réseaux, problème du plus proche vecteur, round-off de Babai

Title RNS arithmetic approach of asymmetric cryptography

Abstract This thesis is at the crossroads between cryptography and computer arithmetic. It deals with enhancement of cryptographic primitives with regard to computation acceleration and protection against fault injections through the use of residue number systems (RNS) and their associated arithmetic. So as to contribute to secure the modular multiplication, which is a core operation for many asymmetric cryptographic primitives, a new modular reduction algorithm supplied with fault detection capability is presented. A formal proof guarantees that faults affecting one or more residues during a modular reduction are well detected. Furthermore, this approach is generalized to an arithmetic dedicated to non-prime finite fields $\mathbb{F}_{p^s}$. Afterwards, RNS are used in lattice-based cryptography area. The aim is to exploit acceleration properties enabled by RNS, as it is widely done for finite field arithmetic. As first result, a new version of Babai's round-off algorithm based on hybrid RNS-MRS representation is presented. Then, a new and specific acceleration technique enables to create a full RNS algorithm computing a close lattice vector.

Keywords Asymmetric cryptography, computer arithmetic, residue number systems, RNS, base conversions, fault, detection, redundant RNS, Cox-Rower architecture, lattice-based cryptography, closest vector problem, Babai's round-off