Chiffrement homomorphe et recherche par le contenu sécurisé de données externalisées et mutualisées : Application à l'imagerie médicale et l'aide au diagnostic

Reda Bellafqira

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

UNIVERSITE BRETAGNE LOIRE

THÈSE / IMT Atlantique

sous le sceau de l'Université Bretagne Loire

pour obtenir le grade de

DOCTEUR D'IMT Atlantique

Spécialité : Informatique

École Doctorale Mathématiques et STIC

Présentée par

Reda Bellafqira

Préparée dans le département Image & traitement de l'information

Laboratoire Latim

Chiffrement homomorphe et recherche par le contenu sécurisé de données externalisées et mutualisées - Application à l'imagerie médicale et l'aide au diagnostic

Thèse soutenue le 19 décembre 2017

devant le jury composé de :

Maryline Laurent
Professeure, Télécom SudParis / présidente

Alexandre Moreau-Gaudry
Professeur, Laboratoire TIM-IMAG - La Tronche / rapporteur

François Arnault
Maître de conférences (HDR), Faculté des Sc. et Techniques de Limoges / rapporteur

Gwénolé Quellec
Chargé de recherche, CHRU Morvan - Brest / examinateur

Gouenou Coatrieux
Professeur, IMT Atlantique/ directeur de thèse

Dalel Bouslimi
Ingénieure d'études et développement, Sopra Steria - Colomiers / invitée

Michel Cozic
Directeur R&D, Medecom – Plougastel-Daoulas / invité

Remerciements

Cette thèse a été réalisée au sein du département ITI de IMT Atlantique (Brest) et du LaTIM-Inserm U1101 avec le soutien financier de la région de Bretagne.

Par ces quelques lignes, je tiens à remercier toutes les personnes qui ont participé de près ou de loin au bon déroulement de cette thèse, en espérant n'avoir oublié personne.

Je tiens à adresser en premier lieu mes plus chaleureux remerciements à mon directeur de thèse M. Gouenou COATRIEUX. Il m'a transmis la passion de la recherche et n'a eu de cesse de m'encourager et de me soutenir durant toutes les années que j'ai travaillés avec lui. Il a beaucoup facilité la longue traversée de mes années de doctorat. Merci beaucoup Gouenou pour ta gentillesse, ta patience et tes précieux conseils. J'ai beaucoup apprécié le travail avec toi tant sur le plan scientifique que sur le plan humain.

Je remercie également Mme Dalel BOUSLIMI, M. Gwénolé QUELLEC et M. Michel COZIC pour leur encadrement tout au long de cette thèse, leurs conseils, leurs encouragements et ses qualités humaines.

J'exprime tous mes remerciements à Mme Maryline LAURENT d'avoir accepté de présider le jury d'examen. Je lui adresse mes sentiments les plus respectueux.

Je tiens à remercier les professeurs M. François ARNAULT et M. Alexandre MOREAU-GAUDRY pour avoir bien voulu rapporter sur mon travail de thèse et pour toutes ses précieuses remarques qui m'ont beaucoup aidé à améliorer la qualité de ce travail.

Je remercie très chaleureusement tous mes collègues et ex-collègues de bureau : Wei PAN, Javier FRANCO CONTRERAS, Mohamed KARASAD, David NIYITEGEKA, Sahar HADDAD...Je remercie aussi tous les membres du département ITI et du laboratoire LaTIM pour la sympathie et l'aide qu'ils m'ont témoignées durant ces années.

Enfin, je tiens à remercier mes parents et mon frère. Malgré mon éloignement depuis de nombreuses années, leur confiance, leur tendresse, leur amour me portent et me guident tous les jours. Merci maman et papa pour avoir fait de moi ce que je suis aujourd'hui.

Table des matières

Table des figures

Liste des tableaux

Abstract

Cloud computing has emerged as a successful paradigm allowing individuals and companies to exibly store and process large amounts of data without a need to purchase and maintain their own networks and computer systems. In healthcare for example, different initiatives aim at sharing medical images and Personal Health Records (PHR) in between health professionals or hospitals with the help of the cloud. In such an environment, data security (confidentiality, integrity and traceability) is a major issue. In this context that these thesis works, it concerns in particular the securing of Content Based Image Retrieval (CBIR) techniques and machine learning (ML) which are at the heart of diagnostic decision support systems. These techniques make it possible to find images similar to an image not yet interpreted. The goal is to define approaches that can exploit secure externalized data and enable a cloud to provide a diagnostic support. Several mechanisms allow the processing of encrypted data, but most are dependent on interactions between different entities (the user, the cloud or a trusted third party) and must be combined judiciously so as not to leak information.

During these three years of thesis, we initially focused on securing an outsourced CBIR system under the constraint of no interaction between the users and the service provider. In a second step, we have developed a secure machine learning approach based on multilayer perceptron (MLP), whose learning phase can be outsourced in a secure way, the challenge being to ensure the convergence of the MLP. All the data and parameters of the model are encrypted using homomorphic encryption. Because these systems need to use information from multiple sources, each of which outsources its encrypted data under its own key, we are interested in the problem of sharing encrypted data. A problem known by the "Proxy Re-Encryption" (PRE) schemes. In this context, we have proposed the first PRE scheme that allows both the sharing and the processing of encrypted data. We also worked on watermarking scheme over encrypted data in order to trace and verify the integrity of data in this shared environment. The embedded message is accessible whether or not the image is encrypted and provides several services.

Keywords : Cloud-computing, Homomorphic Encryption, Content Based Image Retrieval, Proxy Re-Encryption, Digital watermarking, Secure machine learning.

Résumé

La mutualisation et l'externalisation de données concernent de nombreux domaines y compris celui de la santé. Au-delà de la réduction des coûts de maintenance, l'intérêt est d'améliorer la prise en charge des patients par le déploiement d'outils d'aide au diagnostic fondés sur la réutilisation des données. Dans un tel environnement, la sécurité des données (confidentialité, intégrité et traçabilité) est un enjeu majeur. C'est dans ce contexte que s'inscrivent ces travaux de thèse. Ils concernent en particulier la sécurisation des techniques de recherche d'images par le contenu (CBIR) et de « machine learning » qui sont au cœur des systèmes d'aide au diagnostic. Ces techniques permettent de trouver des images semblables à une image requête non encore interprétée. L'objectif est de définir des approches capables d'exploiter des données externalisées et sécurisées, et de permettre à un « cloud » de fournir une aide au diagnostic. Plusieurs mécanismes permettent le traitement de données chiffrées, mais la plupart sont dépendants d'interactions entre différentes entités (l'utilisateur, le cloud voire un tiers de confiance) et doivent être combinés judicieusement de manière à ne pas laisser fuir d'information lors d'un traitement.

Au cours de ces trois années de thèse, nous nous sommes dans un premier temps intéressés à la sécurisation à l'aide du chiffrement homomorphe, d'un système de CBIR externalisé sous la contrainte d'aucune interaction entre le fournisseur de service et l'utilisateur. Dans un second temps, nous avons développé une approche de « Machine Learning » sécurisée fondée sur le perceptron multicouches, dont la phase d'apprentissage peut être externalisée de manière sûre, l'enjeu étant d'assurer la convergence de cette dernière. L'ensemble des données et des paramètres du modèle sont chiffrés. Du fait que ces systèmes d'aides doivent exploiter des informations issues de plusieurs sources, chacune externalisant ses données chiffrées sous sa propre clef, nous nous sommes intéressés au problème du partage de données chiffrées. Un problème traité par les schémas de « Proxy Re-Encryption » (PRE). Dans ce contexte, nous avons proposé le premier schéma PRE qui permet à la fois le partage et le traitement des données chiffrées. Nous avons également travaillé sur un schéma de tatouage de données chiffrées pour tracer et vérifier l'intégrité des données dans cet environnement partagé. Le message tatoué dans le chiffré est accessible que l'image soit ou non chiffrée et offre plusieurs services de sécurité fondés sur le tatouage.

Mots clés : Cloud-computing, chiffrement homomorphe, calcul multipartite sécurisé, recherche par le contenu, Proxy Re-Encryption, tatouage des images, apprentissage automatique sécurisé.

Acronyms

CBIR	Content Based Image Retrieval
SCBIR	Secure Content Based Image Retrieval
MLP	MultiLayer Perceptron
QIM	Quantization Index Modulation
DWT	Discrete Wavelet Transform
ML	Machine Learning
SI	Système d'Information
PACS	Picture Archiving and Communication System
DICOM	Digital Imaging and Communications in Medicine
PHE	Partially Homomorphic Encryption
FHE	Fully Homomorphic Encryption
SHE	Somewhat Homomorphic Encryption
PRE	Proxy Re-Encryption
HPRE	Homomorphic Proxy Re-Encryption
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
CLCG	Combined Linear Congruential Generator
SCLCG	Secure Combined Linear Congruential Generator
PKI	Public Key Infrastructure
OPE	Order Preserving Encryption
COA	Ciphertext Only Attack

Introduction générale

La mutualisation et l'externalisation de données concernent de nombreux domaines y compris celui de la santé. Au-delà de la réduction des coûts de maintenance et des services, l'intérêt est aussi de faciliter et d'améliorer la prise en charge des patients par le biais du partage de données et la mise à disposition de nouveaux services comme des outils d'aide au diagnostic fondés sur la réutilisation des données. Dans un tel environnement, la sécurité des données (confidentialité, authenticité, intégrité et traçabilité) est un enjeu majeur. En effet, l'utilisateur perd le contrôle sur les données qu'il externalise. C'est dans ce contexte que s'inscrivent ces travaux de thèse. Ils concernent en particulier la sécurisation des techniques de recherche d'images par le contenu (« Content Based Image Retrieval » - CBIR) et d'apprentissage automatique («Machine learning» - ML) qui sont au coeur des systèmes d'aide au diagnostic. Ces techniques permettent de trouver des images semblables à une image requête non encore interprétée sur la base de signatures extraites des images. L'objectif est de définir des approches capables d'exploiter des données externalisées et sécurisées, notamment sous forme chiffrées, et de permettre à un cloud de trouver des données semblables sans accéder à leur contenu réel. Plusieurs mécanismes permettent le traitement de données chiffrées (chiffrement homomorphe, méthodes de "searchable encryption", de "secret sharing", de calcul multipartite sécurisé « Secure multiparty computation »-SMC, etc...), mais la plupart sont dépendants d'interactions/communications entre différentes entités (l'utilisateur, le cloud voire un tiers de confiance) et doivent être combinés judicieusement de manière à ne pas laisser fuir d'information lors d'un traitement.

Au cours de ces trois années de thèse, nous nous sommes dans un premier temps intéressés à la sécurisation d'un système de CBIR externalisé à l'aide du chiffrement homomorphe sous la contrainte d'aucunes interactions entre le cloud et l'utilisateur ou un tiers de confiance. L'externalisation de tels processus d'analyse de données n'est pas encore aujourd'hui complètement formalisé. La question de savoir comment il est possible d'externaliser les fonctions d'un système de CBIR ou d'un système de Machine Learning, ne fait l'objet d'aucune réponse consensuelle si ce n'est le besoin d'externaliser les procédés dans leur totalité. Dans le même temps, il convient de s'interroger du degré de confiance accordé au fournisseur de service à qui l'on demande de traiter ses données, même chiffrées. Ce niveau de confiance dépend du « modèle d'attaquant » que l'on associera à ce fournisseur de service. Est-il honnête, curieux voire même malveillant ? C'est après avoir identifié ces besoins que nous avons développé une première approche qui extrait une signature globale d'une image. Cela n'a pas été étudié jusqu'aujourd'hui. Nous tirons avantage de méthodes originales qui permettent la comparaison de données chiffrées sans communications [2]. Comme beaucoup de solutions, notamment celles qui exploitent des signatures locales, la signature extraite par cette approche est en clair et a pour conséquence de rendre le système vulnérable à des attaques qui profitent de connaissances *a priori* sur les distributions statistiques des signaux. Nous nous sommes alors intéressés à trouver une solution à ce problème [3, 4] : l'extraction de signatures chiffrées d'images chiffrées sur la base d'approches de calcul sécurisé multipartite (c'est à des traitements de données chiffrées partagés entre différentes entités qui ne colludent pas).

Les techniques d'apprentissage automatique (i.e. machine learning, "ML" ») pour l'aide au diagnostic sont aujourd'hui l'objet d'enjeux importants, notamment avec l'avènement des données massives ou du « big data ». Elles contribuent à l'intelligence artificielle apprenant par exemple à différencier des images contenant des pathologies spécifiques. La mise en oeuvre de ce type d'approches comporte deux phases : une phase d'apprentissage à l'issue de laquelle le système s'est entraîné à reconnaitre des formes, des objets, etc., sur la base ou non d'exemples (apprentissage supervisé vs. non-supervisé) ; et, la phase de classification où le système assure la fonction pour laquelle il a été construit. Si les méthodes classiques de classification (e.g. « Decision tree ») ont été sécurisées, ce n'est pas le cas de la phase d'apprentissage des réseaux de neurones qui sont à la base de l'apprentissage profond (« deep learning »). Nombreux sont ceux qui s'intéressent aujourd'hui au deep learning. C'est une des approches capable de réutiliser des données massives et qui montre des gains de performances réels. Cependant, pour profiter pleinement de ces techniques, encore faut-il pouvoir réaliser l'apprentissage sur des données externalisées sécurisées, tout en s'assurant que les paramètres du modèle appris restent eux aussi confidentiels. Ce sont eux qui constituent la valeur ajoutée du service ainsi construit. C'est une tâche sur laquelle nous sommes concentrés. Nous avons développé un réseau de neurone sécurisé à l'aide du chiffrement homomorphe et le calcul multipartite sécurisé.

Du fait que ces systèmes de CBIR ou ML externalisés exploitent des informations issues de plusieurs sources, chacune externalisant ses données chiffrées avec sa propre clé, nous nous sommes intéressés au problème du partage de données chiffrées. Un problème traité par les méthodes de « Proxy Re-Encryption » (PRE). Il s'avère aujourd'hui que si ces solutions permettent le partage de données entre plusieurs utilisateurs, elles ne permettent de post-traiter les données sans les rapatrier et les externaliser à nouveau. Profitant des propriétés des cryptosystèmes homomorphes, nous avons développé un nouveau concept : « Homomorphic Proxy Re-encryption » [5]. Comme nous le verrons, il diffère en de nombreux d'un PRE classique et s'appuie sur : i) une méthode de calcul de différence de données chiffrées généralisée à plusieurs algorithmes de chiffrement homomorphe (Paillier, Damgard-Jurik, BGV) : ii) la sécurisation de générateurs de nombres aléatoires (LCG et CLCG) qui permet au cloud, et non aux utilisateurs comme actuellement proposé dans la littérature, de générer des séquences aléatoires chiffrées sur la base de clés fournies par les utilisateurs. Une question ouverte, et à laquelle nous avons trouvé une réponse porte sur la possibilité de re-partager et de traiter ces données sans les rapatrier et les re-externaliser.

Pouvoir partager des données entre plusieurs utilisateurs soulève aussi des questions en termes de traçabilité des données et de leur intégrité. Sur ce point les solutions de type Log ou basées sur la blockchain sont limitées. Le tatouage de données est plus approprié car il permet de dissimuler l'information de protection dans la donnée elle-même. Dans notre contexte, il est nécessaire de tatouer des données chiffrées. Nous avons ainsi travaillé sur un schéma de tatouage de données chiffrées homomorphiquement interopérable avec des solutions de CBIR et de machine learning sécurisées [6] pour tracer et vérifier l'intégrité des données dans cet environnement partagé. Le message tatoué dans le chiffré est accessible que l'image soit ou non chiffrée et offre plusieurs services de sécurité fondés sur le tatouage.

Ces travaux de thèse s'articulent autour de cinq chapitres. Dans le chapitre 1, nous reviendrons sur l'intérêt de partager les données des données de santé. Si à l'origine le partage se limitait aux professionnels de santé en charge d'un patient, l'arrivée du « cloud» chamboule toutes les pratiques. Il facilite le partage de données entre professionnels de santé indépendamment des frontières des hôpitaux. L'imagerie médicale, qui joue un rôle essentiel dans la prise en charge des patients, n'échappe pas à cette évolution mais nécessite des techniques de CBIR et d'apprentissage automatiques appropriées. Dans la deuxième partie de ce chapitre, nous aborderons

les besoins de protection des données de santé, qui résultent notamment d'un cadre législatif et déontologique des plus strictes ; comme aussi comment ces données sont protégées aujourd'hui. Nous introduirons des notions de politique de sécurité au niveau d'un système d'information et les différents mécanismes qui permettent de sécuriser des traitements de données (comme la CBIR). Nous présenterons en particulier les nouveaux outils comme le chiffrement homomorphe.

L'objet du deuxième chapitre porte sur une première contribution de ce travail de thèse. Elle concerne un système de recherche par le contenu externalisé sécurisé qui exploite les propriétés d'homomorphie du cryptosystème de Paillier et une méthode de comparaison des données chiffrées sous la contrainte d'aucune communication entre l'utilisateur et le cloud. Le système sécurisé que nous proposons découle de ceux proposés pour la confidentialité des données stockées dans le cloud et également la requête de l'utilisateur. Nous analysons également la sécurité de ce système et discutons de ses faiblesses dans le cas où le cloud à des informations supplémentaires quant aux données qu'il manipule. Nous présentons aussi la complexité de cette solution en termes de calcul, communication et de stockage.

Pour pallier aux contraintes et faiblesses de la solution précédente, nous présentons, dans le Chapitre 3 deux nouveaux systèmes de recherche par le contenu sécurisés dont l'objectif est de garantir la confidentialité des images externalisées, des signatures calculées et des requêtes présentées par les utilisateurs. Nous verrons ainsi une première solution sécurisée qui s'appuie sur l'utilisation de deux fournisseurs de cloud indépendants. Ce système permet de calculer et comparer les signatures extraites d'images sans aucune fuite d'information sur les données et sans interactions avec l'utilisateur. Cette approche a ensuite été améliorée pour ne pas être dépendante de communications entre deux fournisseurs de cloud, au prix cependant d'un accroissement de la complexité de calcul. Celle-ci s'appuie sur un algorithme original semblable à un système PIR (« Private Information Retrieval ») qui permet une sélection secrète de données. Comme précédemment, la sécurité et les performances des deux approches sont évaluées dans le cas de bases de données médicales et biométriques.

Dans le chapitre 4, nous nous intéressons à la sécurité d'une technique d'apprentissage automatique : le perceptron multicouches, et notamment à la sécurisation de sa phase d'apprentissage. Comme évoqué plus haut, il est nécessaire de pouvoir externaliser les phases d'apprentissage des techniques de machine learning pour bénéficier pleinement de données mutualisées et externalisées. Cela n'a pas été fait dans le cas des réseaux de neurones. Les travaux présentés dans ce chapitre restent cependant modestes, puisque nous validons la solution que nous proposons dans le cas de l'apprentissage de la fonction ET-Logique, mais mettent en évidence les problèmes de convergence de l'apprentissage sur des données chiffrées. Difficultés que nous avons réussi à surmonter. Dans ce chapitre, nous discutons l'analyse de sécurité de cette approche et de ses performances en termes d'apprentissage et de classification. Le chapitre 5 présente les travaux réalisés sur le tatouage et le partage de données chiffrées homomorphiquement. Après avoir présenté la définition et les méthodes classiques de « Proxy Re-Encryption » (PRE), nous introduisons un nouveau concept de PRE fondé sur le chiffrement homomophe (HPRE). Ces deux approches différent en particulier sur deux fonctions clés que nous présenterons en détails. Si elles se valent en termes de propriétés de sécurité obtenues (unidirectionnalité et « collusion resistant»), le HPRE est de complexité moindre et permet le post-traitements des données sans les télécharger. Cette approche a été testée dans le cas du partage d'images. Une image de pixels peut être partagée en une minute sur un ordinateur classique. Dans chapitre, nous aborderons également le tatouage de données chiffrées avec une solution qui combine le chiffrement homomorphe et la modulation de tatouage substitutive par modulation d'indice. Elle permet l'insertion d'un message dans une image chiffrée ; message qui est accessible à la fois dans le

domaine spatial (en clair) et chiffré et qui peut servir différents services de sécurité (contrôle d'intégrité et traçabilité des données).

Chapitre 1

Externalisation d'outils d'aide à la décision et enjeux de sécurité

Il est facile de constater aujourd'hui que la mutualisation et l'externalisation de données sont au cœur de nombreuses innovations dans tous les domaines, y compris celui de la santé. Au-delà de la réduction des coûts de maintenance et des services, l'intérêt est notamment de faciliter et d'améliorer la prise en charge des patients par le biais du partage et la mise à disposition de nouveaux services comme des outils d'aide à la décision fondés en particulier sur la réutilisation de données massives. Cependant, et comme nous le verrons dans ce chapitre, cette externalisation soulève de nombreux problèmes en matière de sécurité.

Ce chapitre s'articule en trois parties. Dans la première nous reviendrons sur l'intérêt de partager les données de santé. Si à l'origine le partage se limitait aux professionnels de santé en charge d'un patient, l'avènement de nouvelles technologies comme le « cloud compouting » ou de « data warehouse » (entrepôt de données) permet de mutualiser l'information des patients au-delà des frontières des hôpitaux et ainsi donner accès à de larges bases de données, des données massives, qui peuvent par la suite être réutilisées par le biais de mécanismes d'apprentissage automatique pour l'aide à la pratique médicale au sens large, i.e. tant pour un patient que pour une population. Nous verrons que l'imagerie médicale n'échappe pas à cette évolution bien qu'elle se développe plus tardivement notamment en raison des volumes que ces données représentent. La réutilisation des images médicales nécessite par ailleurs de déployer des mécanismes spécifiques. En particulier, on utilisera des méthodes de recherche d'images par le contenu (« Content Based Image Retrieval » - CBIR) propres au domaine de la santé. Ces techniques de CBIR permettent de trouver des images semblables à une image requête. Elles sont au cœur des systèmes d'aide au diagnostic, au choix thérapeutique et font l'objet d'intérêts industriels [7]. Un enjeu aujourd'hui, encore peu étudié, est de savoir comment externaliser ce type d'approche dans le contexte du cloud. À l'issu de cette première partie, nous suggérons différents scénarios de CBIR externalisée. Ces derniers ont servi de cadre de travail à ces travaux de thèse. La deuxième partie de ce chapitre aborde les besoins de protéger les données. Nous verrons que ceux-ci sont imposés par le citoyen au travers d'un cadre législatif et déontologique particulièrement strict. Ils évoluent aussi en fonction du contexte applicatif, i.e. de la finalité du traitement de données et des moyens technologiques déployés. En effet, la menace est d'autant plus forte que les systèmes d'information (SI) sont ouverts sur Internet.

La troisième partie a trait aux moyens de protection des données qui sont aujourd'hui disponibles. Nous verrons notamment que la sécurité est principalement abordée au niveau des SI au travers d'une politique de sécurité qui, après avoir identifié les besoins de sécurité, définit comment déployer différents mécanismes pour atteindre un niveau de protection optimal. Cependant, ces solutions rencontrent quelques limites dans le contexte externalisé du cloud. Une question clé est « comment » traiter des données externalisées, i.e. des données sorties du

périmètre de leur SI d'origine et hors du contrôle de leur propriétaire, de manière sécurisée. Nous présenterons de nouveaux outils comme le chiffrement homomorphe, qui permet de chiffrer les données tout en laissant la possibilité d'appliquer quelques traitements, et qui peuvent aider à atteindre cet objectif. Des mécanismes de CBIR pour des applications grand publique et limitées le plus souvent au domaine de l'authentification biométrique, ont notamment été sécurisés. Néanmoins, de par leur nature ces algorithmes de CBIR ne sont pas forcément adaptés au domaine de la santé, d'autres méthodes plus efficaces, fondées sur des principes différents comme le calcul d'histogramme ou les réseaux de neurones, n'ont pas encore été entièrement sécurisés. Nous aborderons également d'autres limites, comme le fait que ces méthodes extraient des signatures non sécurisées qui peuvent laisser fuiter de l'information.

I Externalisation/Mutualisation/Réutilisation de données pour l'aide à la décision

La collecte d'information dans la prise en charge d'un patient d'ores et déjà conséquente continue de s'enrichir avec le développement de l'informatique, des télécommunications et des moyens d'observations du vivant. Le patient lui-même peut contribuer à l'aide d'objets connectés (e.g. montre connectée) qui, par le biais de senseurs, fournissent des données de bien-être. Pouvoir accéder aux données de milliers, voire de millions de patients, offre la possibilité d'observer une pathologie sous ces différentes formes, voire de manière exhaustive. La réutilisation de ces données, qui représente de facto une source de connaissances massives et leur comparaison à des données nouvellement acquises et non encore traitées par un médecin offre des perspectives extrêmement fortes en termes de services d'aide à la pratique médicale. La cible première étant le développement d'outils d'aide à la décision. De nombreux projets de recherche publiques comme industriels, nationaux et internationaux, sont aujourd'hui financés sur ce sujet, pour des pathologies qui ont notamment des conséquences économiques lourdes pour la société (e.g. diabètes). Cependant, ces projets sont avant tout des projets en informatique médicale. On pourra par exemple citer le projet ANR INSHARE [1] qui ambitionne de développer une plateforme « Health Big Data » sur le grand ouest français. La généralisation de ces approches aux données d'imagerie médicale, c'est-à-dire aux images et leurs données ancillaires, est en cours. Elle profite, comme nous allons le voir, de deux évolutions majeures : une évolution technologique, qui facilite la mutualisation et le partage des images, et le développement de mécanismes de recherche et de fouille dans les images, permettant notamment la recherche de cas cliniques semblables.

I.1 Du partage à la mutualisation des données d'imagerie

Les premiers systèmes de stockage communicants liés à l'image au sein des établissements hospitaliers ont été les PACS ("Picture Archiving and Communication System") [8]. Ils répondent à l'idée centrale d'aller vers l'hôpital numérique. Ils sont toujours d'actualité et sont au coeur des plateaux techniques d'imagerie. Comme illustré de manière simplifiée en Figure 1.1, ils s'appuient sur différents composante : des modalités d'imagerie (e.g. radiographie, scanner) connectées en réseau avec des stations de diagnostic ou d'interprétation et un serveur d'archivage. Dans un tel système, les images sont stockées selon le standard DICOM [9], le standard de référence en imagerie médicale. Ce dernier défini d'une part un format de fichier mais aussi différents services. Ainsi, une image DICOM sera constituée des pixels de l'image d'un entête contenant des informations relatives à l'examen et aux données « image » (e.g. paramètres d'acquisition,

1. Référence projet : ANR-15-CE19-0024

taille d'un pixel et de l'image, nombre de bits pour coder un pixel, type et identifiant de la modalité d'acquisition, nom du patient, type d'examen, angle de vue, etc.). Considérant une image comme un objet, DICOM spécifie différents services permettant par exemple le transfert d'une image du serveur de stockage à la station de diagnostic ou encore l'impression d'une image sous forme de film. C'est un standard très riche, incontournable pour les éditeurs de logiciels en imagerie médicale.

Si ces systèmes permettent la mutualisation des données d'imagerie au sein d'hôpital, ils ont évolué vers les PACS régionaux [10] [11] [12] qui, comme leur nom l'indique, sont un passage à l'échelle avec l'interconnexion de systèmes PACS de plusieurs hôpitaux. À titre d'exemple, on peut citer le PACS régional breton : RUBIS (RéseaU Breton d'Imagerie de Santé). En parallèle, ces systèmes se sont enrichis avec des applications de télémédecine, où l'image joue un rôle majeur. Aujourd'hui, nous arrivons à la constitution d'entrepôts de données et en particulier au cloud en imagerie médicale ("cloud medical imaging") [9, 13, 14] qui nous intéresse dans ces travaux. En effet, le cloud donne à la mutualisation de données d'imagerie médicale une dimension toute particulière, mettant à disposition une masse exceptionnelle d'expériences et de connaissances. Dans ce qui suit, nous revenons sur la définition du Cloud computing, ses modèles de services et de déploiement en particulier dans le secteur de la santé. Selon la définition du

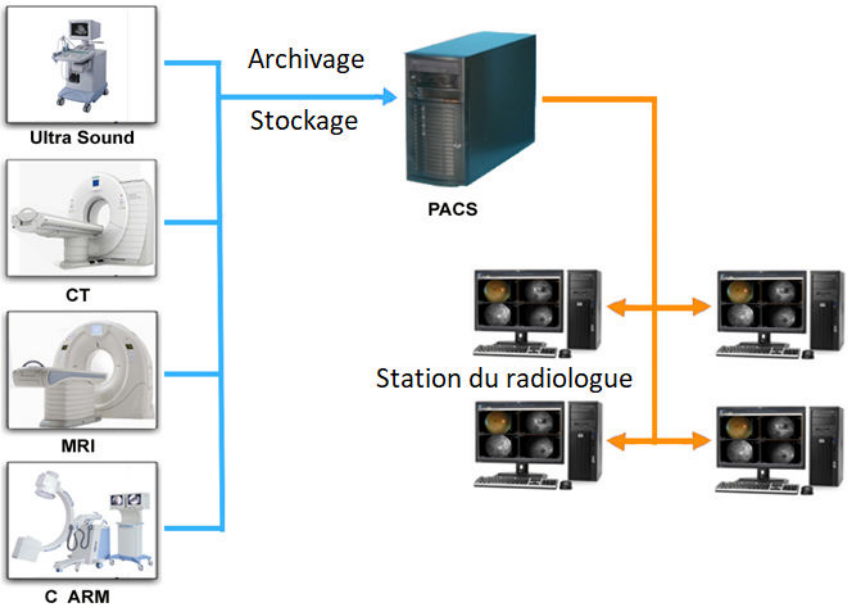


FIGURE 1.1 – Architecture du PACS

National Institute of Standards and Technology (NIST), le cloud désigne un modèle qui permet un accès à des ressources informatiques (e.g. réseaux, serveurs, stockage, applications/logiciels, services, puissance de calcul) partagées, mutualisées et configurables et à la demande ; ressources qui peuvent être approvisionnées avec un effort de gestion minime et une faible interaction avec l'utilisateur [15].

L'accès à un cloud se fait le plus souvent à l'aide d'un navigateur Web, à la demande et en libre-service. Ils existent trois modèles de services cloud (voir Figure 1.2) :

— « software as a service » (SaaS), où le consommateur accède en ligne à des applications hébergées sur l'infrastructure du fournisseur de cloud. Le modèle SaaS peut supporter de

nombreuses applications, telles que la gestion de relation client, la messagerie électronique (e.g. Gmail) et la manipulation collaborative ou non de documents (e.g. : Google Docs (Drive), Lotus Live (IBM), Online CRM (Salesforce.com)).

— « platform as a service » (PaaS). Ce service permet au consommateur de déployer sur une infrastructure cloud des applications qu'il aura lui-même développées à l'aide d'outils mis à sa disposition par le fournisseur de cloud. Un tel environnement d'hébergement et de développement peut être constitué d'outils de bases de données, de composants « middleware » et de logiciels d'infrastructure (e.g. Openshift (Red Hat), Google App Engine (pour les applications web essentiellement)). Ici, le consommateur n'a aucun contrôle sur l'infrastructure. Il n'a que la maîtrise des applications déployées et leurs configurations.

— « infrastructure as a service » (IaaS). Cette fois-ci le fournisseur met à disposition une puissance de traitement, de l'espace de stockage, des infrastructures de réseaux ainsi que d'autres ressources informatiques, en permettant au consommateur de déployer et d'exécuter des applications de son choix. Le consommateur conserve le contrôle des systèmes d'exploitation, des espaces de stockage, des applications déployées et certains composants réseau. Cela permet aux administrateurs du système et aux développeurs du côté « consommateur » d'accéder par eux-mêmes aux ressources de calcul, de stockage et de réseau dont ils ont besoin pour déployer et exécuter des applications et des systèmes d'exploitation sur le cloud (e.g. Amazon EC2 - Elastic Compute Cloud). Plus généralement, le consommateur gère un parc d'ordinateurs ou de machines virtuelles sur le cloud ; machines dont il a le contrôle total. C'est l'administrateur du cloud qui réparti ces machines virtuelles sur ces moyens informatiques, optimisant ainsi l'exploitation de ses ressources.

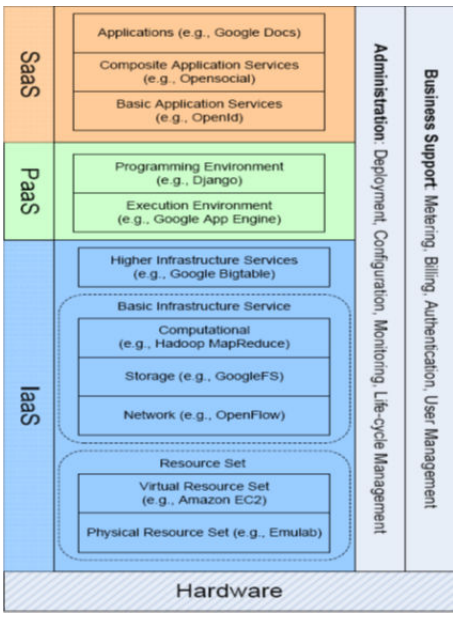


FIGURE 1.2 – Modèles de cloud

Une infrastructure cloud peut être de trois types ou trous modèles distincts :

— Le cloud publique dans lequel l'infrastructure est située chez un hébergeur, un fournisseur qui propose des services cloud à plusieurs consommateurs qui n'ont pas forcément de liens entre eux (e.g. entreprises, hôpitaux ou particuliers). Le fournisseur de cloud est un tiers qui

a le contrôle du matériel, de la connectivité réseau et gère l'optimisation de l'exploitation des ressources informatiques comme logicielles.

— Le cloud privé est lui exclusivement destiné à une seule organisation. Son infrastructure est située dans les locaux de cette organisation qui joue à la fois les rôles de fournisseur de services et de consommateurs. Comparé à un cloud publique, un cloud privé offre des performances moindres, notamment parce que ses ressources sont financées par l'organisation, en générale de petite taille et donc avec des moyens financiers limités.

— Le cloud hybride est à l'intersection entre les deux premiers modèles. Il est composé d'au moins deux infrastructures : une privée et une publique. Celles-ci conservent leur autonomie. Son intérêt est de permettre l'accès à des ressources supplémentaires lorsque le cloud privé a atteint ses limites d'exploitation

Le déploiement de ces trois modèles et de leurs services a été et est toujours étudié dans le domaine de la santé. Dans [16], c'est une solution de cloud public donnant accès des images médicales qui est proposée. Elle a pour objectif de faciliter l'échange et le partage des images d'une part entre professionnels de santé et avec les patients d'autre part. Comme illustré en Figure 1.3, l'infrastructure de ce cloud s'appuie sur une plateforme Hadoop [2] en mode PaaS et offre des services SaaS sous la forme de web-services. Dans l'article [17], les auteurs ont proposés

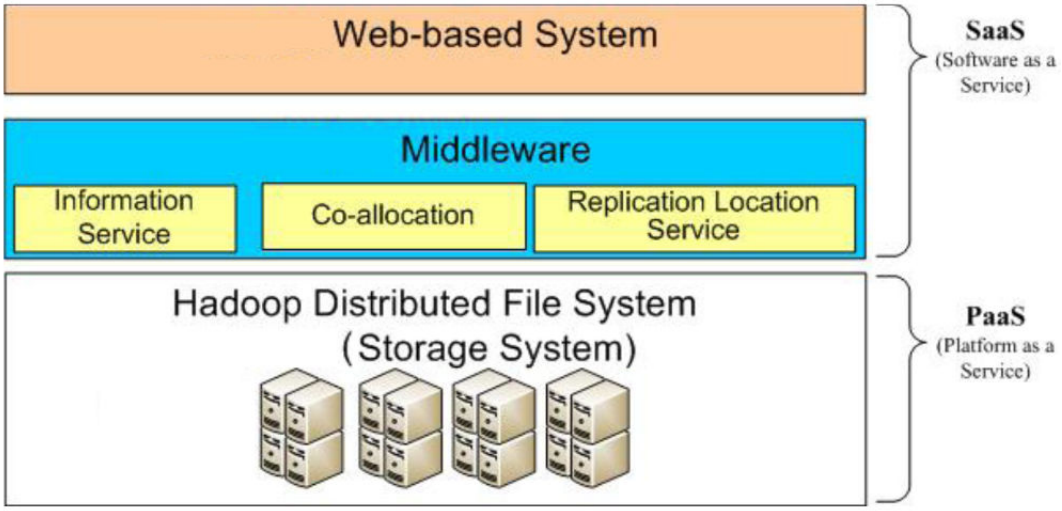


FIGURE 1.3 – Exemple de déploiement du cloud dans le domaine médical

une architecture de plateforme de cloud privé qui comprend six couches selon les exigences spécifiques qui utilise une file d'attente de message (Message Queue) en tant que moteur de cloud. La couche de stockage utilise un système de stockage distribué comme Hadoop [18] afin de fournir un accès plus extensible et efficace.

Dans l'article [19], les auteurs ont présenté un système de santé intelligent fondé sur le cloud publique et les données massives (Big Data), et qui comprend 1) une couche de collecte de données unifiée pour l'intégration des ressources médicales publiques et des dispositifs personnels de santé ; 2) une plate-forme qui permet l'unification des données de multisource, le stockage et l'analyse des données pour les soins, et 3) une API unifiée pour les développeurs et une interface unifiée pour les utilisateurs. Diverses applications et services personnalisés sont développés pour

2. Hadoop est un framework libre propose par google destiné à faciliter la gestion de fichiers dans le cadre du big et facilite et la création d'applications distribuées.

relever les défis dans les soins de santé traditionnels, y compris les ressources centralisées et la participation passive des patients.

Bien que le cloud offre des fonctionnalités particulièrement pertinentes en termes d'économie d'échelle et d'optimisation des ressources, du fait de son caractère ouvert sur internet et l'appel à des tiers, fournisseurs de services, il soulève de nombreuses questions en matière de sécurité et de protection de données. 45% des entreprises interrogées placent la sécurité comme le premier frein d'une migration dans le cloud [20].

I.2 Système d'aide au diagnostic fondé sur la ré-exploitation de données d'imagerie

L'image joue un rôle clé dans la prise en charge des patients. Il existe différentes modalités d'imagerie, comme l'échographie, la tomographie par rayons X, l'IRM, imagerie nucléaire, etc. En fonction des principes physiques impliqués, ces modalités donnent une vue spécifique de l'organisme du patient. Elles permettent d'observer les tissus mous, les structures rigides ou encore la vitesse des flux (cf. table 1.1).

Ces modalités sont aujourd'hui à l'origine d'un flux massif de données. Par exemple, un hôpital génère en moyenne jusqu'à 27000 terabytes par an. Collectées à très large échelle, ces images constituent une masse de connaissances non négligeable où l'on peut espérer trouver les différentes formes ou expression d'une pathologie de manière exhaustive. Profiter de cette masse de données comme référence pour l'aide au diagnostic est un enjeu majeur aujourd'hui. Toute la question est comment reconnaître de manière automatique des signes d'anomalies, caractéristiques d'une pathologie, dans une image nouvellement acquise. La réponse à cette question, passe par l'indexation automatique des images qui s'appuie notamment sur des techniques de recherche par le contenu (Content Based Image Retrieval- CBIR). En imagerie médicale, les applications potentielles de la CBIR sont : la catégorisation, c.-à-d. la recherche d'images contenant des régions anatomiques identiques [21–23]. Le « data mining » [24] qui permet de retrouver des images avec des lésions visuellement similaires [25–28]. Et enfin à l'aide au diagnostic [29–31].

L'objectif d'un système d'aide à la décision fondé sur la CBIR, le type de système qui nous intéresse plus particulièrement dans ces travaux de thèse, est de permettre à un expert de mener une recherche dans une base d'images, sans formuler une description sémantique de l'image qu'il examine. Plus clairement, l'expert envoie simplement une image en requête au système qui va chercher les images les plus similaires contenues dans sa base de données (cf. Figure 1.7). Enfin, l'expert choisit parmi les images retournées celles qui sont de son point de vue les plus similaires à son image. Il peut accéder aux informations associées à ces dernières, et récupérer les diagnostics associés voire l'historique des patients concernés. Une telle recherche résume le contenu de l'image à un vecteur de caractéristiques : une signature. Ces signatures permettent de comparer les images et trouver les plus semblables à une image requête. Deux images seront dites proches si leurs signatures sont proches selon une mesure de distance appropriée (Voir Figure 1.4). Toute la difficulté de la CBIR est donc de trouver des descripteurs qui décrivent l'information pertinente du contenu de l'image ainsi que la mesure de distance en fonction de l'application visée. On pourra distinguer deux grands types d'approches de CBIR : les techniques de CBIR sur la base de descripteur d'images et les techniques sur la base d'apprentissage.

I.2.1 CBIR sur la base de descripteur d'images

La CBIR sur la base de descripteur d'images s'est révélée efficace et très utile dans de nombreuses d'applications. Ces descripteurs visent la plupart du temps à décrire des caractéristiques visuelles comme la couleur, la texture et la forme. Dans la littérature, il existe deux types de

Technique	Détection	Valeurs mesurées	Informations observées
Radiologie & Tomodensitomètre.	Rayon X	Énergie, nombre, position	Puissance d'atténuation des tissus
Médecine nucléaire	Rayon Gamma	Énergie, nombre, position	Distribution fonctionnelle et radioactivité
Échographie & Doppler	Ultrason	Intensité, Fréquence, Position	Réflectivité des tissus
Imagerie par résonance magnétique	Ondes électromagnétiques	Intensité, fréquence, position	Temps de relaxation, Structure moléculaire, Concentration protonique, Flux, diffusion
Optique	lumières infrarouges visibles ou proches	Intensité, diffusion, réflexion	L'absorption et la diffusion des tissus, l'oxygénation

TABLE 1.1 – Techniques d'observation les tissus mous, les structures rigides d'un flux massif de données

caractéristiques : globales et locales. Les premières sont calculées globalement sur l'image alors que les autres sont plus généralement utilisées pour reconnaître des objets plus précisément dans une image. La littérature sur ces descripteurs est très abondante et nous ne pouvons donner ici qu'une vue restreinte de ceux-ci.

* Descripteurs globaux

Ces derniers cherchent à décrire les couleurs, les textures et les formes présentent dans une image. Une définition de sens commun de la notion de texture est la suivante : la texture est la répétition d'éléments de base construits à partir de pixels qui respectent un certain ordre. On peut distinguer deux types extrêmes de textures, entre lesquels se positionnent toutes les autres. D'un côté on trouve les textures régulières, dans lesquelles la périodicité du motif est évidente (e.g. grilles, tissus) et qui peuvent être décrites par des approches fréquentielles (e.g. spectres de Fourrier, ondelettes de Gabor [32]) ou des approches structurelles dans lesquelles on associe un motif et des règles de placement sur un pavage régulier [33]. À l'autre extrémité, se positionnent les textures aléatoires qui peuvent être approchées par des lois statistiques, par exemple extraire des descripteurs à partir de l'histogramme de l'image comme la moyenne, la variance et l'entropie, ou bien de la matrice de cooccurrence [34] comme l'énergie, l'inertie et le moment différentiel inverse.

Les descripteurs d'une forme sont représentés sous la forme d'un masque dans lequel chaque pixel est représenté par le numéro de la région à laquelle il appartient. C'est à partir de cette fonction que sont calculés la plupart des descripteurs de formes, soit à partir de la région entière, soit à partir des contours seulement. Les descripteurs proposés sont souvent spécifiques à une forme particulière. Par exemple, le rapport iso-périmétrique, qui est un descripteur proportionnel au rapport du carré du périmètre de l'objet à sa surface est maximum, dans des images continues, pour le cercle. On peut citer quelques descripteurs classiques : i) Les moments d'inertie : Ils ont la propriété d'être invariants par rotation. Ils décrivent bien l'allongement de formes régulières

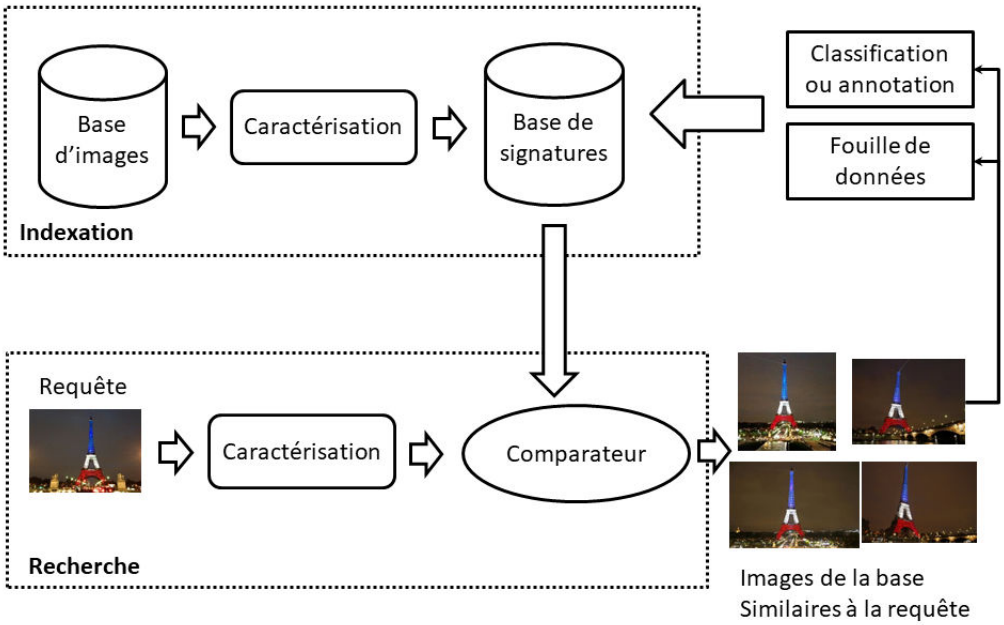


FIGURE 1.4 – Principe de CBIR

comme des ellipses ou des distributions gaussiennes. ii) Les moments invariants : (nommés moments de Hilbert), Ils sont invariants par translation, rotation et changement d'échelle. Cette approche consiste à envelopper l'objet et le reconstruire dans des boites de formes de plus en plus précisément adaptées. Pour plus de détails concernant les descripteurs de formes, le lecteur peut se reporter à [35–37]

En ce qui concerne la couleur l'enjeu porte sur le choix de l'espace de couleur le plus discriminant où calculer des descripteurs comme ceux évoqués plus haut. S'ajoute ensuite la prise en compte des problèmes d'invariance aux conditions d'illumination et de prise de vue. Chaque espace colorimétrique a des caractéristiques intéressantes. L'espace Rouge Vert Bleu (RVB) par exemple est très simple à utiliser. Cependant, dans cet espace l'information est très corrélée et sensible aux changements d'illumination. Il correspond peu à la perception humaine. D'autres espaces comme le HSV (« Hue-Saturation-Value » ou « teinte-saturation-intensité ») séparent mieux l'information et sont plus proches de la vision humaine. Les quaternions offrent aussi la possibilité de manipuler les trois simultanément. Récemment ont été proposé des différents descripteurs qui combinent les quaternions avec les théories des moments [38].

* Descripteurs locaux

L'extraction de descripteurs globaux ne permet pas une recherche efficace d'objets dans une image. Sur ce point, les descripteurs locaux sont plus efficaces, pour une complexité cependant plus coûteuse. Au cours des dernières années, des descripteurs locaux tels que SIFT (Scale-invariant feature transform) [39], HOG (Histogram oriented gradient) [40], SURF (Speeded Up Robust Features) [41] et BRISK (Binary Robust Invariant Scalable Keypoints) [42] ont été appliqués dans les systèmes de CBIR [43–45]. Les SIFT et SURF se sont montrés comme des descripteurs locaux robustes dans la reconnaissance d'objets [46, 47].

Les descripteurs SIFT sont la combinaison de : points d'intérêts identifiés à l'aide d'un détecteur invariant au changement d'échelle ; et, des descripteurs décrivant les distributions statistiques des gradients locaux calculés sur des régions d'intérêt circulaires de taille prédéfinies

centrées sur les points d'intérêts précédents. La distribution des gradients est mesurée selon un histogramme d'orientation à trois dimensions : la position, l'orientation et l'amplitude des gradients (voir Figure 1.5). La quantification de la position et de l'orientation des gradients rend le descripteur SIFT robuste aux petites déformations géométriques et à l'imprécision de la détection des points d'intérêt. Plusieurs améliorations ont été proposées comme la variante PCA-SIFT [48] applique une analyse en composantes principales sur les gradients. Cela permet de réduire la dimension du vecteur de caractéristiques et donc d'augmenter la vitesse de la recherche. Néanmoins, cette variante se révèle moins distinctive. Les descripteurs (SURF) proposés en 2006

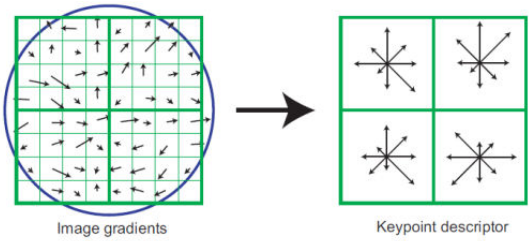


FIGURE 1.5 – Calcul des histogrammes des orientations dans 8 direction dans des fenêtres 4×4 autour du point d'intérêt

par Bay *et al.* [41] sont proches des descripteurs SIFT. Ils couplent une étape de « recalage » de la zone d'analyse (i.e. de la région d'intérêt) avec la construction d'un histogramme de gradients orientés. Sur la base de la transformée en ondelette de Haar, et de son premier niveau de décomposition qui donne accès aux dérivées premières de l'image sur un voisinage carré, cette étape de recalage cherche çà identifier la direction dominante des gradients sur laquelle s'alignée avant de calculer les histogrammes des gradients orientés. La Figure 1.6 schématise cette étape. En termes de performances, les descripteurs SURF égalent les SIFT en termes de répétabilité et de robustesse, mais les dépassent en vitesse d'extraction et de comparaison [49].

* Mesure de similarité :

Une fois les descripteurs calculés, se posent la question leur comparaison afin de compléter un processus de CBIR. Cette étape peut se faire en tenant compte du fait que chaque descripteur constitue une dimension de l'espace de l'ensemble des descripteurs. La mesure de distance couramment utilisées entre deux descripteurs est la distance Euclidienne. Cependant, lorsque l'on manipule des descripteurs comme les histogrammes ou des distributions statistiques, des distances comme celles de Minkowsky, de Battacharrya et la divergence de Kullbakc-Leibler sont utilisées. Dans le cas des partitions floues, on utilise des mesures de similarités comme l'indice de Rand flou proposé par Campello [50].

L'évaluation de la performance d'une technique de recherche par le contenu peut être calculer par la précision et le rappel. La précision est le nombre d'éléments pertinents retrouvés sur le nombre d'éléments total retourné par le système de recherche pour une requête donnée. Son principe est le suivant : quand un utilisateur interroge une base de données, il souhaite que les éléments en réponse à son interrogation correspondent à son attente. Tous les éléments retournés non pertinents constituent du bruit. Si elle est élevée, cela signifie que peu d'éléments inutiles sont proposés par le système et que ce dernier peut être considéré comme "précis". Le rappel est défini par le nombre d'éléments pertinents retrouvés au regard du nombre d'éléments pertinents que possède la base de données. Cela signifie que lorsque l'utilisateur interroge la base il souhaite voir apparaître tous les documents qui pourraient répondre à son besoin d'information.

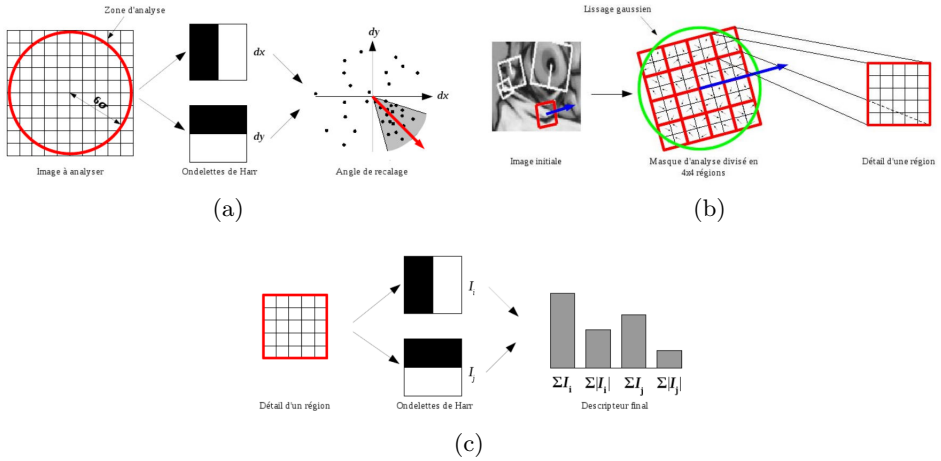


FIGURE 1.6 – (a) Détermination de l'angle de recalage du SURF (b) Masque d'analyse du SURF divisé en 4×4 régions (c) Extraction des différentes composantes du descripteur SURF par le biais des ondelettes de Haar dont la taille est égale à 2σ

I.2.2 Méthode de CBIR fondées sur l'apprentissage automatique

Le principe de l'apprentissage automatique (ou "machine learning » en anglais) est de permettre à un système piloté ou assisté par un ordinateur, d'adapter ses analyses, en se fondant sur une analyse empirique de données provenant d'une base de données. On peut catégoriser les algorithmes d'apprentissage selon le mode d'apprentissage qu'ils emploient :

— Apprentissage supervisé : dans ce cas, les exemples sont connus et leurs classes prédéterminées, i.e. les exemples sont étiquetés/labellisés par un expert. Le processus se passe en deux phases. Lors de la première phase (dite d'apprentissage), il s'agit de déterminer un modèle des données étiquetées. La seconde phase (dite de test) consiste à prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris. Parfois il est préférable d'associer une donnée non pas à une classe unique, mais une probabilité d'appartenance à chacune des classes prédéterminées.

— Apprentissage non supervisé (ou « clustering ») : cette fois le système ne dispose que d'exemples sans labels et le nombre de classes et leur nature ne sont pas prédéterminés. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Pour ce faire, il s'appuie sur une mesure de similitude généralement calculée selon une fonction de distance entre paires d'exemples

Les techniques d'apprentissage automatique reçoivent actuellement un intérêt croissant de la communauté en CBIR. Dans [51] les auteurs ont adapté une méthode par discrimination (Support Vector Machine) avec une méthode par modélisation (Mélange de gaussienne) au contexte de CBIR. L'objectif visé dans [51] est de classifier des images à partir d'une base de données supervisé. Dans [52], les auteurs proposent une approche basée sur les réseaux de neurones. La base de test est composée de 600 formulaires appartenant à 5 classes. Une précision de 92% est obtenue. Ces deux exemples montrent qu'il est nécessaire à la fois de connaître le nombre de classes afin de procéder à la classification supervisé. Il faut également que la base d'apprentissage soit de taille suffisamment importante afin de permettre un apprentissage performant.

I.2.3 Méthodes de CBIR en santé

Selon [53], les méthodes de recherche d'images par le contenu, spécifiques au domaine médical, peuvent être regroupées en plusieurs catégories :

- Les méthodes basées sur la segmentation de formes d'intérêt telles que des lésions [54] ou des régions [55,56]. En général, il n'est pas possible d'extraire automatiquement toutes les formes d'intérêt. Ainsi, des experts médicaux sont sollicités pour déterminer des régions d'intérêt (human/physician in the loop approach) [57].
- Les méthodes utilisant directement la description des lésions faite par les médecins [58,59].
- Les méthodes consistant à caractériser l'agencement des formes intéressantes (organes, lésions, ...) présentes dans l'image à l'aide d'un graphe topologique, qui sert alors d'index à l'image [54,60,61].
- Les méthodes basées sur l'extraction de descripteurs bas niveau connus pour bien caractériser les pathologies étudiées [62].
- Les méthodes basées sur la caractérisation de la couleur, de la texture ou de formes génériques [63,64]. Contrairement aux applications grand public, il est possible de sélectionner [65,66] ou de pondérer [67] les attributs discriminants pour prendre une décision car l'usage qui est fait de l'image est connue.

Il est important de noter que dans les applications médicales, la rotation et l'invariance au changement d'échelle sont des propriétés recherchées notamment s'il s'agit d'identifier des régions anatomique [21, 22], des lésions ou les modèles dans une ROI (region of interest) défini par l'utilisateur [25] ou segmentée automatiquement [24,31,68].

Dans ce mémoire, nous nous sommes intéressés à un système d'aide au diagnostic en rétinopathie qui recherche des images contenant des lésions similaires sans interactions avec l'utilisateur pour définir des régions d'intérêt fondé sur une segmentation automatique des lésions. En effet, les images contenant parfois des centaines de lésions, il n'est pas envisageable de solliciter l'utilisateur pour les délimiter. Également, il est difficile de concevoir un algorithme de segmentation robuste pour caractériser chaque type de lésions. Ce système s'appuie sur la méthode de CBIR proposée par Quellec *et al.* [7]. Elle permet de caractériser les images par leur contenu de texture à différentes échelles, autrement dit, elle caractérise des lésions de différentes tailles de manière générique. Cette méthode n'est qu'invariante à la translation, mais cela ne pose pas un problème car pour de nombreuses modalités d'imagerie médicale, comme l'imagerie de rétinopathie, les images sont orientées et mises à l'échelle lors du protocole d'acquisition. Par conséquent, la rotation et l'invariance de taille ne sont pas des problèmes réels dans notre application.

Dans la sous-section suivante nous proposons un scénario global pour externaliser un tel système de recherche par le contenu à des fins d'aide au diagnostic chez un fournisseur de cloud publique.

I.3 Scénario d'externalisation d'un système de recherche par le contenu

L'externalisation d'un système d'aide au diagnostic par présentation de cas similaires n'a pas, à notre connaissance, été étudié. Il s'agit en fait de pouvoir externaliser les différentes fonctionnalités d'un tel système pour qu'il profite des données des patients externalisées et mutualisées sur le cloud. Nous avons en conséquence identifié différents scénarios possibles d'externalisation des fonctionnalités d'une chaine de CBIR appelées par le système d'aide ; système mis à disposition sur le cloud par un tiers, i.e. ce n'est pas un service offert par le cloud. Nous décrivons ces différents scénarios ci-après de manière synthétique.

Les fonctionnalités d'un système de CBIR sont au nombre de trois :

1. Enregistrement d'un nouveau dossier d'examen – cette fonction permet d'enregistrer un nouveau dossier d'examen dans la base de données du serveur.
2. Récupération d'un dossier d'examen – il s'agit de récupérer un dossier d'examen stocké dans la base de données pour une seconde lecture.
3. Présentation des k examens les plus proches (aide au diagnostic).

Ces fonctions peuvent être réparties sur les différents systèmes d'information permettant l'externalisation des données, c'est-à-dire :

— Le système d'information du radiologue ou du médecin.

— Le nuage (« cloud »).

Elles sont sous le contrôle de différentes entités qui sont :

— Le radiologue (ou l'utilisateur).

— Le fournisseur de service de cloud.

— L'ingénieur système qui est rattaché à l'entité ou l'organisation qui offre le service d'aide. Celui-ci a un rôle d'intermédiaire entre l'utilisateur et le cloud au niveau des fonctionnalités du système de CBIR, e.g. mise à jour des paramètres du système de CBIR ou des identifiants (pseudonymes) liant les données aux patients et aux médecins.

I.3.1 Scénario d'externalisation d'un système d'aide au diagnostic de CBIR

Nous décrivons ici, les modes d'externalisation des différentes fonctionnalités de CBIR décrites ci-dessus que nous avons retenu. D'autres scénarios sont disponibles dans le rapport produit dans le cadre du laboratoire commun SePEMeD [3].

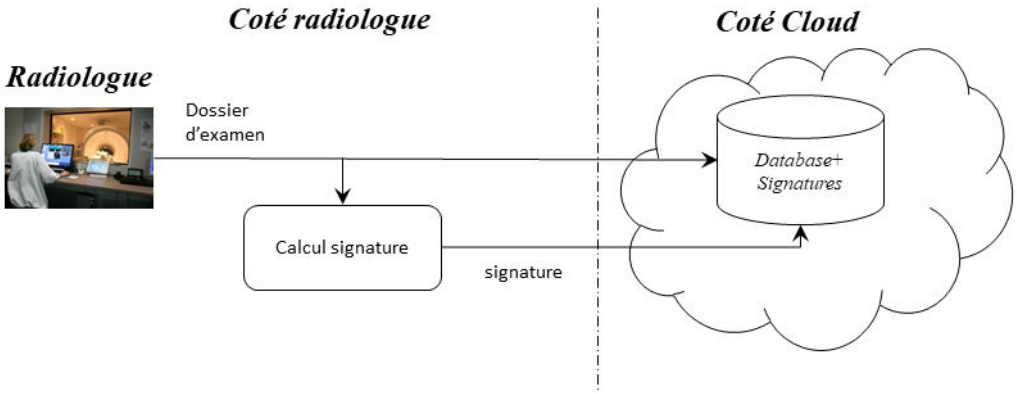


FIGURE 1.7 – Signature calculée chez le radiologue et sauvegardée chez le cloud

Cas d'utilisation 1 : Enregistrement d'un nouveau dossier d'examen Nous présentons en Figure 1.7, un premier schéma où le radiologue calcule la signature de l'image avant de l'externaliser avec le dossier d'examen. Ce scénario n'est pas pratique dans le cas où l'on doit mettre à jour la signature ou calculer une nouvelle signature sur la base d'une autre méthode. En effet, toutes les données sauvegardées dans le cloud doivent alors être rapatriées pour les calculs, ce qui n'a pas de sens. Nous n'avons donc pas retenu cette approche, lui préférant le scénario donné en Figure 1.8 où la signature est calculée par le cloud.

3. Référence projet : ANR-13-LAB2-0006

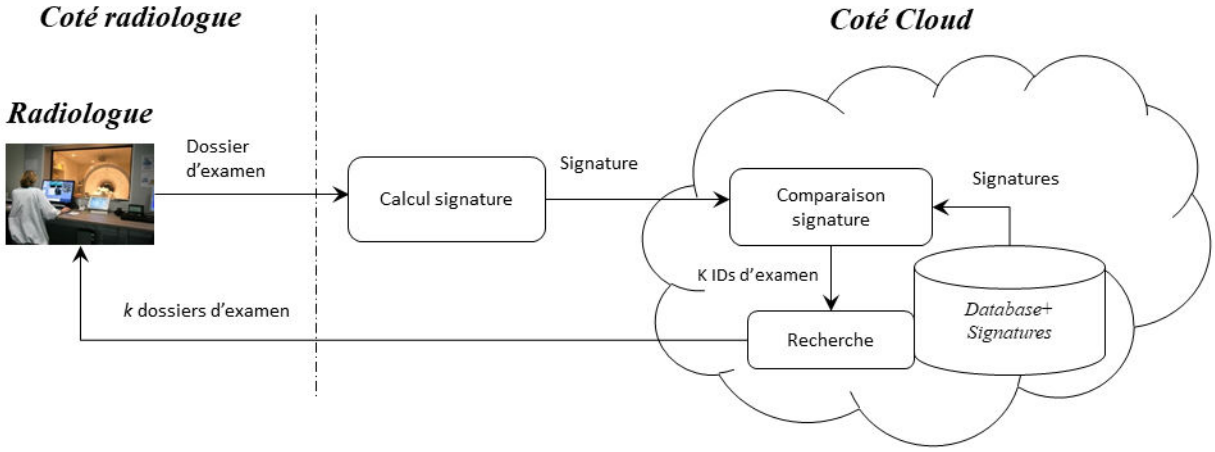


FIGURE 1.8 – Signature calculée par le cloud et sauvegardée chez le cloud

Cas d'utilisation 2 : Récupération d'un dossier d'examen (Seconde lecture – hors CBIR) Comme illustrée en Figure 1.9, l'externalisation de cette fonctionnalité est simple. Le radiologue réclame au cloud le dossier d'un patient sur la base d'une requête. Une hypothèse importante à considérer ici est que le radiologue qui veut récupérer un dossier d'examen n'est pas forcément celui qui a stocké ce dossier au départ (voir Figure 1.9).

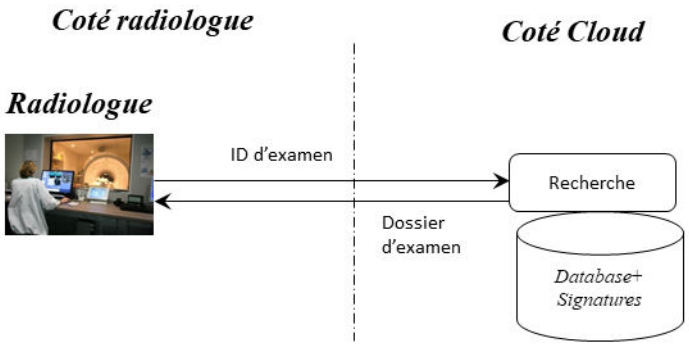


FIGURE 1.9 – Récupération d'un dossier d'examen stocké chez le

Cas d'utilisation 3 : Présentation des k dossiers d'examens les plus proches à un examen en requête La Figure 1.10, résume un scénario où le radiologue calcule la signature de son coté puis la présente sous forme d'une requête au cloud pour récupérer les k examens les plus similaires. Comme pour l'enregistrement d'un nouvel examen, cette solution n'est pas pratique. Le problème réside dans la mise à jour ou le calcul de nouveaux descripteurs. Pour éviter ce problème, nous proposons d'externaliser sur le cloud le calcul des signatures. Le radiologue présente donc en requête un examen et le cloud cherche les k examens les plus semblables et les retourne au radiologue (Voir Figure 1.11).

C'est ce scénario, le plus général, que nous avons retenu. D'autres solutions fondées sur le stockage de signatures et associés à des pointeurs sur les examens ont aussi été envisagées, mais elles ne tirent pas complètement avantage du potentiel du cloud computing.

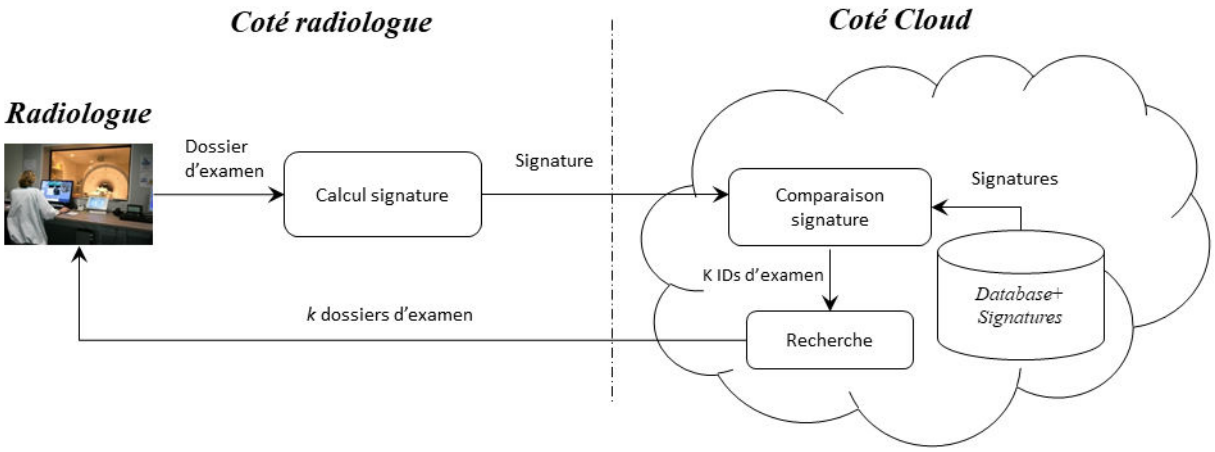


FIGURE 1.10 – Signature calculée par le radiologue

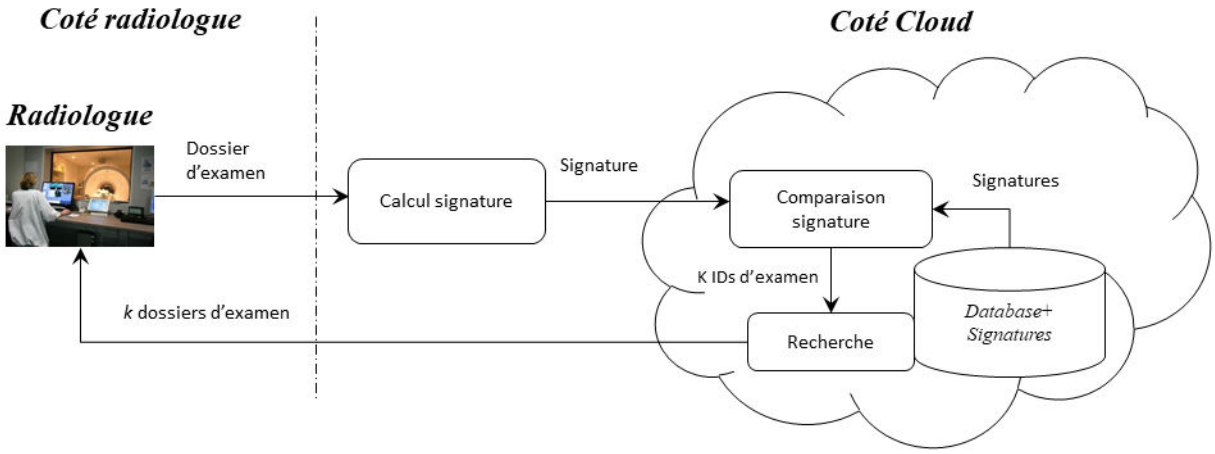


FIGURE 1.11 – Signature calculée par le cloud

II Besoins en sécurité des données de santé

II.1 La sécurité : une caractéristique de l'information médicale

Il est important de savoir ce qu'est l'information médicale, avant de pouvoir dire pourquoi la protéger et de quelle manière. Dans ce qui suit, nous abordons ces différentes questions.

II.1.1 Qu'est ce que l'information médicale ?

Diverses définitions sont données à l'information médicale (IM). Pour la Communauté Européenne, une information est dite médicale si elle réfère à toute donnée personnelle concernant la santé d'un individu, mais aussi lorsqu'il s'agit d'une information permettant de l'identifier [69]. Une autre définition, avec de fortes caractéristiques communes, est apportée par l'U.S Health Information Management Association qui considère comme information médicale toute donnée ou information quelques soient leurs supports, même oral, qui identifie ou permet d'identifier un

patient et : 1. qui sont relatives à l'état de santé d'un patient ; 2. qui sont obtenues du patient ou de ses proches. Nous pouvons d'ores et déjà souligner ici le caractère personnel et confidentiel de l'IM et du besoin de respecter le droit à la vie privé des personnes ("privacy"). Ainsi, la notion ou l'idée d'IM couvre toutes les données qui concernent l'historique ou le passé médical du patient, l'interrogatoire du patient et/ou de son entourage, les tests biologiques, les prescriptions et les résultats de techniques d'investigation. Les informations médicales sont donc très variées mais en même temps complémentaires. Elles contribuent entre autre à poser un diagnostic et/ou prendre une décision thérapeutique par le rapprochement des observations (e.g. ensemble de symptômes ou de signes) avec les connaissances médicales afférentes.

À l'hôpital, l'ensemble des informations médicales d'un patient est regroupé dans le dossier du patient qui constitue une obligation depuis 1993 [70]. L'Agence Nationale pour le Développement de l'Evaluation Médicale (ANDEM) qui est devenue en 1996 l'Agence nationale d'accréditation et d'évaluation en santé (ANAES) pour être enfin remplacée par la Haute Autorité de Santé en 2004, a défini le dossier du patient comme le support de l'ensemble des informations recueillies concernant la prise en charge du patient et dont les composantes (dossier médical, dossier de soins infirmiers, dossier administratif) intègrent des éléments communs, voire partagés. Selon l'article R.710–2-2 (décret num 92-329 du 30.03.92), trois composantes obligatoires doivent constituer a minima ce dossier du patient :

— Le dossier administratif qui doit comporter tous les éléments permettant d'identifier le patient, sa position administrative et sa couverture sociale.

— Le dossier médical qui peut être défini comme une mémoire écrite des informations cliniques, biologiques, diagnostiques et thérapeutiques d'un malade à la fois individuelle et collective, constamment mise à jour [71].

— Le dossier de soins infirmiers qui regroupe l'ensemble des informations concernant la personne soignée. Il contient notamment des informations spécifiques à la pratique infirmière comme l'application de soins.

II.1.2 Spécificités et qualités requises

L'information médicale est à « caractère sensible » pour les soins du patient et sa vie privée. Ces informations étant le plus souvent nominatives, elles sont confidentielles et leur accès limité aux seuls ayant-droits ; elles ne doivent pas être modifiées. Au-delà de ce premier point, la qualité des soins dépend fortement de la qualité de l'information. Dans [72], L. Dusserre définit les critères de qualité nécessaires de l'information médicale pour qu'elle soit valide. Il s'agit notamment de : la pertinence, la précision, l'actualité, la fiabilité, l'accessibilité, l'exhaustivité et la finesse de jugement du praticien. En ce qui concerne la fiabilité, elle correspond à la confiance accordée à une information et va donc avoir un lien direct avec les notions d'intégrité et d'authentification.

II.2 Contexte législatif et déontologique

II.2.1 Les besoins de sécurité : un cadre général

C'est le législateur qui impose la "sécurité" avec pour objectif d'instaurer une relation de confiance entre le patient, le professionnel de santé et le système de santé en général. En effet, un patient sera mieux pris en charge s'il donne une vue sans zones d'ombre de ses symptômes, de sa qualité de vie... Ce cadre législatif est très riche et s'exprime par le biais de nombreuses sources de réglementations (nationales et internationales, pénales ou déontologiques, ...), néanmoins toutes ont ce même objectif.

Pour donner le ton, regardons dans un premier temps l'obligation de sécurité en elle-même. Du fait qu'une information médicale est en générale nominative, il convient de respecter les articles 34 et 35 de la Loi «informatique et Liberté » [4] du 6 janvier 1978, qui traitent du droit du patient à la sécurité des informations nominatives et qui imposent en particulier au responsable du fichier, ou toute personne ordonnant ou effectuant un traitement, de s'engager à prendre « toutes précautions utiles, au regard de la nature des données et des risques présentés par le traitement, pour préserver la sécurité des données et, notamment, empêcher qu'elles soient déformées, endommagées, ou que des tiers non autorisés y aient accès. ». Le non respect de cette réglementation est passable de sanctions pénales, avec des amendes de 15000 à 300000 euros accompagnées de peines de prison de 1 à 5 ans. Il existe bien évidemment de nombreuses autres règles législatives comme déontologiques. Comme illustré en Figure 1.12, pour répondre à l'ensemble de ces obligations, le GMSIH suggère d'assurer des exigences de confidentialité, d'intégrité, de preuve et de contrôle et de disponibilité.

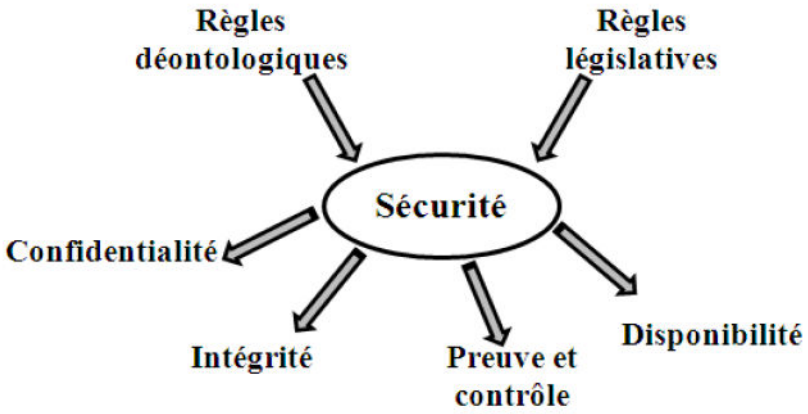


FIGURE 1.12 – Sécurité de l'information du point de vue GMSIH.

- Confidentialité

Cette propriété assure que l'information n'est accessible qu'aux seules personnes autorisées. Elle s'appuie sur la notion de « secret médical » qui, en France, comme l'indique l'article 4 du Code de Déontologie médical, s'exprime en termes de « secret professionnel ». Il est institué dans l'intérêt des patients et s'impose à tout médecin. Il couvre par ailleurs tout ce qui est venu à la connaissance du médecin. Faisant référence au secret professionnel, tout manquement est couvert par le code pénal qui sanctionne la divulgation (article 226-13), le détournement (article-226-21) et la divulgation d'informations à des entités inaptes à les recevoir (article -226-22).

- Intégrité

Différentes définitions sont données à l'intégrité. Nous n'en retiendrons que deux. Pour certains, l'intégrité est « stricte » et revient à assurer l'exactitude de l'information et des méthodes de traitement. C'est notamment le point de vue de la Loi informatique et libertés qui stipule qu'il revient au responsable de fichier de prendre « toutes précautions utiles afin de préserver la sécurité des informations et notamment d'empêcher qu'elles ne soient déformées, endommagées ... » (artilce 29 - couvert par l'article 226-17 du code pénal). D'autre comme J.M. Lamère [73],

4. la loi « informatique et Liberté » : Loi n 78-17 du 6 janvier 1978, modifiée par la Loi n 2004-801 du 6 août 2004, modifiée par la Loi n 2006-64 du 23 janvier 2006, modifiée par la Loi n 2008-696 du 15 juillet 2008, modifiée par la Loi n 2009-526 du 12 mai 2009 et modifiée par la loi n 2011-525 du 17 mai 2011.

voit l'intégrité comme « la propriété qui assure qu'une information n'est modifiée que par les utilisateurs habilités dans les conditions d'accès normalement prévues ». Cette dernière définition recouvre la précédente, nous suggérons de la retenir.

- Disponibilité

Cette propriété n'est pas explicitement présente dans les textes de lois. Elle est le dual de la confidentialité et peut être traduite comme « l'aptitude d'un système d'information à pouvoir être employé par les utilisateurs habituels dans les conditions d'accès et d'usage normalement prévues ». La disponibilité de l'information est cruciale dans certaines situations d'urgence. Son absence peut être la conséquence d'une destruction partielle ou totale de l'information ou d'une défaillance des mécanismes contrôlant l'accès aux données.

- Preuve et contrôle (Traçabilité)

La non-répudiation en fait partie. L'ANAP (Agence Nationale d'Appui à la Performance) définit cette exigence de sécurité comme « la preuve de l'origine ou de la livraison des données afin de protéger l'émetteur contre une fausse déclaration de non réception par le destinataire et le destinataire contre une fausse déclaration de non-émission par l'émetteur. Généralement établi par un tiers de confiance ». Ce sont donc les moyens de preuve et contrôle nécessaires qui peuvent aussi servir aux utilisateurs pour accorder leur confiance dans l'information fournie/reçue. Nous en avons peu parlé, au-delà de l'obligation légale et donc de la responsabilisation du professionnel de santé, se trouve la sanction pénale et ses implications financières. D'où l'intérêt des acteurs du système de soins de mettre en oeuvre la sécurité des données médicales.

Au final nous pouvons remarquer le lien qui existe entre la propriété de fiabilité d'une donnée médicale et les notions d'intégrité et d'authenticité. C'est à partir des preuves de l'intégrité de la donnée, de ses origines et de son attachement à un patient qu'un médecin décidera d'utiliser ou non cette information dans le cadre des soins ou plus largement de son activité.

Les solutions que nous avons étudiées dans ses travaux de thèse portent essentiellement sur la confidentialité et la fiabilité des données de santé, avec la volonté de les inscrire de comme complémentaire avec les solutions aujourd'hui disponible.

III Sécurité des données et des traitements

III.1 Outils de protection usuels

En pratique et usuellement, la sécurité passe sur la définition et le déploiement d'une politique de sécurité, par la suite déployée via la configuration de différents composants de sécurité du système. Dans cette section, nous revenons sur ce qu'est une politique de sécurité ainsi que sur les outils de sécurité les plus classiquement utilisés. Nous aborderons ensuite les solutions nouvelles permettant le traitement sécurisé de données, avant de présenter différents méthodes de CBIR sécurisées.

III.1.1 Notion politique de sécurité

La politique de sécurité spécifie les règles et les exigences de sécurité à satisfaire par le système d'information. Ces règles précisent, entre autres, qui peut ou non accéder à l'information, les procédures de reprises sur incidents de recrutement (enregistrement d'un nouvel utilisateur) et aussi comment les services de sécurité doivent être déployés/agencés/paramétrés/etc. La première étape dans le déploiement d'une politique de sécurité consiste en une analyse de risques à partir de laquelle des objectifs de sécurité seront définis. Pour ce faire, il est possible d'utiliser des référentiels comme la norme EBIOS (Expression des Besoins et Identification des Objectifs

de Sécurité) [74] qui fournit une liste de risques classés suivant les trois composantes de sécurité DICP (Disponibilité, Intégrité, Confidentialité, Preuve) pour des données stockées, traitées et communiquées en fonction de l'environnement où sont placées ces données. Elle a été utilisée dans le cadre des applications de télémédecine [75]. Il existe bien entendu d'autres approches comme MEHARI (MEthode Harmonisée d'Analyse de RIsques) [5].

L'analyse des risques est une partie du processus permettant les objectifs de sécurité à atteindre pour un SI. Ce processus dont nous donnons une vue synthétique est fondé sur les étapes suivantes :

— « caractérisation du système » qui permet d'identifier les activités et les ressources entrant dans le champ de l'analyse des risques.

— « Analyse de l'impact » et « impact » qui permettent d'évaluer l'impacte des sinistres potentiels sur les activités en utilisant l'échelle d'impact du GMSIH pour en déduire la sensibilité des ressources.

— « Analyse de la menace » et des « vulnérabilités retenues » qui permettent d'identifier des vulnérabilités (e.g. usurpation de droits) les plus pertinentes pour les différentes ressources du domaine de sécurité.

— « Détermination du niveau de besoin » qui permet d'évaluer des besoins de sécurité pour les différentes ressources en tenant compte des différentes vulnérabilités retenues.

Au sortir de ce processus, les exigences de sécurité sont établies et déployées en exploitant les différents moyens de sécurité existants pour contrer/réduire les risques identifiés.

En santé, les risques ne sont pas très différents d'autres domaines. Ces sont leurs probabilité de survenance qui sont différentes. Ils peuvent être classés en trois catégories : les accidents, les erreurs, les malveillances, et séparables suivant la nature des menaces (physique, technique, environnementale, humaine,...) [76]. Ces risques portent atteinte indépendamment ou conjointement aux quatre propriétés de sécurité DICP.

- Les accidents - Les problèmes liés à l'environnement du système d'information (SI) ou à son fonctionnement propre sont regroupés dans cette catégorie. Il peut s'agir de : dysfonctionnements du matériel ou des logiciels, de l'environnement technologique (coupure de courant, perte du réseau, support mémoire défaillant) ; la destruction partielle ou totale du matériel ; la négligence ou la défaillance/absence des personnels techniques. Ces risques seront toujours présents et on ne peut donc qu'essayer d'en limiter les conséquences.
- Les erreurs - Les sources d'erreur les plus fréquentes au niveau d'un SI sont les : erreurs de saisie ; erreurs de la transmission ; erreurs de manipulation des fonctions d'exploitation du SI ; erreurs résultant de la mauvaise utilisation du SI. Ces erreurs constituent un large spectre où la responsabilité des utilisateurs est importante mais où les défauts de conception des logiciels et des systèmes tiennent une place non négligeable.
- Les malveillances - Si les erreurs sont des risques identifiés, ce n'est pas le cas de la malveillance qui est inévitable. En effet, dès que le facteur humain est présent, il devient compliqué d'en évaluer la portée. La malveillance peut avoir pour finalité un chantage ou des intérêts économiques. Les altérations malveillantes peuvent aller de la suppression des preuves d'une erreur de prescription ou de diagnostic jusqu'à l'engagement de la responsabilité d'un tiers. Aujourd'hui l'accent est mis sur les risques de vols de données de santé. L'information médicale est une donnée personnelle à haute valeur économique pour nombre d'industrie (e.g. banque, assurance, industrie pharmaceutique).

5. Disponible sur : www.clusif.asso.fr.

III.1.2 Mécanismes de sécurité « classiques »

La phase de déploiement d'une politique de sécurité consiste à exploiter différents moyens pour sécuriser le système d'information (SI). Ces outils de protection peuvent être distingués en mécanismes de protection physiques et logiques. Les premiers concernent d'abord les matériels et visent essentiellement à contrer les divers risques naturels comme le feu et des accès physiques non autorisés. On parlera dans ce cas de contrôle d'accès physique par badge ou mesure biométrique pour filtrer l'accès aux salles informatiques.

Les mécanismes de protection logiques sont exploités au niveau logiciel. Ils sont nombreux et certains sont largement utilisés. Il s'agit par exemple : de l'authentification des utilisateurs (login, mot de passe, carte à puces) ; du contrôle d'accès (avec la mise en place d'une politique de contrôle d'accès de type R-BAC, Or-BAC, etc.) ; des techniques de chiffrement, des mécanismes de gestion de certificats pour la distribution des clés de chiffrement (e.g. « PKI » ou infrastructure à clés publiques), des mécanismes de filtrage réseau (« Firewall »), de traçabilité (e.g. logs générés et remontés par SYSLOG), de détection d'intrusion, etc.

Les outils cryptographiques jouant un rôle particulier, prenons le temps de définir les solutions les plus « classiques », i.e. les plus couramment utilisées, qui sont : le chiffrement et la signature numérique.

- Le chiffrement

Une opération de chiffrement transforme à l'aide d'un processus de chiffrement et d'une clé de chiffrement un texte en clair en un texte chiffré incompréhensible. Le texte chiffré ne peut être déchiffré avec le processus de déchiffrement que si on dispose de la clé de déchiffrement. Selon les principes de la cryptographie moderne, la sécurité d'un système de chiffrement repose sur la connaissance de la clé de chiffrement et non sur celle de l'algorithme de chiffrement (Principe de Kerckhoffs [77]). Il existe deux types de chiffrement en fonction de la dépendance entre les clés de chiffrement et de déchiffrement :

— Chiffrement symétrique : les clés de chiffrement et de déchiffrement sont identiques.

— Chiffrement asymétrique : les deux clés sont distinctes mais cependant liées mathématiquement. L'une est dite publique, accessible à tous, et l'autre privée, seulement connue d'une seule personne. Si la clé publique est utilisée pour le chiffrement, la confidentialité des données est assurée. Inversement, si la clé privée est utilisée, on obtient une propriété de non répudiation.

Des exemples d'algorithmes de chiffrement symétrique sont le DES ou le triple DES (Data Encryption Standard) et l'AES (Advanced Encryption Standard). Ils sont utilisés tous les trois dans le standard DICOM et préférés aux algorithmes de chiffrement asymétrique bien plus lents (100 à 1000 fois plus). Concernant les algorithmes de chiffrement asymétrique, on peut citer le RSA. Nous reviendrons sur ces algorithmes en sous-section III.2 où nous nous intéresserons aux algorithmes qui possèdent des propriétés d'homomorphie permettant d'effectuer des traitements sur des données chiffrées.

III.1.3 Signature numérique

L'intégrité d'un message transmis en clair peut être vérifiée en envoyant avec celui-ci une version de lui-même plus compacte (condensat, empreinte ou résumé) et chiffrée par un algorithme de chiffrement asymétrique : « une signature ». L'émetteur utilise dans ce cas sa clé privée. L'intégrité est assurée dans le sens où il n'est pas possible pour un tiers de modifier le message et de générer une « signature » valide sans accéder à la clé privée de l'émetteur. Comme en

plus, le texte chiffré n'est déchiffrable qu'avec la clé publique de l'émetteur, la non-répudiation et l'authentification de l'émetteur sont assurées.

Classiquement, l'empreinte d'un un message A est obtenue à l'aide une fonction de hachage cryptographique H comme par exemple le SHA [78] $H(A)$, qui tient en quelques centaines de bits. $H(A)$ est ensuite chiffré en utilisant un algorithme de chiffrement asymétrique. Le hash chiffré obtenu $DS(A)$ est appelé « signature numérique ».

Il nous semble important de souligner le caractère *a priori* de la protection offerte par les mécanismes cryptographiques. Une fois déchiffrée ou la signature supprimée, une image n'est plus protégée et il devient très difficile de vérifier son intégrité et ses origines. Nous le verrons plus loin, ils peuvent être avantageusement compléter par le tatouage qui offre lui une protection *a posteriori* des données.

III.2 Outils permettant le traitement sécurisé de données

Les solutions de protection précédentes sont appropriées lorsque les données sont concentrées sur un SI sous le contrôle du responsable des données. Elles présentent cependant des limites dans le cas où les données sont externalisées sur le cloud. En effet, l'utilisateur peut externaliser ses données sous forme chiffrées pour en assurer la confidentialité mais il devra les récupérer s'il veut les traiter. De nouvelles approches sont aujourd'hui étudiées pour permettre d'externaliser le traitement de données. Comme nous allons le voir elles s'appuient sur des techniques de chiffrement homomorphe, de calcul multipartite et aussi de tatouage de données qui offrent différents services de sécurité tout en permettant le traitement de données. Ces solutions prennent aussi en compte un modèle d'adversaire qu'il est important de préciser avant de les détailler.

III.2.1 Modèle d'adversaire et externalisation de données

Dans la littérature, il y a deux modes de confiance vis-à-vis de l'externalisation de données dans le cloud. Un cloud peut ainsi être :

- Honnête mais curieux – Encore désigné sous le terme passif ou semi-honnête, le cloud suit correctement les spécifications imposées par un protocole (e.g., un enchaînement d'actions nécessaires à la réalisation d'une fonction de CBIR) mais peut essayer d'obtenir des éléments d'information supplémentaires sur les données lors de l'exécution du protocole. Par exemple, il peut chercher à ré-identifier les patients ou connaître la pathologie associée à une image chiffrée.
- Malveillant - Ce type d'adversaire peut se comporter de manière arbitraire sans suivre les instructions d'un protocole. Par exemple, il peut corrompre la valeur de la sortie d'un traitement ou plus simplement faire échouer l'exécution du protocole.

Nous n'étudierons pas ce second mode car il existe un protocole de test assez efficace, dit « zero-knowledge » ou « Preuve à divulgation nulle de connaissance », qui permet d'identifier si un cloud est ou non malveillant. Par ailleurs, les différentes solutions de calcul multipartite, qui sont définies sur la base de protocoles et que nous verrons par la suite, peuvent passer du mode semi-honnête au mode malveillant assez facilement. Il suffit d'appeler le protocole zero-knowledge avant de lancer le calcul. À noter par ailleurs, qu'il est difficile d'imaginer qu'un fournisseur de services de cloud bien établi tente de falsifier des données de ses clients. Les dommages en termes d'images et de réputation vis à vis de sa clientèle seraient importants et irréversibles.

III.2.2 Chiffrement homomorphe

Le chiffrement homomorphe est l'un des nouveaux sujets les plus passionnants de la recherche en cryptographie et est une promesse pour un "Cloud Computing" parfaitement sécurisé. Il doit permettre à un utilisateur de stocker ses données chiffrées sur le cloud, et de demander à ce dernier de les traiter sans les déchiffrer, tout en obtenant le résultat du traitement lui aussi sous forme chiffrée.

Le concept de chiffrement homomorphe a été introduit par Rivest *et al.* [79]. Par définition, chaque opération algébrique effectuée dans l'espace de texte en clair correspond à une autre opération algébrique effectuée dans l'espace des chiffrés. Si m_1 et m_2 sont deux textes clairs, nous avons :

$$D[E[m_1] * E[m_2]] = m_1 + m_2 \tag{1.1}$$

où $D[.]$ et $E[.]$ sont les fonctions de déchiffrement et de chiffrement, respectivement. L'opérateur "$*$" désigne l'opération algébrique effectuée dans le domaine chiffré et "$+$" l'opération algébrique correspondante au domaine en clair.

Un exemple simple d'une application fondé sur le chiffrement homomorphe (CH) est donné en Figure 1.13. Dans ce scénario, le client C, chiffre d'abord ses données (étape 1) et les envoie au serveur du cloud, S (étape 2). Lorsque le client veut effectuer une fonction $f(.)$ sur ses propres données, il envoie la fonction au serveur (étape 3). Le serveur effectue une opération homomorphique sur les données chiffrées en utilisant la fonction *Eval* qui permet le calcul de la fonction f sur les données chiffrées sans en connaître le résultat (étape 4). Ce résultat chiffré est envoyé au client (étape 5) qui le déchiffre avec sa clé secrète et obtient $f(m)$ (étape 6). Comme on le voit dans cet exemple, l'opération *Eval* ne nécessite pas la clé privée du client. Les

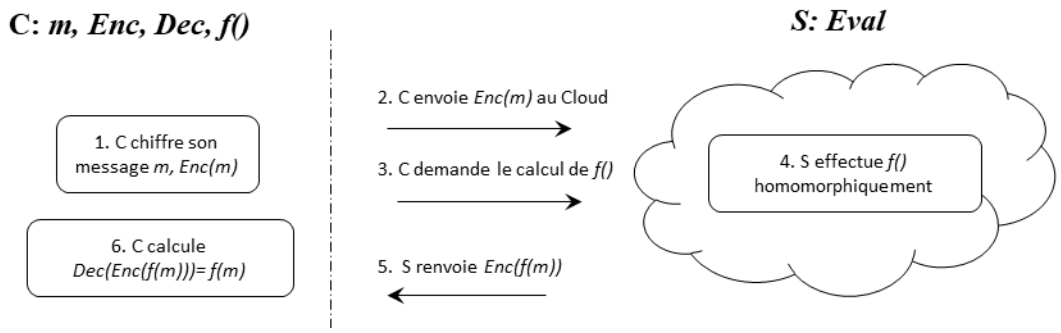


FIGURE 1.13 – un simple scénario fondé sur le CH, où C est le Client et S est le serveur

premières tentatives pour définir un cryptosystème homomorphe Rivest *et al.* [80] ; Goldwasser et Micali [81] ; ElGamal [82] ; Benaloh [83] ; Naccache et Stern [84] ; Okamoto et Uchiyama [85] ; Paillier [86] ; Damgärd et Jurik [87] ; Kawachi *et al.* [88] ; Boneh *et al.* [89] permettent une seule opération ou un nombre limité d'opérations sur les données chiffrées. En fait, toutes peuvent être classées en trois classes en fonction du nombre d'opérations (addition, multiplication, XOR, etc.) autorisées :

1. « Chiffrement partiellement homomorphe » ou "Partially Homomorphic Encryption" (PHE) qui permet un seul type d'opération qui peut être appliqué un nombre illimité de fois.
2. "Somewhat Homomorphic Encryption" (SWHE) permet plusieurs types d'opérations mais qui ne peuvent être appliquées qu'un nombre limité de fois. En effet, si ces schémas supportent par exemple l'addition et la multiplication, la taille des chiffrées croît après chaque

opération homomorphique, limitant le nombre maximal d'opérations homomorphes autorisées.

3. Fully Homomorphic Encryption (FHE) permet un nombre illimité d'opérations et un nombre de fois illimité. Le premier schéma FHE a été proposé par Gentry [90]. Fondé sur les réseaux idéaux ("ideal-lattices"), et son utilisation en pratique n'est pas faisable. Depuis de nombreuses améliorations ont été proposées.

Les schémas PHE sont déployés dans certaines applications telles que le vote électronique [91]. Cependant, c'est la popularité croissante des services fondés sur le cloud qui a accéléré la conception de schémas FHE qui peuvent supporter un nombre arbitraire d'opérations homomorphes avec des fonctions aléatoires.

Les schémas de chiffrement homomorphe Un schéma de chiffrement homomorphe (HE) se caractérise par quatre opérations : $KeyGen$, Enc, Dec et $Eval$. $KeyGen$ est l'opération qui génère soit une paire de clés publique et secrète pour un cryptosystème homormorphe asymétrique ; soit une seule clé pour un cryptosystème homormorphe symétrique. $Eval$ est une opération spécifique au chiffrement homomorphe qui prend en entrée des données chiffrées et produit un résultat chiffré. Il est important que la taille du chiffré après un processus d'évaluation reste la même pour permettre le déchiffrement et supporter un nombre illimité d'opérations. C'est la raison pourquoi la plupart des algorithmes FHE utilise une technique de "bootstrapping", une procédure de rafraîchissement intermédiaire d'un texte chiffré qui permet de maintenir la taille des chiffrés, au prix cependant d'une complexité plus élevée.

Les schémas PHE Il existe aujourd'hui une dizaine de cryptosystèmes PHE [80–88]. Certains d'entre eux n'ont pas à l'origine été développé pour faire du traitement sécurisé, comme par exemple :

- RSA

Du nom de ses auteurs Rivest, Shamir et Adleman, le RSA est un des premiers cryptosystème asymétrique à clé publique Rivest *et al.* [79] après l'invention du concept de cryptographie à clé publique introduit par Diffie et Helman [92]. La sécurité du RSA est basée sur le problème de la factorisation du produit de deux grands nombres premiers. La propriété d'homomorphie du RSA ont été démontrées par Rivest, Adleman et Dertouzous *et al.* [80]. Il s'agit d'un PHE multiplicatif, i.e. le produit des chiffrés permet de calculer le produit des clairs

- Le cryptosystème Goldwasser-Micali (GM)

Goldwasser et Micali ont proposé le premier schéma probabiliste de chiffrement à clé publique Goldwasser et Micali [81]. Il est sémantique sûr, c'est-à-dire qu'un texte en clair peut avoir plusieurs chiffrés différents. Cette propriété est importante et généralement obtenue par la prise en compte d'un aléa (i.e. une valeur aléatoire) dans la fonction de chiffrement Enc. La sécurité du cryptosystème GM est basée sur le problème de la résiduosité quadratique [93]. C'est un PHE additif qui permet des opérations sur des données binaires. Le produit des chiffrés équivaut au xor entre les clairs.

- Le Elgamal

Proposé en 1985, le Elgamal [82] est une version améliorée de l'algorithme original de Diffie-Hellman Key Exchange [92], qui repose sur le problème du logarithme discret [93]. C'est un PHE multiplicatif.

- Autres cryptosystèmes asymétriques avec des propriétés d'homomorphie

Algorithme de chiffrement	homomorphisme	Calcul
RSA [Rivest *et al* 1978]	Multiplicatif	Mod. Exp en $\mathbb{Z}_{pq}$
GM [Goldwasser et Micali 1982]	XOR	Mod. Exp en $\mathbb{Z}_{pq}$
ELGamal [ElGamal 1985]	Multiplicatif	Mod. Exp en $GF(p)$
Benaloh [Benaloh 1994]	Additif	Mod. Exp en $\mathbb{Z}_{pq}$
NS [Naccache-Stern 1998]	Additif	Mod. Exp en $\mathbb{Z}_{pq}$
OU [Okamoto-Uchiyama 1998]	Additif	Mod. Exp en $\mathbb{Z}_{p^2q}$
Paillier [Paillier 1999]	Additif	Mod. Exp en $\mathbb{Z}^2_{pq}$
DJ [Damgärd et Jurik 2001]	Additif	Mod. Exp en $\mathbb{Z}^s_{pq}$
Galbraith [Galbraith 2002]	Additif	Mult dans les courbes elliptiques
KTX [Kawachi-Tanaka-Xagawa 2007]	Additif	Réseau euclidien

TABLE 1.2 – Quelques algorithmes de chiffrement homomorphe partiel.

D'autres algorithmes ont été proposés depuis, améliorant les algorithmes ci-dessus et préservant leurs propriétés d'homomorphies. C'est le cas par exemple du cryptosystème Benaloh qui est une extension du cryptosystème GM fondé sur le problème de "Higher residuosity problem" [93], une généralisation du problème de la résiduosité quadratique. On pourra également citer les algorithmes d'Okamoto-Uchiyama (OU) [85] et de Naccache et Stern (NS) [84].

Des algorithmes de PHE développés spécifiquement pour faire du traitement de données chiffrées ont depuis été proposés. Le plus connu d'entre eux est le cryptosystème de Paillier [86]. C'est un PHE additif, probabiliste ou sémantiquement sûr, dont la sécurité repose sur le problème " composite residuosity problem" [93]. Damgärd et Jurik (DJ) [87] ont introduit un cryptosystème PHE qui porte leur nom et qui est une généralisation de celui de Paillier. Nous reviendrons sur ces algorithmes de chiffrement plus en détail dans les chapitre 2 et 2, où nous les exploitons. Galbraith [94] ont eux étendu le cryptosystème de Paillier aux courbes elliptiques tout en préservant la propriété d'homomorphie. Un autre schéma PHE additif a été suggéré par Kawachi (KTX) *et al.* [88], sa sécurité profite sur les problèmes de réseaux euclidiens. Ils obtiennent cependant une propriété pseudo-homomorphe, qui permet des opérations sur les texte chiffrés mais dont le résultat déchiffré peut comporter une erreur. Les propriétés d'homomorphie de l'ensemble de ces schémas PHE sont résumés brièvement dans la Table 1.2.

Les schémas SWHE Comme précédemment on peut trouver des cryptosystèmes d'abord proposés pour la confidentialité des données et qui montrent des caractéristiques de SWHE. C'est le cas par exemple du cryptosystème SYY [95]. Il est cependant difficilement utilisable pour faire du traitement de données chiffrées car la taille du texte chiffré augmente exponentiellement après chaque opération. Il faut attendre 2005, pour voir apparaître les premiers algorithmes SWHE. Le BGN, du nom de ses auteurs Boneh-Goh-Nissim [89], est le premier algorithme de ce type. Il prend en charge un nombre arbitraire d'additions mais ne permet de faire qu'une seule multiplication en maintenant la taille du texte chiffré constante. Sa sécurité est fondée sur le problème de décision de sous-groupe [93], qui consiste à décider si un élément est un membre d'un sous-groupe G_p du groupe G d'ordre composé égale à pq, où p et q sont des nombres distincts. À noter que le BGN a été une des étapes les plus significatives vers un schéma FHE.

On peut citer également le cryptosystème IP [96]. Il a cependant le défaut de voir la taille des chiffrés augmenter après chaque opération. Nous résumons dans la Table 1.3, les propriétés de ces différents algorithmes.

Algorithme de chiffrement	Fonction *Eval*	La taille du chiffré
SYY [Sander *et al.* 1999]	And illimité / un seul OR/NOT	augmente exponentiellement
BGN [Boneh *et al.* 2005]	Addition illimitée/ une seule multiplication	constante
IP [Boneh *et al.* 2005]	arbitraire	augmente linéairement

TABLE 1.3 – Comparaison de certains schémas SWHE bien connus avant le travail de Gentry

En conséquence, Gentry, Halevi et Vaikuntanathan ont ensuite simplifié le cryptosystème BGN [97]. Dans leur version, l'hypothèse de sécurité est modifiée en s'appuyant sur le problème LWE (Learning with errors). Le cryptosystème BGN choisit l'entrée d'un petit ensemble pour déchiffrer correctement. En revanche, le schéma introduit dans [97], ils ont augmenté la taille des messages en clair.

Les schémas FHE C'est Gentry qui a proposé le premier FHE [90]. Les travaux de Gentry débouchent non seulement un schéma FHE fondé sur les réseaux idéaux, mais aussi un cadre théorique général pour définir un schéma FHE. Sa solution présente cependant le défaut d'une complexité de calcul très forte du fait qu'elle repose sur les problèmes "ad hoc problem" et "spare subset sum problem" (SSSP), et ne peut pas satisfaire les exigences d'applications pratiques. Néanmoins, son cadre théorique est à l'origine de nombreux schémas FHE qui peuvent être utilisés.

Aujourd'hui, le chiffrement complètement homomorphe se composent de trois familles. La première regroupe les schémas FHE fondés sur le schéma initial de Gentry [98–102] et sont de fait de très forte complexité.

La seconde classe est constituée des schémas FHE travaillant sur des nombres entiers [103–107]. Ces solutions offrent des performances de mise en œuvre relativement efficaces, mais leur sécurité repose sur un problème faible (le "partially approximate common divisor" (PACD)).

La troisième classe regroupe les algorithmes qui s'appuient sur le problème « Learning with errors » ("LWE") ou de « Ring LWE » (une variante de LWE) [108–114].

Aujourd'hui, ce sont les solutions de la dernière classe qui sont les plus performantes en termes de complexité, de la taille du chiffré et en termes de sécurité. Dans ces travaux de thèse, nous nous sommes intéressés en particulier au cryptosystème de BGV (Brakerski - Gentry - Vaikuntanathan) [115] implémenté par IBM sous forme d'une librairie HElib [116, 117] avec des optimisations importantes (e.g. re-linéarisation, Bootstraping, squashing, batching...). Nous reviendrons sur cet algorithme dans le chapitre 5.

III.2.3 Calcul multipartite sécurisé (SMC)

Le chiffrement homomorphe ne permet de faire toutes les opérations, notamment les opérations non linéaires comme la comparaison et la division. Le calcul multipartite sécurisé (SMC) est une solution pour réaliser ces autres traitements. Par définition, le SMC permet à un ensemble de parties ou d'entités (a minima un client et un serveur) de faire un calcul de manière interactive avec en entrée leurs données privées sans qu'aucune des parties ne déduise d'informations sur les entrées des autres parties. Le résultat est connu pour tout le monde. Il peut être, par exemple, un booléen, ou l'indice de l'élément le plus proche dans la base de données, ou une liste d'indices des éléments les plus proches. Il existe plusieurs SMC élémentaires utiles dans les traitements de données chiffrées.

Oblivious Transfer (OT) C'est un protocole introduit par Rabin [118] en 1981. Un OT_1^N est une primitive cryptographique qui permet à un récepteur d'obtenir un éléments sur N détenus par un expéditeur, sans savoir ou apprendre des informations sur les autres éléments de l'expéditeur et sans que l'expéditeur sache quel élément a été choisi. La fonctionnalité du OT_1^N est la suivante : un expéditeur possède une liste de N éléments $\{x_1, x_2, .., x_N\}$ et un récepteur possède un indice $i \in \{1, ..., N\}$. À la fin du protocole, le récepteur reçoit x_i, sans que l'expéditeur connaisse i et le récepteur n'a aucune information sur les x_j, pour $j \neq i$.

Protocole de Yao' (Garbled Circuit) En 1986, Andrew Yao a présenté le « Garbled Circuit » [119] qui permet à deux entité de collaborer pour calculer correctement la sortie d'une fonction sans que les deux parties aient besoin de révéler leurs entrées. Un exemple courant de ce problème est le problème du Millionnaire, dans lequel deux millionnaires veulent déterminer qui est le plus riche, sans qu'aucun d'eux ne révèle à l'autre combien il a d'argent. Yao modélise ce problème comme une série de portes binaires prenant en entrée des données chiffrées avec des algorithmes de chiffrement classiques (e.g. AES). Considérée comme théoriquement intéressante, cette solution reste cependant trop coûteuse en calcul et ne s'applique qu'à un nombre limité de problème. Elle a depuis été améliorée en utilisant le chiffrement homomorphe pour un nombre élargi d'applications [120]. Les dernières approches modélisent la fonction comme un circuit booléen partagé entre les entités impliquées, et chiffrent les entrées et sorties de chaque porte afin que l'entité qui exécute une partie du traitement ne puisse extraire d'informations sur les entrées ou les valeurs intermédiaires.

Partage du secret Le partage du secret est une technique introduite par Shamir [121] en 1979, par lequel une valeur donnée (le secret) est divisée entre plusieurs entités de telle sorte que la coopération entre un certain nombre de ces entités est nécessaire pour récupérer le secret. Aucune entité ne peut avoir accès au secret toute seule. Plus formellement, on parle de partage de secret à n participants avec un seuil t de sorte que :

— t personnes prises parmi ces n participants peuvent toujours reconstituer l'information secrète.

— $(t-1)$ personnes prises parmi ces n participants ne peuvent jamais reconstituer l'information secrète.

La solution proposée par Shamir utilise des polynômes de degré t et l'interpolation de Lagrange pour distribuer des données à n personnes. Supposant que la valeur secrète est r, un corps finis est choisi de manière à ce que le secret soit de la taille d'un élément du corps. Par exemple, un secret de 128 bits donne le corps $K = \mathbb{F}_2^{128}$. Pour que le secret soit retrouvable par au moins t personnes parmi n on choisit un polynôme f sur $K[x]$ de degré $t-1$ tel que $f(0) = r$ (il suffit de fixer la constante de f à r).

$$f(x) = r + \sum_{i=1}^{t-1} b_i X^i \quad b_i \in \mathbb{F}_2^{128} \tag{1.2}$$

On distribue alors aux n personnes une valeur $f(a_i)$ pour des a_i distincts et non nuls. Si t personnes collaborent, alors ils sont capables de reconstruire f grâce à l'interpolation de Lagrange car f est de degré $t-1$. Si moins de t personnes collaborent, ils retrouveront un polynôme à une constante indéterminée près. Comme le secret est justement une constante, ils ne peuvent pas retrouver d'informations supplémentaires sur le secret.

III.2.4 Proxy re-encryption (PRE)

Le partage de données chiffrées est une fonction importante notamment si l'on souhaite utiliser des données externalisées par des personnes différentes. Ce type de problème est traité à l'aide de systèmes de type « proxy re-encryption (PRE) », où Alice (le délégateur ou l'émetteur) veut partager avec Bob (le délégué ou le destinataire) certaines données chiffrées qu'elle a précédemment externalisées dans le cloud (le Proxy). Lorsque le chiffrement asymétrique est utilisé, l'objectif du proxy est de re-chiffrer le texte chiffré d'Alice, $m \in M$ chiffré avec sa clé publique, dans un texte chiffré qui peut être déchiffré avec la clé privée de Bob. Pour ce faire, Alice génère une clé de re-chiffrement $rk_{A->B}$ que le proxy va utiliser pour convertir un son texte chiffré c_A sous la clé publique pk_A d'Alice en un autre texte chiffré c_B sous la clé publique pk_B de Bob $m \in M$. Bob sera capable d'obtenir le message en clair m à l'aide de sa clé privée sk_B. Lors de l'exécution d'un schéma PRE sécurisé, un attaquant (par exemple le proxy) ne doit pas pouvoir déduire d'informations sur le message m ou les clés privées (sk_A ou sk_B). La définition d'un PRE proposé dans [122] correspond à un jeu de cinq fonctions :

- $KeyGen(k)$: Algorithme de génération de clé - il prend en entrée le paramètre de sécurité k, et il retourne en sortie une paire de clés publique/privé (pk, sk).
- $ReKey(pk_A, sk_A, pk_B, sk_B)$: Algorithme de génération de la clé de re-chiffrement - il prend en entrée les clés de l'émetteur et le du destinataire (la clé privée du destinataire est optionnelle), il retourne en sortie une clé de re-chiffrement $rk_{A->B}$. Cette fonction est effectuée par Alice (émetteur).
- $Encrypt(m, pk)$: Algorithme de chiffrement - cette fonction permet de chiffrer le message m par la clé publique pk pour générer un message chiffré c.
- $ReEncrypt(c_A, rk_{A->B})$: Algorithme de re-chiffrement - qui prend entrée le message chiffré de Alice c_A et la clé de re-chiffrement $rk_{A->B}$. Cette fonction est effectué par le Proxy pour transformer le message chiffré c_A à c_B via la clé de re-chiffrement.
- $Decrypt(c, sk)$: Algorithme de déchiffrement - cette fonction prend en entrée un message chiffré c et une clé secrète sk, elle retourne un message en clair m.

Afin de partager des données, ces fonctions sont utilisées comme suit :

$$Decrypt(sk_A, Encrypt(pk_A, m)) = m \tag{1.3}$$

$$Decrypt(sk_B, ReEncrypt(ReKey(pk_A, sk_A, pk_B, sk_B), Encrypt(pk_A, m))) = m. \tag{1.4}$$

Aujourd'hui, la plupart sinon tous les PRE sont dérivés d'un algorithme de chiffrement à clé publique comme indiqué dans [122]. Ce sont les deux fonctions ReEncrypt et ReKey qui différencient un PRE du cryptosystème à clé publique associé. Nous reviendrons dans le chapitre 5, où nous présentons la conception d'un nouveau schéma PRE homomorphe (HPRE).

III.2.5 Tatouage/ Protection *a posteriori*

Le tatouage ou le marquage de données offre une solution complémentaire au chiffrement de données. Il rentre dans le contexte plus général de la dissimulation d'informations qui inclue aussi la stéganographie, du grec écriture cachée. Si la stéganographie a pour objectif la communication secrète entre deux personnes, en dissimulant un message utile dans un document hôte anodin, le tatouage a pour but de protéger le document (image, vidéo, musique, ...) dans lequel le message est dissimulé à des fins, par exemple, de protection de la propriété intellectuelle. Pour les images, le message est inséré par modification aussi imperceptible que possible des niveaux de gris de l'image. Le signal de différence entre l'image originale et sa version tatouée est ce qu'on appelle la

marque qui est associée au message tatoué. Telle que définit, la protection assurée par le tatouage est indépendante du format de stockage des données (la protection est dans la donnée elle-même). C'est une protection *a posteriori*, la donnée peut être utilisée tout en la maintenant protégée par la marque. C'est une solution intéressante dans le cas de l'externalisation de données.

Le schéma général d'une chaine de tatouage d'images est donné en Figure 1.14. Les principales opérations du processus du tatouage des images sont :

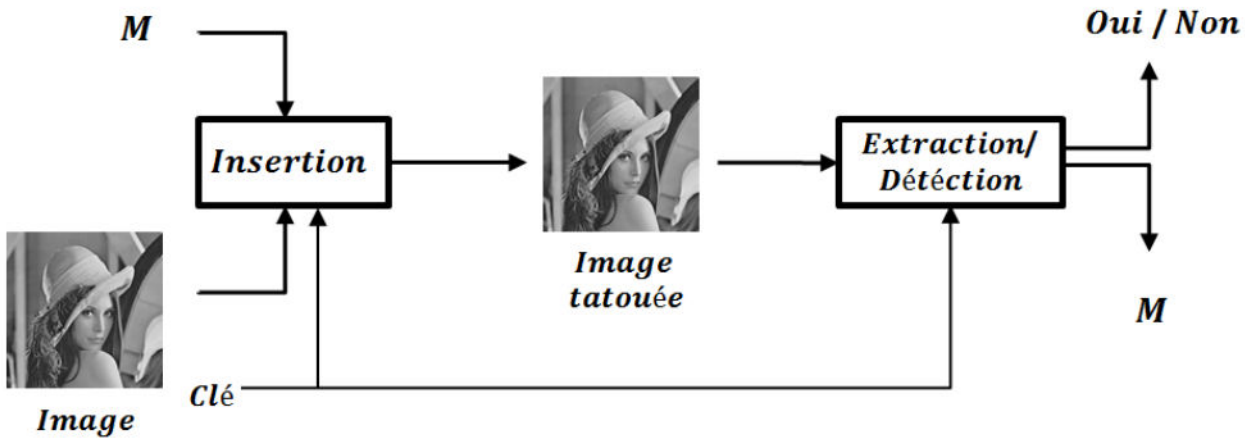


FIGURE 1.14 – Schéma générale du processus du tatouage

- **L'insertion** - Cette étape permet d'insérer un message dans une image par modification des niveaux de gris de cette image ou des coefficients d'une transformée (TCD, ondelettes, ...). Elle dépend d'une clé secrète de tatouage qui permet par exemple de sélectionner les pixels à tatouer.
- **La détection et/ou l'extraction** - Cette étape dépend de la clé de tatouage. La marque insérée peut être détectée et/ou extraite suivant la nécessité d'avoir ou non l'image originale. On distinguera une détection/extraction aveugle ou non. Une méthode est dite aveugle si elle ne nécessite pas la présence de l'image originale pour extraire le message, semi aveugle si elle nécessite un résumé de l'image, ou non-aveugle sinon.

Le tatouage a d'abord été proposé pendant les années 90 pour répondre aux problèmes de la protection des droits d'auteurs de documents numériques. Avant de distribuer son œuvre, l'auteur y insère une marque, e.g. un message constitué de son nom ou de sa signature, que lui seul est capable d'extraire. En cas de litige, il sera à même d'apporter la preuve de sa propriété. Depuis, le tatouage a été suggéré pour répondre à d'autres objectifs de sécurité tels que :

- **Le contrôle d'intégrité** - À partir du message tatoué, le destinataire peut vérifier que l'image n'a pas été modifiée pendant sa transmission. Pour ce faire, le message peut être la signature numérique (voir section III.1) de l'image. En comparant la signature recalculée à celle extraite de l'image il est possible de détecter une falsification de données.
- **Authenticité** – En santé il s'agit dans ce cas d'apporter la preuve de l'origine d'une donnée et de son attachement à un patient donnée. Il s'agira par exemple de l'identifiant unique de la modalité d'acquisition (UIN DICOM) et d'un identifiant patient (e.g. un pseudonyme hospitalier ou son numéro de sécurité sociale).

- **Traçabilité** – L'objectif est ici de pouvoir tracer la donnée tout au long de son existence. Cela peut se faire en permettant l'insertion d'un message dans une image portant l'identifiant d'un noeud dans un réseau par exemple. Il s'agit d'une généralisation du concept de tracé de traître (ou « traitor tracing ») ou de « fingerprinting » considéré dans des applications de vidéo à la demande, où l'intérêt porte sur l'identification du client ou du vendeur à l'origine d'une redistribution illégale d'une vidéo.

Au-delà de ces services de sécurité, le tatouage permet aussi l'ajout de nouvelles fonctionnalités aux contenus multimédias comme :

- **L'indexation de documents** - Le message inséré dans le document est une information décrivant son contenu. Pour les images, il peut s'agir d'un pointeur vers un lien Internet qui décrit le détail de l'image.
- **L'ajout de méta-données** - Prenant le domaine médical en exemple, Deux médecins peuvent échanger le diagnostic d'un certain patient par le biais de ses images.

Propriétés Nombreuses et différentes sont les techniques qui existent pour tatouer des images. Ces méthodes sont caractérisées par un compromis établi entre différentes propriétés parmi lesquelles on trouve :

- **L'invisibilité** C'est la propriété la plus importante notamment en santé. La marque doit être invisible pour ne pas interférer avec le diagnostic.
- **La capacité d'insertion** : Elle correspond au le taux d'insertion exprimé en nombre de bits de message enfoui par pixel de l'image (*bpp* pour "bit per pixel"). C'est un indicateur sur la taille du message qu'on peut insérer dans une image.
- **La robustesse** : Un tatouage est dit robuste si après toute modification de l'image tatouée, on peut toujours retrouver la marque dissimulée. Ces modifications peuvent être innocentes, prévues dans le cycle d'exploitation des données (e.g : compression, opérations de rehaussement de contraste, transformations géométriques ...) ; ou malveillantes, c-à-d. qui cherchent à supprimer ou à modifier la marque. À l'opposé de la robustesse on trouve la fragilité, une propriété utile pour détecter une modification des données, la marque disparaitra.
- **La sécurité** : – L'accès aux données insérées peut être conditionné à la connaissance d'une clé de tatouage. Comme pour la cryptographie, il existe des méthodes de marquage symétrique où les clés d'insertion et d'extraction sont les mêmes, et asymétrique où les clés sont différentes. Ces derniers schémas sont assez rares.
- **La complexité** : C'est une indication sur les temps nécessaires pour effectuer l'insertion et l'extraction du message. Il faut en tenir compte lorsque l'on doit protéger des images très volumineuses.
- **La réversibilité** : Cette propriété garantie qu'il est possible de restaurer les niveaux de gris de l'image originale après extraction de la marque.

Le degré d'importance de chacune de ces propriétés dépend du contexte applicatif (contrôle d'intégrité, protection de la propriété intellectuelle, ...). Ainsi, une méthode de tatouage est choisie en fonction du compromis qu'elle établit. Par exemple, dans le cas de contrôle d'intégrité, la méthode de tatouage peut être fragile contrairement à la protection du copyright où la marque doit être robuste.

Type de tatouage : Il existe deux grandes familles de modulations de tatouage, qui sont parfois combinées pour gagner en robustesse, par exemple. On distinguera ainsi les modulations de tatouage additives et des modulations substitutifves, appliquées directement à l'image ou à une transformée de celle-ci (transformée en cosinus discret, transformée en ondelette ...).

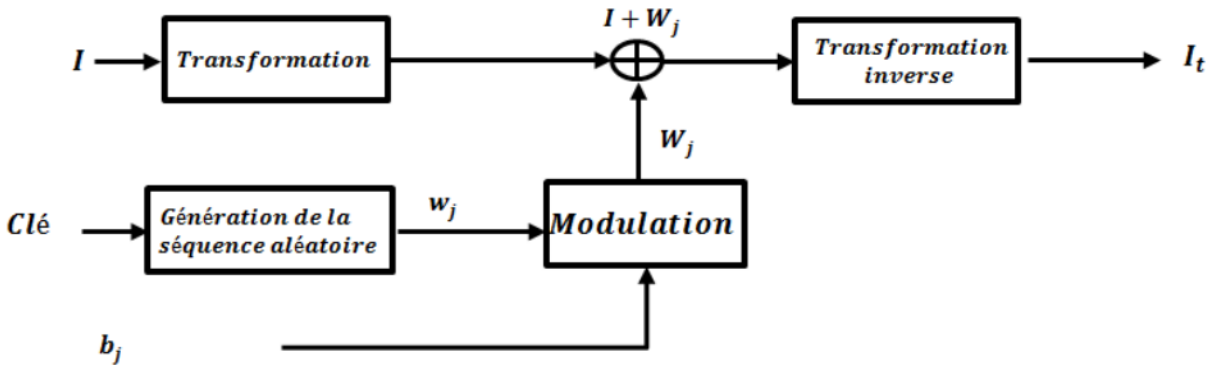


FIGURE 1.15 – Schéma d'insertion sur la base d'une modulation de tatouage additive

— **Tatouage additif** : Les algorithmes de cette famille utilisent une marque, un signal, engendrée à partir du message (une séquence de bits). La génération se fait par un étalement de spectre où chaque bit b_j est associé à la valeur $d_j = 1 - 2b_j$, multipliée par une porteuse w_j (une séquence de bits) de faible énergie ensuite ajoutée à l'image I pour obtenir l'image tatouée I_t. La formule d'insertion est alors : $I_t = I + \alpha d_j w_j = I + W_j$, où α est un paramètre de force d'insertion ou d'incrustation (paramètre de robustesse). Plus α est grand, plus que la robustesse augmente plus la marque devient visible. La Figure 1.15 illustre le principe d'un schéma de marquage utilisant ce type de modulation :

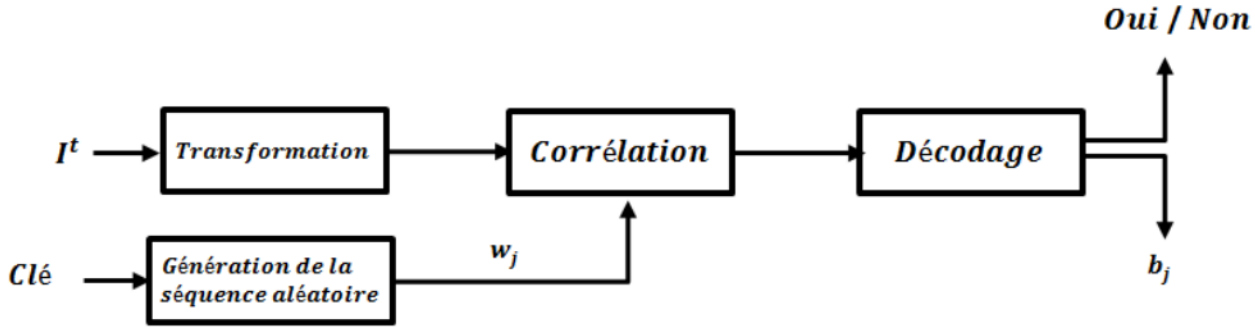


FIGURE 1.16 – Schéma d'extraction d'un message sur la base d'une modulation de tatouage additive

La détection de la marque, illustrée en Figure 1.16 , se fait par corrélation entre l'image tatouée reçue I'_t et la marque W_j, Considérant l'orthogonalité des porteuses w_j, le signe de chaque produit de corrélation donnera la valeur du bit enfoui qui est nul si la marque n'existe

pas. Ainsi, si W_j est présente, on aura $b_j = 0$ si $< I'_t, W_j >=< I, W_j > + < W_j, W'_j >= 0 \pm |W_j|^2$ est strictement positif et $b_j = 1$ sinon.

A noter que pour gagner en robustesse tout en maximisant l'invisibilité de la marque, les techniques actuelles exploitent des masques psycho-visuels de manière à adapter la force d'insertion localement à l'image. Ces masques profitent par exemple du fait que l'oeil humain est peu sensible aux variations de textures.

— **Le tatouage substitutif** : Les algorithmes de cette classe sont très simples de par leur principe. Des « caractéristiques » de l'image, comme ses niveaux de gris ou des coefficients d'une transformation de celle-ci, sont substituées par d'autres issues de dictionnaires codant les symboles (e.g. des bits) du message à insérer. L'extraction du message passe par une simple interprétation des caractéristiques de l'image à l'aide des dictionnaires utilisés à l'insertion.

L'algorithme le plus célèbre et le plus simple de cette famille est la méthode de substitution des bits de poids faible (LSB). Elle consiste simplement à remplacer les bits de poids faible des pixels de l'image par les bits du message. Cette méthode est cependant fragile, le moindre traitement d'image altérera le message, mais elle fournit une forte capacité d'insertion de *bpp*. Dans le chapitre 5, nous utiliserons la modulation par quantification d'index (QIM « Quantization Index Modulation ») qui est une généralisation de la méthode des LSB.

Tel que définit, si le tatouage ne permet pas d'assurer la confidentialité des données, il offre cependant des services de sécurité comme le contrôle d'intégrité et la traçabilité. Il laisse la donnée accessible tout en la maintenant protégée. Avant d'externaliser les données sur le cloud, l'utilisateur peut par exemple les tatouer avec l'identifiant du fournisseur de services à qui il confie ses données. Il pourra par la suite l'identifier s'il retrouve ses données distribuées de manière illégale.

Aujourd'hui, il y a un intérêt à pouvoir combiner le tatouage avec le chiffrement de données pour accéder à des services de tatouage à partir de données chiffrées pour les tracer ou vérifier leur intégrité [6]. Nous y reviendrons dans le chapitre 5 où nous présentons une approche de tatouage de données chiffrées de manière homomorphe.

III.2.6 Compromis entre sécurité et efficacité de traitements

Les solutions précédentes (chiffrement homomorphe, SMC, PRE, tatouage ...) permettent de réaliser des traitements de données sécurisés. Cependant, chacun d'eux impose aussi des contraintes. Par exemple, les approches SMC nécessitent des communications et l'implication de plusieurs parties voir des tiers de confiance. Le chiffrement homomorphe requiert des fortes capacités de stockage et de puissance de calcul. Il ne permet pas les traitements non-linéaires, c'est pourquoi il est souvent couplé au SMC dans des applications. Les coûts en communications et en calculs peuvent ne pas être négligeables en fonction de l'application visée. Par ailleurs, une perte de communication peut empêcher l'utilisation de l'application.

Dans le même temps, le chiffrement ne permet pas de travailler avec des nombres réels. Ils fonctionnent avec des entiers. C'est pourtant le type de données le plus courant dans les applications de traitement du signal. Par conséquent, la version sécurisée d'un traitement n'aura pas la même précision que dans le domaine en clair.

Comme nous le verrons dans la section III.3, s'il est possible de réaliser des traitements sur des données chiffrées les résultats de ces traitements sécurisés peuvent être en clair. Il convient donc de s'interroger sur la sécurisation d'un traitement dans sa totalité, i.e. à toutes ses étapes. En effet, chaque étape d'un traitement peut laisser fuiter une information exploitable par un

adversaire mettant en danger toute la sécurité du système. Une conséquence immédiate est une augmentation des coûts de complexité, de stockage et de communication.

Ainsi en pratique, la sécurisation d'application de traitements du signal nécessite de trouver un compromis entre l'efficacité d'un traitement sécurisé (i.e. sa complexité de calcul et de communication), les performances de traitement (précision obtenue par rapport aux résultats en clair) et le niveau de sécurité ou de confidentialité atteint. Dans la section suivante, nous abordons l'état de l'art des solutions proposées pour sécuriser les algorithmes de recherche par le contenu (CBIR). Nous verrons que ce compromis joue un rôle clé.

III.3 CBIR sécurisé

Différentes méthodes de CBIR sécurisées ont été proposées dans la littérature. Elles peuvent être différenciées selon le type de signature extraite des images (i.e. locale ou globale) et en fonction de la manière de calculer et de comparer ces signatures et par qui (l'utilisateur ou le cloud).

Une des premières solutions consiste à externaliser une image avec sa signature calculée *a priori* [123]. Dans ce scénario, le serveur (le cloud) effectue uniquement la comparaison des signatures. Les images sont chiffrées avec des algorithmes tels que l'AES ou 3-DES. Le fait de chiffrer les images avec de tels cryptosystèmes ne permet pas de manipuler les données chiffrées dans le cloud. Le processus de CBIR est donc partagé entre le client et le serveur. Une alternative est d'extraire des signatures ou des caractéristiques directement à partir des images chiffrées. Pour ce faire, la plupart des solutions utilisent le chiffrement homomorphe et le calcul multipartite sécurisé. Celles-ci ne concernent que les techniques de CBIR à base de descripteurs locaux. Il n'existe pas de méthodes sécurisées de CBIR à base de descripteurs globaux. Nous étudierons cette problématique dans les chapitres 2 et 3.

Nous proposons de distinguer ces méthodes en deux classes. La première regroupe des schémas où la signature est calculée par le client puis chiffrée homomorphiquement et envoyée à un cloud qui de son côté possède une base de données d'images en clair. La deuxième classe est constituée des méthodes où les images sont stockées chiffrées sur le cloud ; cloud qui calcule également les signatures des images, y compris celles des images envoyées en requête sous forme chiffrée par le client.

Méthode de CBIR sécurisée et partagée entre client-serveur. Il est important de souligner que ces méthodes ont été proposées dans le contexte de l'authentification/identification biométrique. À notre connaissance, Erkin et al. [124] ont été les premiers à sécuriser un tel protocole de reconnaissance de visage fondé sur l'algorithme de reconnaissance « Eigenfaces ». Ils utilisent un cryptosystème homomorphe additif et le SMC entre deux entités : le client qui veut savoir si une donnée biométrique correspond à un ou plusieurs enregistrements dans la base de données d'un serveur. Le protocole consiste à demander au serveur de calculer la distance euclidienne entre le vecteur d'Eigenfaces extrait de l'image et chiffré par le client avec ceux de la base de données du serveur. Le serveur renvoie les informations de profil associées à l'enregistrement qui a la plus petite distance par rapport aux données biométriques fournies par le client. À l'issue du protocole, le client ne connaît que sa requête et les profils retourné par le serveur. Le serveur n'a aucune idée de la requête de l'utilisateur. Sadeghi *et al.* [125] ont amélioré cette solution. Celle-ci utilise également un chiffrement homomorphe additif pour calculer de manière sécurisée la distance euclidienne mais utilise le Garbled Circuit [126, 127] pour trouver la distance minimale. De leur côté, Evans *et al.* ont proposé un protocole d'identification biométrique sécurisé basé sur la reconnaissance d'empreintes digitales [128]. Le protocole fournit la même

garantie de sécurité que les protocoles mentionnés précédemment. S'il combine également un chiffrement homomorphe additif avec le Garbled Circuit, ils améliorent l'efficacité tant pour la phase de calcul de distance et pour la pour la recherche de minimum. Blanton et Gasti [129] ont développé un protocole sécurisé d'identification de l'oeil basé sur la distance de Hamming. À nouveau cette solution couple le chiffrement homomorphe additif et le Garbled Circuit. Cependant la complexité des circuits utilisés pour la comparaison basée sur des portes XOR est réduite. SCiFI [130] est un système d'identification de visage sécurisé. La technique de représentation d'image de visage adoptée est robuste aux différentes conditions d'acquisition telles que l'illumination, les occlusions et les changements d'apparence (comme le port de lunettes). Le protocole utilise la distance de Hamming pour mesurer la similitude entre les images et sa mise en oeuvre repose sur le chiffrement homomorphe additif et l'Oblivious Transfert [131].

Bien que les protocoles précédents d'authentification biométrique préservent la protection de la vie privée, le serveur a accès à toutes les données stockées dans sa base de données. Pour éviter cela, Blanton et Aliasgari [132] ont développé une approche sécurisée où les données externalisées sont chiffrées. Ils suggèrent en fait deux protocoles. Un est basé sur un seul serveur et le deuxième sur plusieurs serveurs. Dans le premier cas, le protocole utilise un schéma de chiffrement prédictif [133,134] qui permet au serveur d'effectuer des calculs sans interactions avec le client. Le système de chiffrement prédictif n'est pas aussi sûr que le chiffrement homomorphe additif. Dans le cas plusieurs serveurs, leur solution s'appuie sur le partage du secret pour "chiffrer" la base de données biométriques externalisée. Néanmoins, ce protocole nécessite au moins trois serveurs indépendants pour effectuer les calculs intermédiaires. Comme nous le verrons dans les chapitres suivants, les solutions auxquelles nous sommes arrivées ne nécessitent qu'un seul serveur sans communications avec le client.

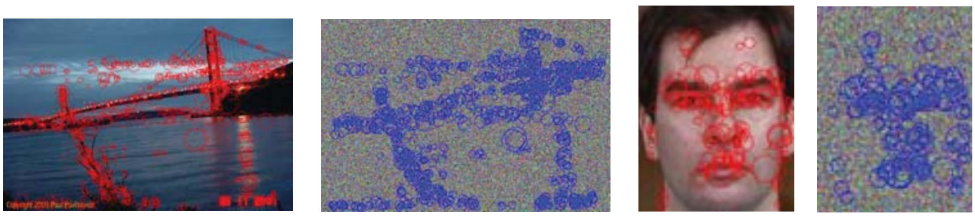
Méthodes de CBIR sécurisées sur le cloud Ces solutions extraient des descripteurs locaux comme les SIFT et les SURF d'images chiffrées. Toutes les fonctionnalités d'un système de CBIR sont externalisées mais elles nécessitent des interactions entre le cloud et le client ou entre plusieurs cloud.

Sécurisation des descripteurs SIFT. Le calcul des descripteurs SIFT a été sécurisé. Dans [1], ils sont extraits d'une image chiffrée avec le cryptosystème de Paillier. Cette méthode nécessite un « tour » de communication entre le serveur et l'utilisateur lors de l'extraction des caractéristiques et bien plus pour comparer les signatures d'images. Il a été montré que ce schéma est inutilisable dans des applications réelles et qu'il a certaines faiblesses de sécurité [135]. Les auteurs de [135] améliorent ce schéma avec l'aide d'un cryptosystème complètement homomorphe et du calcul multipartite sécurisé, mais cependant avec comme conséquence une grande complexité de stockage, de calcul et de communication. Une autre amélioration suggérée dans [136] repose sur l'utilisation d'un algorithme de chiffrement qui préserve l'ordre après le chiffrement (Order-preserving encryption (OPE) [137]), un chiffrement non-homomorphe, et trois fournisseurs de cloud indépendants. Il convient de noter que les deux solutions données dans [135] et [136] ne conservent pas bien la performance du SIFT dans le domaine en clair. Pour résoudre ces problèmes, [138] et [139] proposent un SIFT sécurisé fondé sur deux fournisseurs de cloud indépendants. Si la complexité de la communication avec l'utilisateur est réduite, ils supposent que les deux serveurs ne colludent pas.

Sécurisation des descripteurs SURF Comme les systèmes précédents, une première solution [140] extrait les descripteurs SURF d'images chiffrées homomorphiquement à l'aide du cryptosystème de Paillier. Cette solution nécessite des communications entre le serveur et

l'utilisateur pour exécuter certaines opérations de base dans le domaine chiffré que le serveur ne peut pas conduire seul sans avoir accès aux données en clair (par exemple, division, racine carrée...) lors de l'extraction de la signature d'une image. Dans [141], les autheurs ont proposé une solution afin de réduire les besoins en communication de [140] à l'aide d'un cryptosystème SWHE et de deux fournisseurs indépendants de cloud. Mais l'utilisation d'un cryptosystème SHE limite le nombre d'opérations d'additions et de multiplications que l'on peut exécuter à plusieurs reprises sur les images chiffrées. De ce fait, les résultats des calculs ne peuvent pas être ré-exploités facilement.

Au-delà du fait que ces systèmes de CBIR sont sécurisés et nécessitent des communications entre l'utilisateur et un ou plusieurs serveurs, ils calculent des signatures sous forme non chiffrées. Même si le contenu de l'image n'est pas disponible, connaître les caractéristiques SIFT, par exemple, donne des idées sur le contenu de l'image. Ces caractéristiques sont en effet des points d'intérêt dans l'image. Connaître leurs positions peut contribuer à la reconnaissance d'objets ou de personnes dans une image chiffrées. Nous donnons en Figure 1.17 un exemple d'une telle situation. Il apparaît donc nécessaire d'extraire des signatures d'images elles-mêmes chiffrées. Nous avons abordé ce type dans [2–4], et nous y reviendrons dans le Chapitre 3. Il également important de souligner le fait que ces techniques considèrent que toutes les images sont chiffrées avec la clé publique. Elles n'envisagent pas d'utiliser des données chiffrées par des utilisateurs différents. C'est pourtant le cas en santé où le cloud stocke des dossiers de patients externalisés par plusieurs professionnels de santé. Des solutions de partage de données chiffrées homorphiquement sont nécessaires. L'objectif est de permettre le partage de données sans les rapatriées, les re-chiffrées et les externaliser à nouveau pour les traiter. Nous proposons la première solution de ce type en Chapitre 3. A noter également qu'aucun algorithme de CBIR fondé sur des signatures



(a) (b) (c) (d)

FIGURE 1.17 – Exemples illustratifs issus de Hsu *et al.* [1] de la détection des caractéristiques SIFT dans le domaine en clair (a), (c) et dans le domaine chiffré (b), (d).

globales n'a été sécurisé. Il s'agit d'extraire certaines caractéristiques globales qui résument par exemple une image à la distribution de ses coefficients d'ondelettes dans différentes sous-bandes. Comme nous l'avons vu en Section I.2, ce type de signature se montre plus optimale en termes de performances de recherche lorsque la texture de l'image joue un rôle majeur, ce qui est souvent le cas en imagerie médicale.

IV Conclusion

Comme nous l'avons vu dans ce chapitre, l'externalisation et la mutualisation concernent de nombreux domaines y compris celui de l'imagerie médicale qui évolue aujourd'hui vers le « medical cloud imaging » . Au-delà de la réduction des coûts de maintenance et des services, l'intérêt est aussi de faciliter et d'améliorer la prise en charge des patients par le biais du partage

de données et la mise à disposition de nouveaux services comme des outils d'aide au diagnostic fondés sur la réutilisation des données d'imagerie par le biais techniques de recherche d'images par le contenu (CBIR) et d'apprentissage automatique (ML) qui sont au cœur des systèmes d'aide au diagnostic. Dans ce chapitre, nous avons proposé un scénario d'externalisation d'un système d'aide au diagnostic fondé sur la CBIR. Nous nous servirons de ce dernier comme cadre de travail et d'expérimentation.

Dans un tel environnement, la sécurité des données est un enjeu majeur imposée par le citoyen. En environnement comme le cloud, la confidentialité et le respect du droit à la vie privée sont essentiels, mais il ne faut pas négliger l'intégrité et la traçabilité de données. Pour pouvoir réutiliser les données, le traitement et le partage de données chiffrées sont des éléments clés. L'objectif est de définir des approches capables d'exploiter des données externalisées et sécurisées, notamment sous forme chiffrées, et de permettre à un cloud de trouver des données semblables sans accéder à leur contenu réel. Nous avons montré que les solutions existantes qui répondent à ce problème de sécurité nécessitent beaucoup de communications et souvent plusieurs entités (interactions « cloud-utilisateur » ou entre plusieurs clouds). En cas de pertes de réseaux ces solutions restent limitées. Par ailleurs, les signatures extraites ne sont pas chiffrées et peuvent donner des indices à un cloud honnête mais curieux. À noter également qu'avec ces solutions, les données sont chiffrées avec la même clé. Elles ne permettent aussi d'extraire que des signatures locales qui ne sont pas forcément adaptées à l'imagerie médicale. Il faut trouver des solutions originales adaptées à l'imagerie médicale tant pour l'extraction de signature que pour la réutilisation de données chiffrées par des utilisateurs différents. Ce sont là les objectifs de ces travaux de thèse que nous abordons dans les prochains chapitres de ce manuscrit.

Dans le chapitre 2, nous avons travaillé sur la sécurisation d'un schéma de CBIR global à l'aide du chiffrement homomorphe sous les contraintes : de « zéro » communication entre l'utilisateur et le serveur du cloud ; et la réalisation de l'ensemble des traitements par un seul cloud. Nous avons choisi de sécurisé la technique de CBIR proposée par Gwénolé *et al.* [7] de par son adéquation aux images médicales et ses performances. Dans ce chapitre nous proposons une méthode originale, rapide permettant de comparer des données chiffrées.

Dans le chapitre 3, nous abordons une limite du schéma précédent liée à la sécurité de la signature qui est non chiffrée et qui peut laisser fuiter de l'information sur les données. Nous proposons deux autres approches qui permettent de travailler avec des signatures sécurisées. La première est basée sur le chiffrement homomorphe et deux fournisseurs de cloud indépendants. La seconde s'appuie sur le chiffrement homomorphe mais cette fois avec un seul fournisseur de cloud.

Le chapitre 4, porte sur la sécurisation d'un système d'apprentissage automatique (le perceptron multicouches) et en particulier la sécurisation de sa phase d'apprentissage. Une telle approche n'existe pas aujourd'hui. Elle exploite le chiffrement homomorphe et des techniques de calcul sécurisées que nous avons développées dans nos recherches.

Enfin, dans le chapitre 5, nous proposons le premier système de partage de données chiffrées homomorphiquement par utilisateurs différents (i.e. des clés différentes). Dans cet environnement externalisé, il convient également de pouvoir s'assurer de l'intégrité des données qu'elles soient ou non chiffrées. Pour ce faire, nous proposons une technique de tatouage d'images chiffrées.

Chapitre 2

Système de recherche par le contenu externalisé

Comme nous l'avons vu, les besoins de sécurité décrits dans le chapitre précédent constituent un cadre général qui doit cependant être précisé en fonction du contexte applicatif. Dans le cadre de la sécurisation de systèmes d'aide au diagnostic externalisés et fondés sur la CBIR, il convient d'assurer en premier lieu la confidentialité des données et le respect du droit à la vie privée. Il est alors nécessaire de déployer ces solutions sur des données chiffrées. En ce qui concerne l'imagerie médicale, nous avons vu l'inexistence de techniques de recherche par le contenu qui permettent d'extraire des signatures globales à partir d'images chiffrées. C'est un des objectifs de ces travaux de thèse.

Dans ce chapitre, nous présentons un système de recherche par le contenu sécurisé (SCBIR) externalisé sur le cloud sur la base du chiffrement homomorphe et en particulier du l'algorithme de Paillier, un cryptosystème de type PHE. Notre objectif est d'extraire des signatures globales d'images chiffrées sous la contrainte d'utiliser un fournisseur de cloud et « zéro » communication entre l'utilisateur et le serveur pendant les traitements. Plus clairement, dans notre système, le calcul et la comparaison de signatures sont entièrement effectués par le serveur (ou de manière équivalente le cloud) sans communication avec un tiers de confiance (i.e. l'utilisateur ou un autre serveur). Notre solution profite d'une nouvelle façon que nous proposons pour comparer rapidement les données chiffrées de Paillier sans interactions, même si les données sont chiffrées avec différentes clés publiques.

Avant d'aborder cette solution, nous revenons dans la première partie de ce chapitre sur la définition d'un framework d'externalisation de CBIR avec pour objectif d'identifier les entités participantes et leur rôle comme les menaces qui pèsent sur la confidentialité des images externalisées. Nous aborderons ensuite, les principes de calcul de l'extraction d'une signature globale et la comparaison des signatures dans le domaine en clair. Avant de présenter notre système de recherche par le contenu sécurisé, nous ferons quelques rappels sur les outils sur lesquels il s'appuie, notamment le cryptosystème de Paillier, accompagné d'un état de l'art sur les méthodes de comparaison sécurisées. Nous en profiterons pour décrire une des originalités de ces travaux qui porte sur la comparaison rapide de données chiffrées. Ce chapitre se conclura par l'analyse des performances de notre système en terme de sécurité, de complexité et de performance de retrouvaille en comparaison avec la version de ce système réalisé dans le domaine en clair.

I Définition du framework d'externalisation de CBIR

L'application visée est l'aide au diagnostic où le système d'information ou le logiciel d'images d'un médecin interroge un système distant pour obtenir une idée de l'interprétation possible d'une image avant que le médecin ne se connecte au logiciel. Sur cette base le médecin pourra alors confirmer ou invalider son diagnostic pour le nouveau cas qu'il n'a pas encore regardé en détail. Pour ce faire, et comme illustré en Figure 2.1, le système envoie l'image acquise au système distant qui est en charge de retrouver les K images les plus similaires avec leurs diagnostics associés.

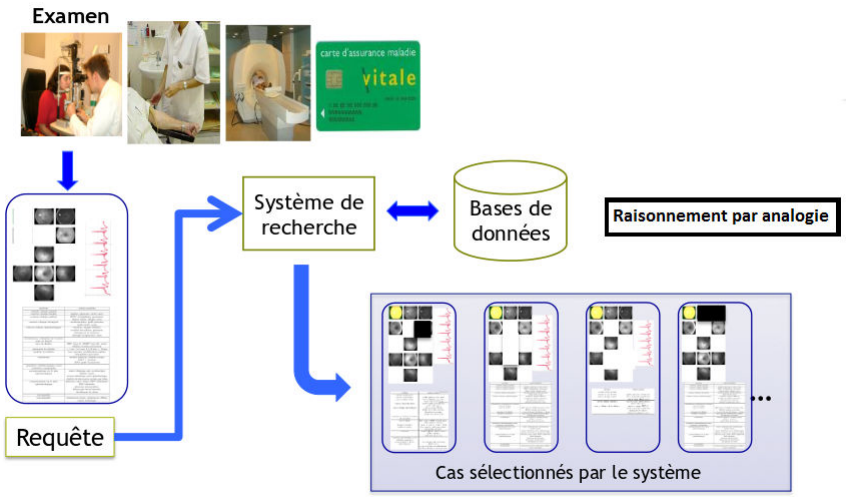


FIGURE 2.1 – Système d'aide au diagnostic

Dans le scénario que nous avons étudié, ce schéma s'appuie sur le *Cloud – Computing*, où le système distant est le serveur (cloud) qui possède une base de donnée d'images et qui a également une capacité de calcul et de stockage importante. En vis-à-vis, un utilisateur présente des requêtes au serveur et attend en retour des images semblables.

Dans ce schéma, le serveur ne peut accéder au contenu des images. Celles-ci sont confidentielles. Elles sont donc stockées et transmises sous forme chiffrées. Ce qui implique que l'utilisateur (médecin) a une capacité de chiffrement et déchiffrement. Il n'a cependant pas les capacités de stockage et de calcul du cloud. Le serveur compare l'image requête avec toutes les images qui existent déjà dans sa base de données et renvoie les plus similaires. La Figure 2.2 présente le déroulement de ce scénario. Dans un tel scénario du cloud publique, les problèmes de sécurité des données se multiplient notamment en termes de confidentialité et de respect de la vie privée. En effet, les utilisateurs perdent le contrôle sur les données qu'ils externalisent [142]. L'externalisation des données dans un cloud publique n'est pas sûr contre des menaces à la confidentialité, celles-ci peuvent être externes (par exemple, pirates [143]) ou internes [144, 145]. Il y a donc un intérêt à développer des méthodes d'extraction d'image basée sur le contenu externalisé sécurisé (SOCBIR).

L'hypothèse est faite que le cloud calcule les signatures de toutes les images qu'il reçoit. Il peut stocker celles-ci ou les recalculer si besoin. Ce cloud nous le considérons comme honnête mais curieux (cf. chapitre 1 section III.2). Plus clairement, le cloud suit correctement les spécifications

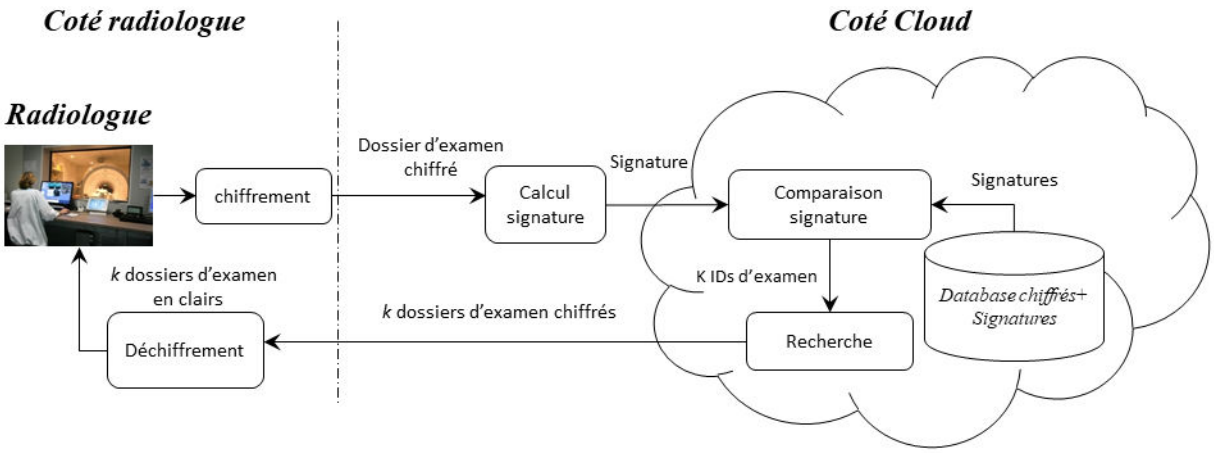


FIGURE 2.2 – Scénario du système CBIR sécurisé

imposées pour la réalisation d'une fonction de CBIR mais il peut essayer d'obtenir des éléments d'information sur les données lors de l'exécution du protocole CBIR. Dans ce mode d'adversaire, le cloud peut par exemple, chercher à ré-identifier les patients ou connaître la pathologie associée à une image chiffrée. Nous reviendrons sur ce type d'attaque lors de l'analyse de sécurité de notre système. Rappelons que si le cloud est un adversaire de type malveillant, il peut se comporter de manière arbitraire sans suivre les instructions du protocole. Par exemple, il pourrait corrompre la valeur de la sortie d'un traitement ou plus simplement faire échouer l'exécution du protocole. Comme discuté en chapitre 1 section III.2 ; il existe un protocole de test assez efficace, dit « zero-knowledge » ou « Preuve à divulgation nulle de connaissance », qui permet d'identifier si un cloud est ou non malveillant. Au-delà, ce type de comportement de la part du cloud est peu plausible dans le scénario qui nous concerne ; le cloud perdrait son image de marque et ses clients. Par conséquent, nous n'étudierons pas ce second mode d'attaquant.

Comme évoqué plus haut, le schéma de CBIR que nous avons retenu a été proposé par Quellec *et al* dans [7]. Cette solution s'appuie sur la comparaison de signatures calculer dans les différentes sous-bandes de la transformée en ondelettes des images. Nous décrivons ce procédé ci-dessous dans le domaine en clair avant de détailler comment nous l'avons sécurisé.

II CBIR à base de signature globale dans le domaine en clair

Dans ce travail, nous considérons l'algorithme de CBIR fondé sur la signature d'image globale proposée dans [7] qui résume une image aux histogrammes des différentes sous-bandes d'ondelettes de l'image. Le calcul de cette signature repose sur deux étapes :

— Calcul de la transformée en ondelettes de l'image jusqu'au $d^{ème}$ niveau de décomposition.

— Calcul et comparaison des histogrammes des sous-bandes d'ondelettes du niveau 0 au d ; niveau.

Une fois les signatures calculées, elle sont comparées sur la base de la mesure distance L^1. Les images qui minimisent cette distance seront considérées comme semblables.

II.1 Transformée en ondelettes

La transformée en ondelettes permet de décomposer un signal ou une image sur une base de fonctions, appelées ondelettes, qui sont des versions dilatées et translatées d'une fonction de base, appelée ondelette mère. Si l'on change l'ondelette mère, une décomposition différente est obtenue. La décomposition en ondelettes d'un signal ou d'une image peut être obtenue simplement à l'aide d'une analyse multi-résolution. Dans le cas d'un signal $1-D$ de longueur N, un banc de filtres est utilisé pour extraire deux signaux de longueur $\frac{N}{2}$ l'un contenant une approximation du signal, l'autre contenant les détails (voir Figure 2.3). Les coefficients des filtres dépendent de l'ondelette mère. Le banc de filtres est appliqué en cascade à l'approximation du signal jusqu'à obtenir une approximation très grossière et des détails à différentes échelles.

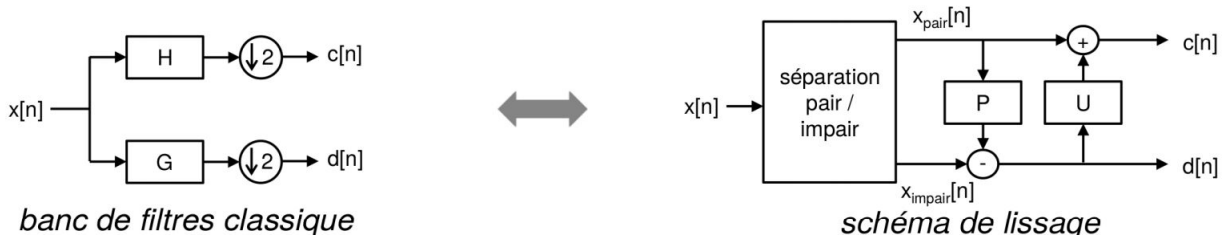


FIGURE 2.3 – Équivalence des bancs de filtres - cas des ondelettes séparables

L'adaptation dans le cas des images varie selon que l'ondelette mère est séparable ou non, c'est-à-dire selon qu'elle peut ou ne peut pas s'écrire comme le produit d'une ondelette $1-D$ horizontale et d'une ondelette $1-D$ verticale. Si l'ondelette mère est séparable, alors le traitement ci-dessus peut être appliqué séparément suivant les lignes et suivant les colonnes. Nous obtenons alors la décomposition en ondelettes de l'image, les coefficients de la projection étant regroupés par échelle et par direction (horizontale, verticale et diagonale). Si l'ondelette mère n'est pas séparable, alors l'espace des pixels de l'image est subdivisé en plusieurs (M) treillis complémentaires (voir Figure 2.3). Comme dans le cas $1-D$, un banc de filtres est appliqué à l'image : après filtrage, l'un des treillis contient l'approximation de l'image, les autres contiennent les coefficients de détail selon différentes directions ; le banc de filtre est appliqué en cascade à l'approximation de l'image.

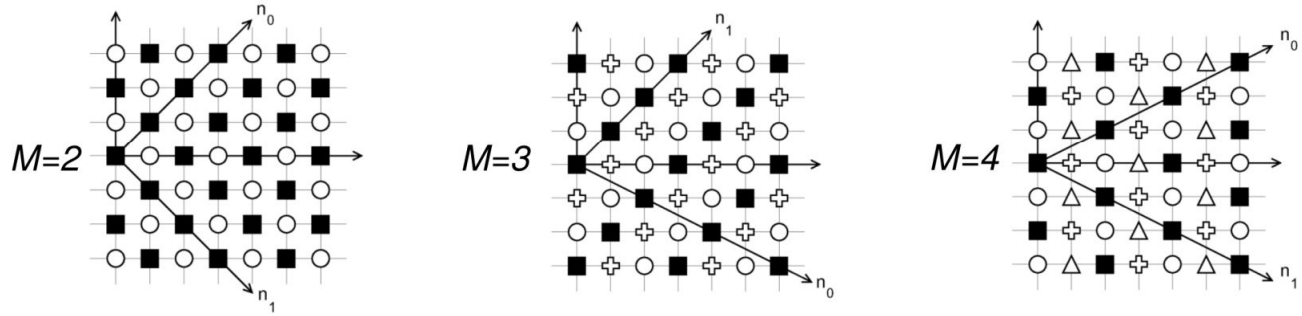


FIGURE 2.4 – Exemples de subdivisions en treillis - cas des ondelettes non séparables

Selon l'algorithme de Mallat [146], la transformée en ondelettes peut être défini de manière récursive :

$$a_j(k) = \sum_{l\in\mathbb{Z}} h(2k-l)a_{j-1}(l) \tag{2.1}$$

$$d_j(k) = \sum_{l\in\mathbb{Z}} g(2k-l)a_{j-1}(l) \tag{2.2}$$

avec $h(k)$ et $g(k)$ représentent les coefficients des filtres de décomposition passe-haut et passe-bas respectivement. $j = 1, 2, 3, ..., L$ où L est le niveau de la décomposition du signale d'entrée. $a_j(k)$ et $d_j(k)$ sont les coefficients d'approximation et de détails respectivement, on peut les calculer à partir de $a_{j-1}(k)$. $\{a_0(l)\}_{l\in\mathbb{Z}}$ est défini comme étant le signal d'entrée qui est désignée par $\{x(l)\}_{l\in\mathbb{Z}}$.

Dans le cas des images et ce qui concerne ce mémoire, une décomposition en ondelettes séparable est couramment utilisée, cela signifie que les opérations de filtrage unidimensionnelles décrites ci-dessus sont effectuées sur les lignes et les colonnes séparément (les mêmes filtres sont utilisés le long de deux directions). À chaque niveau de décomposition on obtient trois sous-bandes qui représentent les détails de l'image, et une unique sous-bande obtenu à la dernière décomposition et qui présente l'approximation de l'image. Si on considère que l'ondelette est séparable, alors la transformée en ondelettes $2D$ d'une image I est donné par les équations ci-dessous :

$$a^d(x,y) = \sum_{l\in\mathbb{Z}}\sum_{l'\in\mathbb{Z}} h(2x-l)h(2y-l')a^{d-1}(l,l') \tag{2.3}$$

$$b_u^d(x,y) = \sum_{l\in\mathbb{Z}}\sum_{l'\in\mathbb{Z}} w(2x-l)w'(2y-l')a^{d-1}(l,l') \tag{2.4}$$

Où $a^d(x,y)$ et $b_u^d(x,y)$ sont les coefficients d'approximation et de détail du $d^{ème}$ niveau de décomposition, respectivement ; (x,y) donne la position du coefficient dans la sous-bande ; $w, w' \in \{h,g\}$ où h et g représentent les coefficients des filtres de décomposition passe-haut et passe-bas respectivement ; $u \in \{hg, gh, gg\}$ est l'indice des sous-bandes de coefficient de détail (c'est-à-dire horizontal, vertical et diagonal). Dans le cas $d = 0$, $a^0(x,y)$ correspond au pixel d'image $I(x,y)$. Comme on peut le voir, le calcul de la transformée en ondelette s'appuie sur des opérations linéaires (additions et multiplications). Nous verrons en section III.4, qu'elles peuvent être sécurisées à l'aide du chiffrement homomorphe.

II.2 Calcul et comparaison d'histogrammes

Pour construire la signature d'une image, l'étape suivante consiste à calculer les histogrammes de chacune des sous-bandes d'ondelette de détails jusqu'à un niveau de résolution donné. La signature de l'image correspond à l'ensemble de ces histogrammes. La Figure 2.5 donne l'exemple de la signature d'une image de rétinopathie pour un niveau de décomposition..

Pour pouvoir sécuriser le calcul de la signature et leur comparaison, nous rappelons ci-après les principes de construction d'un histogramme dans le domaine en clair comme aussi comment comparer deux histogrammes sur la base de la distance L^1.

- **Calcul de l'histogramme dans le domaine en clair**

Soit $C_u^d(x,y)$ un coefficient d'ondelettes à la position (x,y) dans la sous-bande u, $u \in \{hh, gh, hg, gg\}$, au niveau de décomposition d. Pour construire l'histogramme $H_{C_u^d}$ de la sous-bande C_u^d, la dynamique des coefficients est d'abord subdivisée en K intervalles uniformes ou classes de taille Δ (voir la Figure 2.6). Par définition, la valeur $H_{C_u^d}(k)$ indique le nombre de coefficients $C_u^d(x,y)$ dont les valeurs appartiennent au $k^{ème}$ intervalle de $H_{C_u^d}$. Plus clairement, $H_{C_u^d}(k)$ donne la

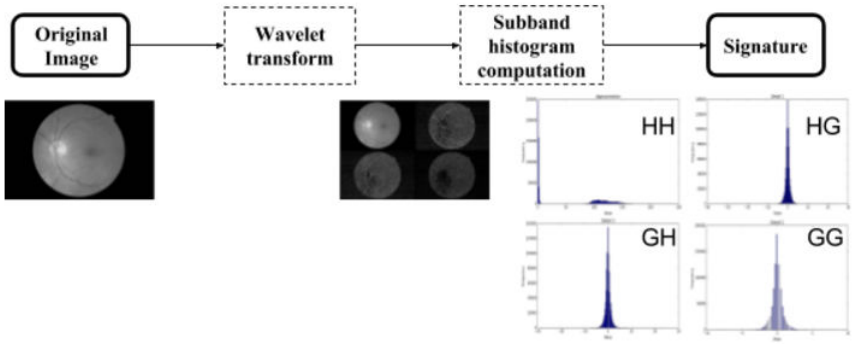


FIGURE 2.5 – La signature de l'image correspond à l'ensemble de ses histogrammes de sous-bandes d'ondelettes jusqu'à un niveau de décomposition donné. Dans cet exemple, un seul niveau de décomposition est considéré. hh, hg, gh et gg représentent la sous-bande d'approximation et les sous-bandes de détail horizontales, verticales et diagonales, respectivement

cardinalité de la classe C_k. Si T_k désigne le centre du $k^{ème}$ intervalle de $H_{C_u^d}$, alors la classe C_k de $C_u^d(x,y)$ est donnée par :

$$k = \arg\min_p |C_u^d(x,y) - T_p| \tag{2.5}$$

Une autre manière pour calculer la classe d'appartenance d'un coefficient est d'utiliser la quantification scalaire uniforme (QSU). Dans ce cas on fait l'hypothèse, comme précédemment que chaque classe est de même largeur Δ, qui est dans ce cas le pas de quantification de la QSU. La classe d'un coefficient est alors directement :

$$k = [\frac{C_u^d(x,y)}{\Delta}] \tag{2.6}$$

où [] est l'opérateur d'arrondi.

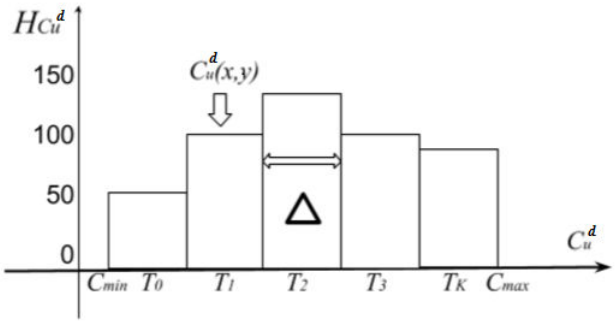


FIGURE 2.6 – L'histogramme $H_{C_u^d}$ correspond à la sous-bande C_u^d dans le cas où $K = 5$ et une dynamique de coefficients $C_u^d(x,y) \in [C_{min}, C_{max}]$. Les $\{T_u\}_{0 \leq u \leq 4}$ représentent les centres des classes de l'histogramme

- **Comparaison entre deux images**

La distance utilisée pour comparer deux images Im_1 et Im_2 est fondée sur la distance L^1 entre les histogrammes de leurs différentes sous-bandes de coefficients.

$$d(Im_1, Im_2) = \sum_{d=1}^{3L+1} (H_{C_u^{d(1)}} - H_{C_u^{d(2)}}) \tag{2.7}$$

$$H_{C_u^{d(1)}} - H_{C_u^{d(2)}} = \sum_{k=1}^{N_B} |H_{C_u^{d(1)}}(k) - H_{C_u^{d(2)}}(k)| \tag{2.8}$$

où $H_{C_u^{d(i)}}(k)$ est la cardinalité de la $k^{\text{ème}}$ classe de l' histogramme de la sous-bande u. L est le niveau de décomposition et N_B le nombre de classes chaque histogramme $H_{C_u^{d(i)}}$ où $i \in \{1, 2\}$.

Pour sécuriser ce système de CBIR, il convient de pouvoir calculer la transformée en ondelettes d'images chiffrées, et être capable de calculer les histogrammes des différentes sous-bandes et de pouvoir ensuite mesurer la distance L^1 entre eux.

III Système de recherche par le contenu sur des données chiffrées

L'objectif que nous nous sommes fixé est de trouver une solution pour implémenter le système de CBIR précédent sur des images chiffrées sous la contrainte d'aucune interaction entre le cloud et le client ou un autre tiers lors des calculs liés à l'extraction et la comparaison des signatures des images.

Les grandes étapes à sécuriser sont le calcul de la transformée en ondelettes de l'image, le calcul d'histogramme et la distance L^1 entre deux histogrammes. Nous avons vu que la transformée en ondelette s'appuie sur des opérations linéaires. Dans le domaine chiffré, elle peut être calculée à l'aide d'un cryptosystème qui possède simplement la propriété d'homomorphie additive (i.e. à partir de deux chiffrer $E[x]$ et $E[y]$ on peut calculer $E[x + y]$ où $E[.]$ la fonction de chiffrement). Le calcul d'histogramme et de la distance sont eux plus complexes. Comme nous l'avons vu la construction d'un histogramme peut se faire soit à l'aide d'une opération de quantification ou soit par la minimisation de la distance L^1 entre les coefficients d'ondelettes et des seuils de références. Ces deux opérations sont des opérations non linéaires. L'un utilise une division l'autre une valeur absolue. Des solutions sont à trouver.

Dans ces travaux nous avons fait le choix d'utiliser le cryptosystème de Paillier. Nous décrivons ci-après ce système et comment l'utiliser pour implémenter la transformée en ondelettes dans le domaine chiffré. Nous la décrirons après un état de l'art sur les solutions existantes de comparaison de données dans le domaine chiffré, nous présenterons une méthode originale qui permet de comparer des données chiffrées avec le cryptosystème de Paillier sans communications. Nous détaillons ensuite comment utiliser cette solution le calcul et la comparaison de signatures.

III.1 Cryptosystème de Paillier

Le cryptosystème de Paillier inventé en 1999, par le chercheur français Pascal Paillier est un algorithme de cryptographie à clé publique. Ce système fournit un homomorphisme additif. Comme évoqué dans le chapitre 1, chaque cryptosystème homomorphe est définie par quatre fonctions ($KeyGen()$, $Enc()$, $Dec(.)$ et $Eval(.)$). Les descriptions de ces fonctions pour le cryptosystème de Paillier sont les suivantes :

* $KeyGen()$: On choisit deux nombres premiers de grande taille, indépendants et aléatoires p et q, à partir desquels calculés le produit $K_p = pq$ et $K_s = PPCM(p-1, q-1)$ où $PPCM(.)$ est le Plus Petit Multiple Commun. On choisit un entier $g \in \mathbb{Z}^*_{K_p^2}$ tel que $PGCD(K_p, L(g^{K_s} \mod K_p^2)) = 1$ où : $PGCD(.)$ est le Plus grand commun diviseur ; la fonction L est définie comme $L(u) = \frac{u-1}{K_p}$; $\mathbb{Z}^*_{K_p^2}$ représente le sous groupe multiplicatif de $\mathbb{Z}_{K_p^2} = \{0, .., K_p^2 - 1\}$. La clé publique est (K_p, g) et la clé secrète est K_s.
* $Enc(m)$: Pour un message $m \in \mathbb{Z}_{K_p}$, on choisit aléatoirement un nombre $r \in \mathbb{Z}^*_{K_p}$ et le chiffrement de m est de la forme :

$$c = E[m, r] = g^m r^{K_p} \mod K_p^2 \tag{2.9}$$

  il est important de noter que l'entier r rend le cryptosystème de Paillier probabiliste ou sémantiquement sûr. Plus clairement, en fonction de la valeur de r, un message aura plusieurs chiffrés possibles sans modifier la clé de déchiffrement. Comme introduit dans [87], il est possible d'obtenir une version rapide de l'opération de chiffrement (2.9) en fixant $g = 1 + K_p$ sans réduire la sécurité de l'algorithme. Ce faisant, le chiffrement de m en c n'exige qu'une seule exponentiation modulaire et deux multiplications modulaires.

$$c = E[m, r] = (1 + mK_p) r^{K_p} \mod K_p^2 \tag{2.10}$$

  cette propriété sera importante pour comparer deux messages chiffrés par le cryptosystème de Paillier.
* $Dec(c)$: Pour un message chiffré $c \in \mathbb{Z}^*_{K_p^2}$, le déchiffrement est définie comme :

$$m = \frac{L(c^\lambda \mod K_p^2)}{L(g^\lambda \mod K_p^2)} \mod K_p \tag{2.11}$$

* Propriétés d'homomorphie ($Eval(c_1 = E[m_1, r_1], c_2 = E[m_2, r_2])$) :

$$E[m_1, r_1] \times E[m_2, r_2] = E[m_1 + m_2, r_1 r_2] \tag{2.12}$$

$$E[m_1, r_1]^{m_2} = E[m_1 m_2, r_1^{m_2}] \tag{2.13}$$

  où m_1 et m_2 sont deux messages en clair et c_1, c_2 leurs chiffrés associés respectifs. Les équations ci-dessus montrent que le cryptosystème de Paillier est homomorphiquement additif. Ces propriétés nous aideront à implémenter la transformée en ondelettes dans le domaine de Paillier et par la suite construire l'histogramme d'une image chiffrée via un algorithme de comparaison entre des messages chiffrés par le cryptosystème de Paillier.

III.2 Transformée en ondelettes dans le domaine chiffré

Afin d'implémenter la transformée en ondelettes dans le domaine chiffré de Paillier, nous nous somme appuyés sur les travaux de Zheng *et al* [147].

III.2.1 Codage des nombres réels

Puisque le domaine en clair et celui des chiffrés sont représentés par des entiers positifs dans le cryptosystème de Paillier, l'ensemble des données doivent être des entiers, cela vaut pour les données à traiter comme pour les opérateurs de filtrages.

Prétraitement sur les données en entrée

Si l'on suppose que les pixels de l'image sont des entiers $I(x,y) \in \mathbb{Z}$, se pose encore le problème de comment présenter les entiers négatifs du fait que la transformée en ondelettes implique des soustractions.

Zheng *et al.* [147] ont proposé de représenter les valeurs positives dans l'intervalle $[0, \frac{K_p-1}{2}]$ et les valeurs négatives dans l'intervalle $[\frac{K_p+1}{2}, K_p-1]$, où K_p est la clé publique utilisée comme module dans le cryptosystème de Paillier. Ils ont montré que cela est possible si K_p vérifie la condition suivante

$$K_p \geq 2 \sup_{(x,y)} |I(x,y)| + 1 = 2U + 1 \tag{2.14}$$

Comme K_p est un produit de deux grands nombres premiers, cette condition est toujours vérifiée.

Prétraitement sur les coefficients des filtres : utilisation d'un facteur d'expansion

De manière générale, les coefficients de filtrage h_d et g_d de la décomposition en ondelette (cf. section II.1) sont des nombres réels. Ils doivent aussi être convertis en entiers pour que l'on puisse les traiter dans le domaine chiffré de Paillier. Un procédé de conversion est le suivant :

$$H_d = [Qh_d] \tag{2.15}$$

$$G_d = [Qg_d] \tag{2.16}$$

où [.] est l'opérateur d'arrondi et $Q \in \mathbb{N}$ dénote le facteur d'expansion. Avec ce facteur d'expansion, la forme récursive de la transformée en ondelettes telle que nous l'avons vu auparavant en section II.1 devient alors

$$A^d(x,y) = \sum_{l\in\mathbb{Z}} \sum_{l'\in\mathbb{Z}} H(2x-l)H(2y-l')a^{d-1}(l,l') \tag{2.17}$$

$$B_u^d(x,y) = \sum_{l\in\mathbb{Z}} \sum_{l'\in\mathbb{Z}} W(2x-l)W'(2y-l')a^{d-1}(l,l') \tag{2.18}$$

Où $A^d(x,y)$ et $B_u^d(x,y)$ sont les coefficients d'approximation et de détails du $d^{ème}$ niveau de décomposition, respectivement ; $W, W' \in \{H, G\}$ où H et G sont les versions quantifiées des coefficients de filtrage de décomposition passe-bas et passe-haut h et g, respectivement ; $u \in \{HG, GH, GG\}$ indique l'indice des sous-bandes de coefficient de détail (c'est-à-dire horizontal, vertical et diagonal). Il faut noter que toutes les multiplications et les exponentiations sont effectuées modulo $\mathbb{Z}^*_{K_p^2}$.

Nous soulignons que l'utilisation de ce facteur d'expansion influera de facto sur la précision de la transformée en ondelettes. Il peut y avoir un écart avec la transformée calculée en clair. Cela dépend de l'ondelette choisie.

III.2.2 La transformée en ondelettes dans le domaine de Paillier

Grâce aux prétraitements précédents, la transformée en ondelettes s'exprime maintenant par des multiplications et des additions entre des nombres entiers. De ce fait, les équations (2.17) et (2.18) peuvent être calculées dans le domaine chiffré de Paillier en profitant de ses propriétés d'homomorphie. L'implémentation de la transformée en ondelettes dans le domaine chiffré est donnée par

$$E[A^d(x,y), r^d(x,y)] = \prod_{l,l'\in\mathbb{Z}} E[A^{d-1}(l,l'), r^{d-1}(l,l')]^{H(2x-l)H(2y-l')} \tag{2.19}$$

Niveau de décomposition	Borne B de la clé publique
1	$\lfloor\sqrt{2n}UQ + nU/2\rfloor$
j	$\lfloor\sqrt{n}U_A(\sqrt{2}Q)^j + (\frac{n}{2}U_A)\sum_{r=0}^{j}(nQ+\frac{n}{2})^{j-r-1}(\sqrt{2}Q)^r\rfloor$

TABLE 2.1 – Bornes que la clé publique doit respecter pour le calcul sans erreurs d'un ou plusieurs niveaux de décomposition d'ondelettes. Q représente le facteur d'expansion, n la taille de l'image d'entrée et U et U_A la valeur maximale des pixels de l'image en entrée et les coefficients au $j^{ème}$ niveau de décomposition, respectivement.

$$E[B_u^d(x,y), r_u^d(x,y)] = \prod_{l,l' \in \mathbb{Z}} E[A^{d-1}(l,l'), r^{d-1}(l,l')]^{W(2x-l)W'(2y-l')} \tag{2.20}$$

où $r^d(x,y)$ et $r_u^d(x,y)$ sont les aléas associés aux coefficients d'approximation et de détail du $d^{ème}$ niveau de décomposition, respectivement.

Il n'y aura pas d'écart entre la transformée en ondelette et sa version chiffrée si toutes les valeurs $A^d(x,y)$ (positifs ou négatifs) sont bien présentées dans $\mathbb{Z}_{K_p}$. De la même manière que les nombres réels, cela est vérifié si la condition suivante est satisfaite :

$$K_p \geq 2sup|A^d(x,y)| + 1 = 2U_A + 1 \tag{2.21}$$

Si cette condition est vérifiée alors la valeur de $D[E[A^d(x,y)]]$ sera calculée sans erreur, c'est à dire, $D[E[A^d(x,y)]] = A^d(x,y)$. Pour ce faire, il faut trouver une borne B pour majorer U_A et choisir un module qui soit supérieure à $2B+1$. Zheng *et al* [147]. ont donné un tableau de valeur de cette borne pour un seul niveau de décomposition et j niveaux de décomposition (voir table 2.1). En général, si on choisit un module K_p codé sur 1024 bits alors la condition (2.21) est toujours vérifiée.

où Q présente le facteur d'expansion, n la taille de l'image d'entrée et U la valeur maximal des pixels de l'image d'entrée. En général, quand on choisit un module K_p codé sur 1024 bits la condition (2.21) est toujours vérifiée.

III.3 Calcul et comparaison d'histogramme dans le domaine chiffré

Dans la partie précédente, nous avons montré comment on peut implémenter la transformée en ondelettes dans le domaine chiffré de Paillier. Dans ce qui suit nous allons expliquer comment on peut calculer et comparer les signatures d'images chiffrées ; signatures qui sont les histogrammes des différentes sous-bandes de l'image. Avant de rentrer dans les détails pour expliquer le calcul d'un histogramme d'une image chiffrée, nous allons montrer comment on peut comparer des données chiffrées avec le cryptosystème de Paillier. C'est une fonction essentielle au calcul d'histogramme.

III.3.1 Comparaison dans le domaine chiffré

Il existe deux classes de méthode de comparaison de données dans le domaine chiffré. La première classe, qui est aussi celle étudiée depuis longtemps, regroupe les méthodes qui sont basées sur des protocoles, un jeu d'interactions, entre deux entités. De nombreux protocoles sécurisés sont connus pour la comparaison de deux nombres entiers, aussi nommé le « problème des millionnaires » qui veulent savoir qui est le plus riche sans qu'aucun d'eux ne dévoile à l'autre le montant de sa richesse. Le premier protocole a été proposé par Yao en 1982 [119]. Il correspond au « Garbled Circuit » qui a été amélioré plusieurs fois depuis. L'une des implémentations les plus efficaces basées sur le chiffrement homomorphe est décrite par T. Veuguen [148]. Mais, celle-ci,

comme toutes les autres, nécessite de nombreuses communications entre les entités participantes afin d'effectuer une comparaison.

La deuxième classe est sans communications et a été récemment présentée par Hsu *et al.* dans [1]. La solution que nous proposons est une amélioration de cette approche, elle permet de comparer deux données chiffrées ou non avec la même clé publique. Notre solution prend en compte la version rapide du cryptosystème de Paillier proposée par Damgärd [87].

Avant de présenter notre solution, revenons sur la méthode de Hsu *et al.* [1]. Considérons un scénario client-serveur où le serveur possède deux messages chiffrées $c_1 = E[m_1, r_1]$ et $c_2 = E[m_2, r_2]$ par un client ; m_1 et m_2 sont deux entiers que le serveur veut comparer en sachant que c_1 et c_2. L'idée de base de Hsu *et al.* n'est pas de comparer directement les données mais d'établir leur relation (plus petite, identique ou grande) par la quantification des données à partir de leurs versions chiffrées. Dans un premier temps, le client envoie avec c_1 et c_2 un ensemble de seuils $\{T_i\}_{1 \leqslant i \leqslant N}$ qui quantifient la dynamique des deux valeurs entières m_1 et m_2. En pratique, le client choisit une suite croissante de seuils répartis de manière aléatoire $\{T_i \in \mathbb{Z}_{K_p}\}_{1 \leqslant i \leqslant N}$ et envoie $\{E[T_i, r_1]\}_{1 \leqslant i \leqslant N}$ et $\{E[T_i, r_2]\}_{1 \leqslant i \leqslant N}$ au serveur. Pour comparer les données, le serveur doit d'abord identifier l'intervalle auquel appartient chaque message m_u, $u \in \{1, 2\}$. Dans ce cas, on considéra que m_u appartient au $k^{ième}$ intervalle, si le seuil le plus proche de m_u est T_{k_u}. Pour identifier k_u, le serveur démarre le processus itératif suivant :

1. **Comparaison de $E[m_u, r_u]$ avec un seuil $E[T_i, r_u]$.** Sur la base de la propriété du cryptosystème de Paillier $E[m_u, r_u].E[a, b] = E[m_u + a, r_u.b]$, il est possible de calculer la distance $D_{iff}(T_i, m_u) = T_i - m_u$ en multipliant itérativement $E[m_u, r_u]$ avec $E[1, 1]$ jusqu'à ce que le produit soit égal à $E[T_i, r_u]$. Le nombre d'itérations inc donne $D_{iff}(T_i, m_u)$ (c'est-à-dire $E[T_i, r_u] = E[m_u + inc, r_u]$; $D_{iff}(m_u, T_i) = inc$)

2. **Seuil le plus proche à m_u.** Le seuil le plus proche T_{k_u} est naturellement donné par la distance minimale entre $E[m_u, r_u]$ et tous les seuils chiffrés $\{E[T_i, r_u]\}_{1 \leqslant i \leqslant N}$. Du fait que $E[m_u, r_u]g^{inc} = E[m_u + inc, r_u]$, tout ce processus peut être résumé pour tout message m_u, $u \in \{1, 2\}$, par la relation

$$(d_u, k_u) = arg \min_{i,inc}(E[T_i, r_u] - E[m_u, r_u]g^{inc} \mod K_p^2) \tag{2.22}$$

où d_u est la distance entre le message m_u et son plus proche seuil d'indice k_u.

Après avoir calculé (d_1, k_1) et (d_2, k_2) pour m_1 et m_2, il est possible de déterminer si $m_1 \leqslant m_2$ ou non, sans les déchiffrer. Cela est lié au fait que les mêmes seuils $\{T_i\}_{1 \leqslant i \leqslant N}$ sont utilisés pour m_1 et m_2. Il est important de souligner que la distance relative entre m_1 et m_2 ne donne aucune idée sur leurs valeurs respectives. Dans le même temps, la complexité de cette approche est plus ou moins importante. Elle dépend du nombre de seuils et de la dynamique ou de la plage de valeurs des données à comparer.

Par rapport à [1], notre solution n'a pas besoin d'une procédure itérative et d'envoyer plusieurs seuils. Au contraire, le client devra simplement envoyer au serveur deux versions chiffrées d'un seul seuil T, c'est-à-dire $E[T, r_1]$ et $E[T, r_2]$. T sera utilisé comme "valeur de référence" dans la dynamique de m_1 et m_2 pour la comparaison. Pour calculer directement la distance entre m_u et T à partir de $E[m_u, r_u]$ et $E[T, r_u]$, nous profitons de la version proposé par Damagärd du cryptosystème de Paillier, c'est-à-dire en prenant on compte que $g = 1 + K_p$ (cf. section III.1),

comme suit

$$\begin{aligned} d_u = D_{iff}(T, m_u) &= D^e_{iff}(E[m_u, r_u], E[T_u, r_u]) \\ &= \frac{E[T, r_u]E[m, r_u]^{-1} - 1 \mod K_p^2}{K_p} \mod K_p \\ &= \frac{g^T r_u g^{-mu} r_u^{-1} \mod K_p^2}{K_p} \mod K_p \\ &= \frac{g^{T-m_u} - 1 \mod K_p^2}{K_p} \mod K_p \\ &= T - m_u \mod K_p \end{aligned} \tag{2.23}$$

Où D_{iff} et D^e_{iff} sont deux fonctions qui permettent de calculer la distance L^1 dans les domaines en clair et chiffré, respectivement. Comme précédemment, une fois que d_1 et d_2 sont calculés, on peut déterminer si $m_1 \leq m_2$ sans aucune communication avec l'utilisateur ou un autre tiers. À nouveau la connaissance de la distance relative entre m_1 et m_2 ne donne aucun indice sur les valeurs de m_1 et m_2. Par rapport à la solution de Hsu *et al.* [1], dont la complexité de calcul est $O(\frac{K_p}{N})$ (pour N seuils, [1] ont besoin de $\frac{K_p}{N}$ itérations pour trouver le seuil le plus proche dans le pire des cas), notre proposition est de complexité constante $O(1)$ (un seuil sans aucune itération).

— Comparaison des données chiffrées avec deux clés publiques différentes

Il convient de noter que les deux solutions ci-dessus exigent que m_1 et m_2 soient chiffrées avec la même clé publique K_p et la valeur aléatoire r_u. Pour surmonter ce problème et pouvoir comparer deux messages m_1 et m_2 chiffrés avec deux clés publiques différentes K_{p1} et K_{p2} et deux valeurs aléatoires r_1 et r_2 définies respectivement par deux utilisateurs U_1 et U_2 (e.g. $c_1 = E[m_1, r_1]$ et $c_2 = E[m_2, r_2]$), nous proposons d'exploiter une valeur de référence P de la manière suivante. On demande à U_1 et U_2 de chiffrer P avec les mêmes paramètres que pour m_1 et m_2, et d'envoyer les résultats au serveur, c'est-à-dire $E[P, r_1]$ et $E[P, r_2]$. Le serveur peut calculer les distances relatives $d_1 = D(P, m_1) = P - m_1$ et $d_2 = D(P, m_2) = P - m_2$, résultats intermédiaires à partir desquels le serveur peut dériver la quantité $m_1 - m_2$ et donc déduire si m_1 est égal ou supérieur à m_2.

III.3.2 Calcul d'histogramme dans le domaine chiffré

Comme discuté en section II.2, la construction d'un histogramme consiste à déterminer la classe auquel appartient un coefficient d'ondelette. Le nombre de coefficients appartenant à la même classe après le parcours d'une sous-bande d'ondelette donne la cardinalité de la classe.

Une première stratégie consiste à ce que le client envoie pour chaque coefficient les centres des classes de l'histogramme chiffrés avec les mêmes aléas que le coefficient. La distance avec chaque seuil peut être calculée sur la base des solutions précédentes et celle qui minimise la distance avec le coefficient indiquera la classe d'appartenance du coefficient. Cette solution présente plusieurs problèmes. Dans un premier temps, la quantité d'information à transmettre est importante. Dans un second, elle peut amener au calcul de différences négatives ; valeurs que Paillier ne peut pas manipuler. En fait, le calcul de différence proposé précédemment fournie la valeur absolue de la différence. Nous proposons une solution qui règle ces deux problèmes simultanément. Ainsi, plutôt que d'envoyer K seuils chiffrés, le client envoie au serveur une valeur référence T chiffrées. Pour garantir que les différences entre les coefficients d'ondelette et cette valeur soient toujours positives, nous prenons pour valeur de T une valeur supérieure à la valeur maximum de la

dynamique des coefficients. Cette contrainte respectée, la classe d'appartenance d'un coefficient peut être déterminée en quantifiant les différences calculées. Plus clairement, soit $d_{x,y}$ la distance entre T et un coefficient d'ondelette $C_u^d(x,y)$:

$$d_{x,y} = T - C_u^d(x,y) \tag{2.24}$$

La classe de $C_u^d(x,y)$ est alors $C_k = [\frac{d_{x,y}}{\Delta}]$, où Δ est le pas de quantification de l'histogramme et [] est l'opérateur d'arrondi. C'est cette stratégie que nous avons sécurisée.

- Calcul de l'histogramme d'une sous-bande d'ondelette chiffrée

La construction de l'histogramme dans le domaine chiffré impose donc de déterminer la classe C_k à laquelle appartient un coefficient d'ondelettes $C_u^d(x,y)$ à partir de sa version chiffrée $C_u^{de}(x,y) = E[C_u^d(x,y), r_u^d(x,y)$ où $r_u^d(x,y)$ est un entier choisit de manière aléatoire dans $\mathbb{Z}_{K_p}^*$. En d'autres termes, nous devons calculer (2.24) dans le domaine chiffré.

Pour ce faire, et comme évoqué ci-dessus, l'utilisateur envoie avec l'image chiffrée, un seuil chiffré par les mêmes aléas que chaque coefficients d'ondelettes, c'est-à-dire $E[T, r_u^d(x,y)]$. Sur la base de cette information et en utilisant la solution de comparaison de données proposée en section III.2, il est possible de déterminer l'intervalle ou la classe d'un coefficient $C_u^d(x,y)$ à partir de $C_u^{de}(x,y)$ comme suit

$$d_{x,y} = D_{iff}^e(C_u^{de}(x,y), E[T, r_u^d(x,y)]) \tag{2.25}$$

Il est important de souligner que pour que le serveur calcule l'équation (2.25), l'utilisateur doit chiffrer le seuil T avec la valeur aléatoire appropriée $r_u^d(x,y)$, c'est-à-dire, la même valeur aléatoire que pour le coefficient chiffré $C_u^{de}(x,y)$. Afin que cette solution fonctionne, il faut donc le client soit capable de calculer *a priori* les aléas associés à un coefficient d'ondelette. Cela est possible du fait de l'existence d'une relation récursive entre les valeurs aléatoire dans la décomposition en ondelette indépendamment du contenu de l'image, du niveau 0 au niveau d. Cette relation profite des propriétés du cryptosystème de Paillier et de la linéraité des ondelettes. Cette relation entres aléas, $r_u^d(x,y)$, $u \in \{HH, GH_d, HG_d, GG_d\}$ est la suivante

$$r_u^d(x,y) = \prod_{l,l' \in \mathbb{Z}} r_u^{d-1}(l,l')^{W(2x-l)W'(2y-l')} \tag{2.26}$$

comme on peut remarquer, $r_u^d(x,y)$ est indépendant du contenu de l'image. Pour $d = 0$, $r_u^0(x,y) = r(x,y)$, c'est-à-dire, les aléas utilisés pour chiffrer les pixels de l'image I.

III.3.3 Comparaison entre histogrammes

À partir des équations (2.25) et (2.26), le serveur peut construire l'histogramme d'une sous-bande d'ondelettes ($H_{C_u^d}$). Afin de comparer deux histogrammes, $H_{C_u^{d(1)}}$ et $H_{C_u^{d(2)}}$ de deux images $I^{(1)}$ et $I^{(2)}$, respectivement, il suffit d'appliquer la distance L^1 entre les deux histogrammes.

$$L^1(H_{C_u^{d(1)}}, H_{C_u^{d(2)}}) = \sum_{1 \leqslant k \leqslant K} |H_{C_u^{d(1)}}(k) - H_{C_u^{d(2)}}(k)| \tag{2.27}$$

Soulignons le fait que ce résultat est calculer dans le clair. En fait l'histogramme de chaque sous-bande n'est pas chiffré et comme nous le verrons dans le Chapitre 3, cela peut permettre au cloud de découvrir en partie le contenu de l'image.

III.4 Le système de recherche par le contenu sécurisé

Dans notre schéma, les utilisateurs ont alimenté la base de données du serveur au cours du temps avec des images médicales et leur diagnostics associés. Ces images ont été envoyées chiffrées à l'aide du cryptosystème de Paillier paramétré avec la clé publique de l'utilisateur, accompagnée des chiffrés de la valeur de référence T pour chaque coefficient d'ondelettes (i.e. chiffré avec le même aléa).

Pour effectuer une recherche, l'utilisateur doit envoyer une image requête ainsi que les chiffrés d'une valeur de référence à l'aide de sa clé publique. Le serveur calcule la signature de l'image requête et des images qu'il possède déjà. Pour ce faire, il applique la transformée en ondelettes sécurisé sur l'image chiffrée afin d'accéder aux versions chiffrées des sous-bandes d'ondelettes d'image et construire ensuite leurs histogrammes.

La distance L^1 est calculée entre les histogrammes pour déterminer les images les plus semblables à l'image requête qui seront renvoyées au médecin avec leurs diagnostics associés. Le principe de notre solution est que l'utilisateur doit simplement chiffrer son image et le serveur effectue l'ensemble du processus de CBIR de manière sécurisée.

Si l'ensemble des calculs de signatures et de comparaison sont réalisés par le cloud, il n'y pas de communications autres que celles nécessaires l'externalisation des données, le client doit lui chiffrer un volume de données. En résumé pour une image de taille 92×112 pixels et une clé publique K_p codé sur 1024 bits, le client doit envoyer 5152 Kilobyte (kB) (l'image requête chiffrée et la valeur de référence T chiffré par les mêmes aléas des coefficients d'ondelettes)

IV Analyse de sécurité et complexité de notre système SCBIR

IV.1 Analyse de sécurité

Notre système CBIR sécurisé externalisé permet d'effectuer complètement une recherche dans une base de données d'images chiffrée externalisée dans le Cloud. Dans ce qui suit, nous discutons de la sécurité de ce système en termes de confidentialité des données dans le cas d'un cloud semi-honnête. Celui-ci stocke honnêtement les données chiffrées par les utilisateurs et répond à leurs demandes, mais il est curieux et peut essayer d'inférer des informations sur le contenu des données de l'utilisateur. Il peut essayer d'apprendre : la clé privée de l'utilisateur ou bien la version en clair des images de sa base de données ou bien la requête d'un utilisateur.

Avant d'entrer dans les détails de cette analyse, rappelons les étapes principales de notre système ainsi que toutes les données auxquelles le cloud a accès. Comme décrit précédemment, en plus de la clé publique de l'utilisateur, le cloud connait et accède :

— Aux images chiffrées de sa base de données ainsi que l'image requête chiffrée.

— Pour chaque coefficient d'ondelettes d'une image : la valeur de référence chiffrée par les mêmes aléas que les coefficients d'ondelette. Ils sont nécessaires pour trouver la classe d'appartenance des coefficients et le calcul des histogrammes $H_{C_u^d}$.

La confidentialité des images et de leurs transformée en ondelettes dépendent entièrement de la sécurité du cryptosystème de Paillier. L'analyse de sécurité de ce cryptosystème a été étudiée dans [86] en considérant différents modèles d'attaques : Attaque à texte chiffré seulement, Attaque à texte clair connu, Attaque à texte clair choisi et Attaque à texte chiffré choisi. Ces attaques correspondent à différents scénarios, en considérant l'accès à une certaine connaissance préalable qui pourrait aider le cloud à découvrir la clé privée de l'utilisateur ou à déchiffrer les données sans cette clé. Dans le contexte du cloud semi-honnête, notre système est concerné par « l'attaque à texte chiffré seulement » (en anglais ciphertext-only attack "COA") car le cloud

n'a accès qu'aux données chiffrées. Il a été démontré que si le cryptosystème Paillier n'est pas sécurisé contre l'attaque à texte chiffré choisi (chosen-ciphertext attack), comme tous les cryptosystèmes homomorphes en fait, il est sécurisé contre les trois autres attaques, y compris le modèle COA. En outre, sous tous ces modèles d'attaques, la clé privée de l'utilisateur (K_s) est sécurisée. Elle ne peut pas être récupérée [86].

En ce qui concerne la confidentialité des histogrammes, le cloud ne doit pas pouvoir définir les valeurs en clair des coefficients d'ondelette ($C_u^d(x,y)$) ou de la valeur de référence à partir du calcul des différences dans le domaine chiffré. Hsu *et al.* ont montré dans [1] que le calcul de la différence entre deux messages chiffrées avec la même valeur aléatoire est sécurisée sous l'attaque à texte chiffré seulement. Mais si le cloud à une idée sur la dynamique des coefficients d'ondelettes utilisée alors il peut reconstruire une version quantifiée de l'image originale. Nous verrons plus en détails cette attaque dans le chapitre suivant.

IV.2 Complexité du système CBIR externalisé sécurisé

Dans l'approche proposée, pour que le cloud cherche les images les plus proches à une image requête, l'utilisateur chiffre l'image et la valeur de référence avec les mêmes aléas que les coefficients d'ondelettes de l'image chiffrée et les envoie au serveur. La complexité de calcul associée à l'utilisateur est bornée par $O(m \times n)$ chiffrements où $m \times n$ est la taille de l'image.

Cette complexité reste correcte malgré l'utilisation d'un cryptosystème homomorphe. À noter également que les données ne sont envoyées qu'une fois et que toutes ces informations peuvent être utilisées à d'autres fins que la CBIR comme par exemple le calcul d'une signature avec une autre famille d'ondelette.

Du côté du serveur, la complexité de calcul dépend du calcul de la transformée en ondelettes de l'image dans le domaine chiffré, la construction des différents histogrammes et le calcul de la distance L^1 entre les signatures. La complexité de calcul de l'histogramme correspond en fait au nombre de comparaisons que le serveur doit faire et qui est borné par $O(m \times n)$ multiplications modulaires. Pour calculer la distance L^1 entre les signatures, il faut $O(L \times K)$ soustractions dans le domaine en clair, où L est la taille de la base de données et K est le nombre de classe dans les histogrammes. Ces soustractions sont négligeables par rapport aux opérations dans le domaine chiffré. En conséquence, la complexité de calcul du serveur est bornée par $O(m \times n)$ chiffrements par ce que la complexité d'une multiplication modulaire est inférieure à celle du chiffrement.

Concernant la complexité de communication de l'approche proposée, elle est bornée par $O(m \times n \times \log_2(K_p))$ bits où K_p est la clé publique du cryptosystème de Paillier. Le nombre d'interactions entre l'utilisateur et le serveur pour calculer et comparer une signature est nul, c'est là la grande différence avec les autres solutions fondées sur le calcul multipartite sécurisé [135,139]. Par ailleurs, sous l'hypothèse où le coût d'une communication équivaudrait celui d'une opération de chiffrement, notre solution est plus intéressante.

V Résultats expérimentaux

Dans cette section, nous présentons les résultats de tests que nous avons conduits sur deux bases de données : une médicale et l'autre biométrique. Nous présentons ces bases et les critères de performances considérés avant de comparer notre solution de SCBIR sécurisée à sa version en clair.

(a) (b)

FIGURE 2.7 – Exemples illustratifs de nos jeux de tests d'images (a) images de rétine, (b) image faciale d'un utilisateur.

V.1 Données expérimentales

Notre schéma SCBIR a été expérimenté dans le cadre de deux applications : l'aide au diagnostic et l'authentification biométrique de personne par reconnaissance de visage. Dans les deux cas, les données sont stockées et chiffrées dans le cloud.

La première base de données est constituée de 400 images de visage : dix photographies de 40 sujets distincts. Pour certains d'entre eux, les images ont été prises avec différentes éclairages et expressions de visage (yeux ouverts fermés, souriant pas souriant) et les détails du visage (lunettes et pas de lunettes). Toutes les images ont été prises sur un fond sombre homogène avec les sujets dans une position à peu près frontal. Les images ont une définition de 92×112 pixels. La base de données peut être consultée sur le site `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`. Cette base de données peut être classée facilement. Les images appartiennent à la même classe si et seulement si elles représentent le même sujet.

La deuxième base de données est constituée de 1200 images numériques de rétinopathie divisées en 3 groupes. Chaque groupe est formé de 4 sous-ensembles qui contiennent 100 images et un fichier Excel avec des diagnostics médicaux pour chaque image. Les images sont capturées en utilisant 8 bits par plan de couleur à 1440×960, 2240×1488 ou 2304×1536 pixels. Cette base de données peut être consultée sur le site `http://messidor.crihan.fr/download-en.php`.

Nous donnons en Figure 2.7 des exemples issus de nos deux bases de test.

V.2 Critère de performance

La performance de notre schéma, qui reste un système de recherche d'images par le contenu, est évaluée en termes de « précision moyenne » qui correspond au nombre d'images retournées par le système avec la même pathologie ou la même personne que dans l'image requête sur le nombre d'images retournées par le système.

$$P = \frac{A}{B} \times 100 \tag{2.28}$$

où P est la précision moyenne, A est le nombre d'images retournées par le système avec la même pathologie ou la même personne que dans l'image requête et B le nombre d'images retournées par le système.

Comme nous le verrons, cette mesure variera en fonction du nombre de classes considérées dans les histogrammes.

V.3 Résultats obtenus

Pour ces tests, nous avons utilisé la transformée en ondelettes $2D$ de Haar. Afin de rendre son calcul possible dans le domaine chiffré jusqu'au deuxième niveau de résolution (c.-à-d $d = 0, 1, 2$),

les coefficients des filtres de décomposition ont été transformés en valeurs entières en fixant le facteur d'expansion Q de (2.15) à 4. La signature d'une image correspond aux histogrammes de sous-bandes jusqu'au $2^{ème}$ niveau de résolution. L'histogramme de la sous-bande d'approximation est aussi pris en compte.

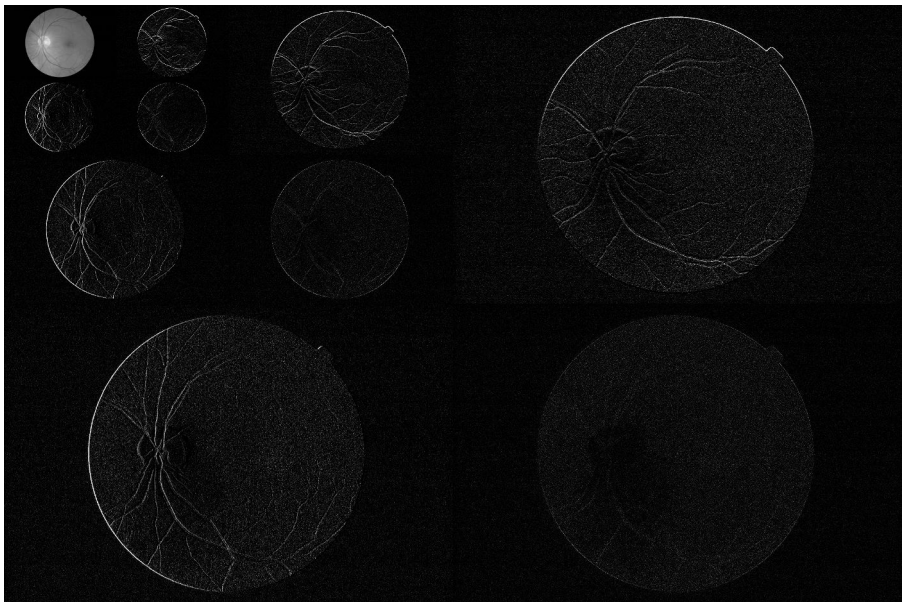


FIGURE 2.8 – Exemple d'une décomposition en 3 niveau de l'image (a) en Figure 2.7 en utilisant les ondelettes de Haar

Avant de détailler les performances de recherche de notre système, revenons sur la quantité de données que l'utilisateur doit envoyer dans ce contexte. Si les images sont chiffrées avec le cryptosystème de Paillier paramétré à l'aide de grands nombres premiers p et q de telle sorte que leur produit K_p soit codé sur 1024 bits, le chiffré d'un entier sera codé en conséquence codé sur 2048 bits. Pour une image de visage de 92×112 pixels, 5152 Kilobyte (kB) sur le cloud sont nécessaires pour stocker l'image et les valeurs de référence chiffrées.

V.3.1 Performance du CBIR sécurisé avec des images médicales

La Figure 2.9 fournit la performance de recherche de notre schéma pour une précision moyenne de cinq, c'est-à-dire que, lorsque le serveur renvoie cinq images les plus similaires à une image requête. Nous comparons également notre système avec la même approche CBIR dans le domaine clair en considérant les mêmes niveaux de décomposition (i.e. $d = \{0, 1, 2\}$) et plusieurs valeurs du pas de quantification pour le calcul de l'histogramme (i.e. $\Delta \in \{1, 2, 4, 8, 16, 32, 64\}$). Il est important de souligner que toutes les courbes sont tracées en moyenne, en utilisant 200 images de notre jeu test d'images de rétine comme images requête ; les 800 autres images constituant la base de données du serveur. Comme on peut le voir, notre système SOCBIR a la même performance que le système CBIR dans le domaine en clair. Un tel résultat peut être expliqué du fait que les coefficients de filtre de la transformée d'ondelette de Haar dans les domaines chiffré et clair sont équivalents. Plus clairement, l'expansion du coefficient n'a pas d'impact sur la précision de la transformée en ondelettes. Cela peut ne pas être le cas avec des transformations d'ondelettes dont les coefficients de filtre sont des nombres réels. En effet, une perte de précision dans le calcul du coefficient d'ondelettes pourrait réduire les performances de recherche. Au-delà,

h

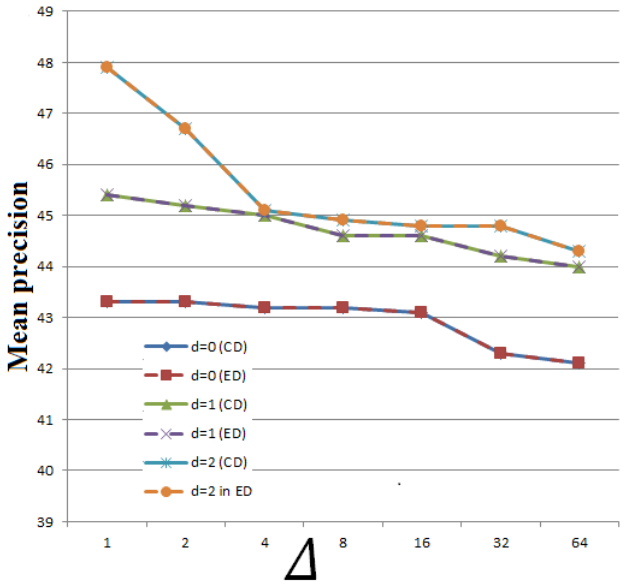


FIGURE 2.9 – Performance de recherche exprimée en précision moyenne à cinq (en %) pour notre base de données médicales en considérant différentes valeurs de pas de quantification (Δ) et différents niveaux de décomposition d. Les courbes pointillées désignent les résultats de SCBIR dans le domaine chiffré (ED "Encrypted Domain"), tandis que les courbes pleines donnent la performance du système CBIR équivalent dans le domaine en clair (CD "Clear Domain").

on pourra remarquer que travailler avec des pas de quantifications Δ de valeurs entre 1 et 16 n'affecte pas les performances de manière importantes. Rappelons que plus Δ est petit plus la signature est plus précise.

V.3.2 Performance du CBIR sécurisé avec des images biométriques

Cette expérience a été réalisée avec le même paramétrage que la précédente. À nouveau, notre base de données d'images test est divisée en deux ensembles : 200 images sont utilisées comme images requêtes tandis que les 200 autres sont utilisées comme images de référence par le serveur. La Figure 2.9 présente la précision moyenne en fonction de différentes valeurs de pas de quantification (i.e. $\Delta \in \{1, 2, 4, 8, 16, 32, 64\}$) et les niveaux de décomposition (i.e. $d = \{0, 1, 2\}$). Comme on le peut voir, la moyenne de précision est équivalente dans les domaines en clair et chiffré. La réponse de notre schéma à une image requête est également illustrée dans la Figure 2.10. Encore une fois, on peut voir que notre approche SCBIR atteint les mêmes performances que l'approche CBIR dans le domaine clair quelle que soit le pas de quantification et le niveau de décomposition. Sur la base du fait que les images faciales sont beaucoup plus petites que les images de rétine, la quantité d'informations que l'utilisateur doit envoyer au serveur est réduite.

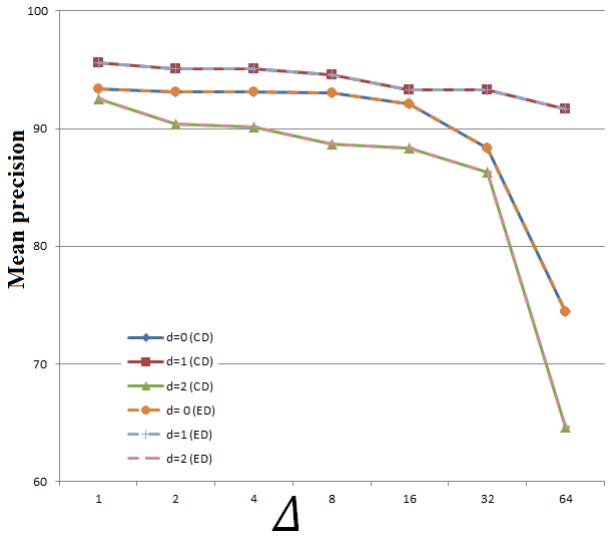


FIGURE 2.10 – La performance de recherche exprimée en précision moyenne à cinq (en %) pour notre base de données d'images biométrique en considérant différentes valeurs de pas de quantification (Δ) et niveaux de décomposition d. Les courbes en traits pointillés désignent les résultats de SCBIR dans le domaine chiffré (ED "Encrypted Domain"), tandis que les courbes pleines donnent la performance du système CBIR équivalent dans le domaine clair (CD "Clear Domain").

VI Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode de recherche par le contenu externalisé sécurisé (SCBIR), qui permet d'effectuer complètement une recherche dans une base de données d'images chiffrées maintenue par un serveur cloud, par exemple. Son originalité repose sur plusieurs points. Contrairement aux schémas de CBIR existant fondés sur le chiffrement homomorphe, il permet l'extraction d'une signature d'image globale et ne nécessite pas d'interactions entre l'utilisateur et le serveur ou un tiers de confiance pour réaliser la recherche d'images. Il exploite un algorithme de comparaison rapide entre des données chiffrées par le cryptosystème de Paillier, que nous proposons. Les résultats expérimentaux montrent que notre système SCBIR réalise les mêmes performances que l'approche CBIR équivalente dans le domaine en clair.

Le point faible de cette solution est que le serveur accède à la fois à la signature d'une image (i.e. les histogrammes des différentes sous-bandes) est en clair et qu'il connait la classe d'appartenance des coefficients d'ondelette chiffrées. Par conséquent, il est capable de reconstruire une version approchée de l'image à condition de connaitre en plus la dynamique des coefficients de l'image. Dans le chapitre suivant, nous rentrons dans le détail de ce type d'attaque et nous proposerons deux approches qui permettent de la contrecarrer. Celles-ci extraient d'une image chiffrée, une signature elle aussi chiffrée.

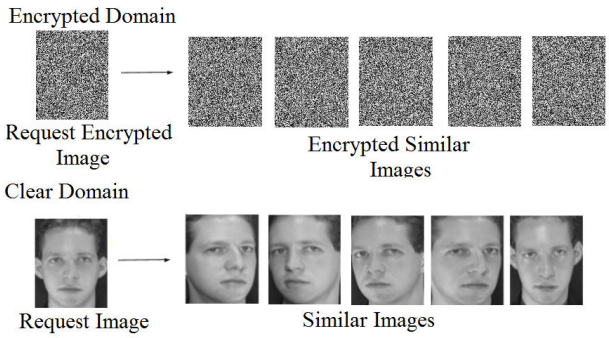


FIGURE 2.11 – Exemple illustrant la réponse de notre système dans le cadre de l'authentification d'utilisateurs par reconnaissance de visages.

Chapitre 3

Système de recherche par le contenu externalisé complètement sécurisé

Nous avons vu dans le chapitre précédent une solution sécurisée d'un système de recherche par le contenu. Cette dernière permet d'extraire une signature globale non-chiffrée à partir d'une image chiffrée sans communications supplémentaires entre l'utilisateur et le serveur. Cela est possible grâce à une méthode de comparaison entre données chiffrées de complexité très faible par rapport aux solutions existantes (Cf. Chapitre 2). Bien que de complexité réduite, cette solution extrait cependant une signature en clair qui peut mettre en danger la confidentialité des données d'utilisateurs sous quelques hypothèses.

Dans ce chapitre nous présentons deux solutions à ce problème. L'objectif est d'extraire une signature sécurisée, i.e. chiffrée, d'une image tout en évitant, comme dans le chapitre précédent, les communications entre l'utilisateur et le serveur. La première approche est fondée sur l'utilisation de deux fournisseurs de cloud indépendants. Cela permet de calculer une signature chiffrée par les deux fournisseurs de cloud sans l'intervention de l'utilisateur. La deuxième approche permet d'extraire une signature chiffrée d'une image chiffrée avec un seul serveur. À cette fin, nous combinons notre méthode de comparaison de données chiffrées avec un algorithme de sélection proche du protocole PIR (Private Information Retrieval).

Dans la première partie de ce chapitre, nous revenons en détail sur une attaque vis a vis la solution proposée dans le chapitre 2). Cette attaque permet au cloud d'obtenir de l'information sur le contenu des images. Nous verrons ensuite la première solution qui passe par deux fournisseurs de cloud, sous l'hypothèse que ceux-ci ne colludent pas. La complexité de cette solution, comme son analyse de sécurité et ses performances de retrouvaille en termes de précision moyenne sont étudiées.

Dans la deuxième partie de ce chapitre, porte sur la seconde solution qui fonctionne avec un seul serveur et qui n'a pas besoin de communiquer avec l'utilisateur pour extraire une signature chiffrée. L'analyse de sécurité et de complexité de cette approche sont aussi étudiées et comparer avec celles de la solution présentée dans le chapitre 2).

I Attaque du schéma de SCBIR à signature en clair

La méthode proposée dans le chapitre 2) consiste à extraire d'une image les histogrammes des coefficients des différentes sous-bandes de la décomposition en ondelettes de l'image. Reprenant les notations du chapitre 2), cette méthode calcule pour une sous-bande de coefficients C_u^d (où d est le niveau de décomposition, u représente les sous-bandes de détails ou de l'approximation) de l'histogramme $H_{C_u^d}$. Pour des histogrammes de K classes, de largeur Δ (où Δ est le pas de quantification), la signature d'une image résulte de la concaténation des cardinalités de chaque histogramme, i.e. $\{H_{C_u^d}(k)\}_{1\leq k\leq K}$.

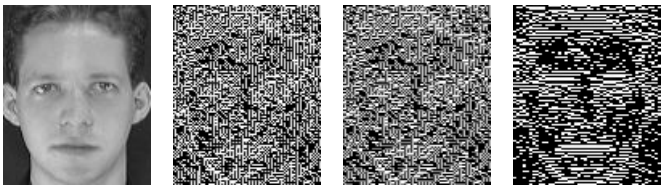
Bien que la solution proposée dans le chapitre 2 travaille sur des données chiffrées, elle souffre de deux problèmes de sécurité. Le premier réside dans le fait que la cardinalité d'une classe d'un histogramme $H_{C_d^u}(k)$ n'est pas chiffrée et le second est lié au fait que les centres des classes d'un histogramme $\{T_k\}_{1\leq k\leq K}$ sont ordonnés. Ainsi, l'histogramme $H_{C_d^u}$ d'une sous-bande apparaît en clair. Même si la dynamique des coefficients est inconnue du serveur, une connaissance *a priori* quant à leur distribution donne au cloud une indication sur le contenu des données chiffrées. Par exemple, il est bien connu que les coefficients d'ondelettes de détails suivent une distribution gaussienne ou laplacienne centrée en zéro. Sur cette base, un attaquant peut alors estimer les valeurs des centres de l'histogramme afin de construire une version approchée de l'image originale. Nous donnons en Figure 3.1 un exemple de ce type d'attaque de reconstruction où le pirate a remplacé par des valeurs approximatives en clair les coefficients de détail de l'ondelette de Haar chiffrés en fonction des classes auxquelles les coefficients appartiennent. Pour construire cet exemple, les valeurs chiffrées autour de zéro ont été remplacées par 0. Comme on peut le voir, même si la sous-bande d'approximation est inconnue et quel que soit le pas de quantification, il est possible d'identifier que l'image chiffrée correspond au visage d'une personne. Cette attaque est encore plus efficace si l'histogramme de la sous-bande d'approximation est connu. Nous donnons un exemple en Figure 3.2 où les coefficients d'approximation du $2^{ème}$ niveau de décomposition de l'image à l'aide de Haar ont été remplacés par des valeurs arbitraires. Ces valeurs prennent en compte la dynamique d'un tel coefficient est telle que $[0, 256]$

Pour cet exemple, qui traite des images en niveaux de gris codés sur 8 bits, la formule que nous avons prise pour substituer la valeur d'un coefficient chiffré est la suivante :

$$C_a(x,y) = [\frac{C_u^d(x,y)}{\Delta}]\Delta \tag{3.1}$$

Où $C_a(x,y)$ est la valeur de substitution du coefficient $C_u^d(x,y)$ et Δ et le pas de quantification. Comme on peut le voir, on fait l'hypothèse que les K classes couvrent toute la dynamique du coefficient.

Il y a donc un besoin de sécuriser l'histogramme lui-même. C'est-à-dire que le cloud ne doit pas pouvoir connaitre les cardinalités de l'histogramme ni pouvoir ordonner les classes.



(a) (b) (c) (d)

FIGURE 3.1 – Exemple d'attaque de reconstruction de l'image (a) à partir de ses coefficients d'ondelettes de détail chiffrés en considérant deux niveaux de décomposition et différents pas de quantification : (b) $\Delta = 32$, (c) $\Delta = 64$, (d) $\Delta = 128$

II SCBIR a signature chiffrée sur la base de deux Cloud

II.1 Architecture du système

L'architecture et le mode d'emploi du système CBIR sécurisé considéré ici sont les mêmes que ceux présentés dans le chapitre précédent (cf. section II - Figure 3.3). Une image requête

(a) (b) (c) (d)

FIGURE 3.2 – Exemple d'attaque de reconstruction de l'image (a) à partir de ses coefficients d'ondelettes d'approximation chiffrés en considérant deux niveaux de décomposition et différents pas de quantification : (b) $\Delta = 32$, (c) $\Delta = 64$, (d) $\Delta = 128$

est envoyée par le client au serveur dans l'objectif de trouver des images de contenu similaire et de diagnostic connu. L'image requête et les images stockées dans la base de données du serveur sont chiffrées à l'aide du cryptosystème de Paillier.

Le calcul des signatures et leur comparaison sont entièrement réalisés par les deux clouds. Nous en rappelons ici les grands principes (voir Chapitre 2) – section III pour plus de détails). Ainsi, une signature est extraite d'une image en deux temps dans le domaine chiffré : 1) calcul de la transformée en ondelettes de l'image chiffrée. 2) calcul des histogrammes des sous-bandes d'ondelettes chiffrées. Une fois les signatures extraites, elles sont comparées selon la distance L^1. La différence entre ce système et celui proposé dans le chapitre 2 réside dans le fait que les signatures d'images ne sont plus en clair mais chiffrées.

Pour réaliser ces différentes opérations nous proposons d'exploiter, comme illustré en Figure 3.3, deux fournisseurs de cloud indépendants. Les différentes fonctions de CBIR sécurisées sont partagées entre deux serveurs P_1 et P_2, chacun sous la responsabilité d'un seul fournisseur de services. À noter cependant, que P_1 jouera un rôle privilégié, c'est lui qui est en interface avec le client et qui a la base de données d'images. P_1 aura pour tâche de réaliser un certain nombre de traitements que P_1 ne peut faire sans mettre en danger la confidentialité des données ou le respect du droit à la vie privée des patients.

Comme précédemment nous nous plaçons dans le cas les serveurs des clouds sont semi-honnêtes. Nous rappelons que dans un contexte, notre objectif est d'assurer la confidentialité des informations suivantes :

— Les données stockés dans la base de données du serveur

— les requêtes d'utilisateurs

— les signatures extraites d'images

II.2 Calcul et comparaison de signatures dans le domaine chiffré

II.2.1 Extraction de signature (Principe de calcul de signature)

Dans notre scénario, l'utilisateur présente une requête sous forme une image chiffrée au serveur P_1. Le calcul des signatures d'images et leur comparaison sont répartis entre les deux serveurs P_1 et P_2. L'idée ici est d'extraire des histogrammes dont les cardinalités sont chiffrées homomorphiquement, permettant leur comparaison de manière elle aussi sécurisée.

Soit $C_u^d(x, y)$ un coefficient d'ondelettes à la position (x, y) dans une sous-bande u de détail ou d'approximation, i.e. $u \in \{hh, gh, hg, gg\}$, au niveau de décomposition d. Pour construire l'histogramme $H_{C_u^d}$ de la sous-bande C_u^d, la dynamique des coefficients est d'abord subdivisée en

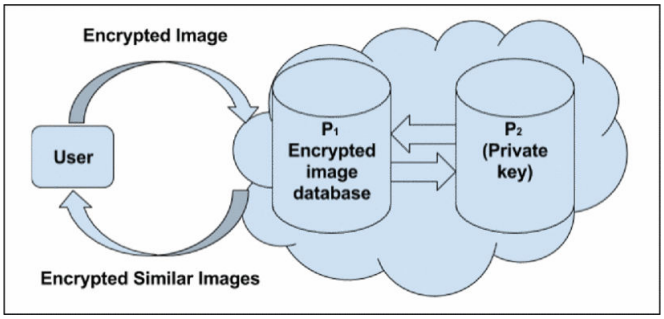


FIGURE 3.3 – Requête-réponse d'un système CBIR externalisé sécurisé à l'aide de deux fournisseur de cloud indépendants.

K intervalles uniformes ou classes de taille Δ où Δ est le pas de quantification. Par définition, la valeur $H_{C_u^d}(k)$ indique le nombre de coefficients $C_u^d(x,y)$ dont les valeurs appartiennent au $k^{ème}$ intervalle de $H_{C_u^d}$. C'est la cardinalité de la classe C_k. Si T_k désigne le centre du $k^{ème}$ intervalle de $H_{C_u^d}$, alors la classe C_k de $C_u^d(x,y)$ est donnée par :

$$k = \arg\min_p |C_u^d(x,y) - T_p| \tag{3.2}$$

Pour présenter notre solution et pour des questions de simplicité, nous utiliserons la notation de chiffrement $E[m]$ à là place de $E[m,r]$. La notion d'aléa n'est pas nécessaire dans les explications qui vont suivre.

Dans notre approche, le calcul de l'histogramme chiffré $E[H_{C_u^d}]$ ou plus précisément des cardinalités chiffrées de ce dernier : $\{E[H_{C_u^d}(1)], ..., E[H_{C_u^d}(K)]\}$, est partagé entre les deux serveurs semi-honnête P_1 et P_1 (voir Figure 3.3). Après la réception de l'image du client, P_1 calcule la transformée en ondelettes dans le domaine chiffré de Paillier de la même manière que celle décrite dans le chapitre 2 Section III. En effet, cette transformée n'impliquant que des calculs linéaires (additions et multiplications), il est facile de la calculer dans le domaine chiffré. Ce calcul fait, la tâche suivante consiste à construire l'histogramme chiffré. Pour éviter que P_1 ne connaisse la classe d'appartenance d'un coefficient et à terme l'histogramme en clair, il va interagir avec le serveur P_2. En fait, c'est P_2 qui va identifier la classe du coefficient chiffré. Il retournera cependant une information à P_1 qui ne pourra rien en déduire sur la classe du coefficient. Considérant une sous-bande d'ondelette et un histogramme de K classes, le principe de cet échange ou protocole est le suivant :

1. P_1 chiffre les centres $\{T_p\}_{p=1...K}$ des classes de l'histogramme $H_{C_u^d}$: $S = \{E[T_1], ..., E[T_K]\}$, qu'il transmet à P_2.
2. Pour un coefficient choisit de manière aléatoire dans la sous-bande C_u^d, $E[C_u^d(x,y)]$ deux sous-étapes sont considérées :
   (a) P_1 calcule les différences chiffrées entre le coefficients et les seuils, c-à-d, sur la base de la méthode vue dans le chapitre 2, section III :

   $$S' = \{E[C_u^d(x,y) - T_p]\}_{p=1...K} = \{E[C_u^d(x,y)]E[T_p]^{-1}\}_{p=1...K} \tag{3.3}$$

   comme dans (3.2), ensuite il l'envoie à P_2. À partir de S', il sera difficile à P_2 de savoir la valeur exacte des centres $\{T_p\}_{p=1...K}$ ou bien la valeur du coefficient $C_u^d(x,y)$.

(b) En se basant sur la connaissance de la clé privée K_s par P_2, ce dernier déchiffre S', trouve la position de la valeur minimale de $D[S']$ (où $D[]$ est la fonction de déchiffrement). Après il génère un vecteur de la même taille que S' dont toutes les composantes sont mises à zero sauf la composante qui se trouve à la même position que celle du min de S'' prend la valeur un. Ce vecteur est après chiffré composante par composante en un vecteur $\beta_{x,y}$ en utilisant la clé publique K_p. $\beta_{x,y}$ est envoyé à P_1.

(c) P_1 stocke les vecteurs $\beta_{x,y}$ pour chaque coefficient dans une sous-bande donnée. Cette procédure est répétée jusqu'à ce que tous les coefficients sont traités.

3. Dès que tous les coefficients de la sous-bande $C_u^d(x,y)$ sont traités, P_1 multiplie les vecteurs stockés $\beta_{x,y}$ entre eux pour calculer l'histogramme chiffré $E[H_{C_u^d}]$. Grâce aux propriétés d'homomorphie du cryptosystème de Paillier, la multiplication entre les vecteurs $\beta_{x,y}$ est équivalente à la somme de tous les composantes de même position des vecteurs, ce qui revient au calcul du cardinal des classes de l'histogramme $\{H_{C_u^d}(p)\}_{p=1...K}$.

Dans cette procédure, comme P_1 n'a aucune connaissance de la clé privée K_s, il ne peut pas déchiffré ni le résultat fourni par P_2 ni l'histogramme chiffré (produit des vecteur $\beta_{x,y}$). En même temps, puisque P_1 choisit d'une manière aléatoire les coefficients de chaque sous-bande, alors P_2 n'a aucune idée sur les valeurs des coefficients d'ondelettes, de quelle sous-bande appartiennent et de leurs positions dans la sous-bande. Cette approche néanmoins n'est valide que si P_1 et P_2 ne colludent pas car P_2 connait la clé privée.

II.2.2 Comparaison de signatures

Afin de comparer deux images, on doit calculer la distance L^1 entre les histogrammes $E[H_{C_u^d}]$ et $E[H'_{C_u^d}]$ où plus clairement la somme cumulée des valeurs absolues des différences entre les cardinalités des histogrammes des différentes sous-bandes d'ondelettes. Pour ce faire, et pour une sous-bande, on fait interagir P_1 et P_2 sur la base de la procédure suivante :

1. P_1 multiplie, composante par composante, l'histogramme associé à la sous-bande de l'image requête, avec l'inverse modulaire de l'histogramme chiffré d'une image de la base de donnée, ce qui est équivalent au calcul de la différence entre chaque composantes de ces histogrammes dans le domaine en clair, c.-à-d :

$$\begin{aligned} E[H_{C_u^d} - H'_{C_u^d}] &= \{E[H_{C_u^d}(p) - H'_{C_u^d}(p)\}_{1\leq p\leq K} \\ &= \{E[H_{C_u^d}(p)]E[H'_{C_u^d}(p)\}^{-1}_{1\leq p\leq K} \end{aligned} \tag{3.4}$$

où K est le nombre de classes de l'histogramme.

2. P_1 multiplie chaque composante de $E[H_{C_u^d} - H'_{C_u^d}]$ par un nombre aléatoire r_p strictement positif, issu d'un vecteur aléatoire $R = \{r_p > 0, 1 \leq p \leq K\}$:

$$E[R.(H_{C_u^d} - H'_{C_u^d})] = \{E[r_p.(H_{C_u^d}(p) - H'_{C_u^d}(p))\}_{1\leq p\leq K} \tag{3.5}$$

P_1 envoie ensuite $E[R.(H_{C_u^d} - H'_{C_u^d})]$ à P_2.

3. P_2 connaissant la clé privée, déchiffre les données. Il calcule les valeurs absolues, chiffre le résultat avec la clé publique K_p, $E[|R.(H_{C_u^d} - H'_{C_u^d})|]$ et envoie les résultat à P_1. R étant inconnu de P_2, ce dernier n'a aucune idée de la valeur exacte des cardinalités de l'histogramme.

4. P_1 multiplie $E[|R.(H_{C_u^d} - H'_{C_u^d})|]$ par l'inverse de R dans $\mathbb{Z}^*_{K_p}$ pour obtenir $A = E[|H_{C_u^d} - H'_{C_u^d}|]$. La distance L^1 chiffrée entre les histogrammes est obtenue en multipliant toutes les composantes de A, ce qui est équivalent dans le domaine en clair à la somme cumulée des valeurs absolues des différences ou plus clairement la distance L^1 entre deux histogrammes sous forme chiffrée, i.e. $E[d] = E[L^1(H_{C_u^d}, H'_{C_u^d})]$.

   Signalons que P_2 n'a aucune idée sur $E[d]$. Pour déduire de l'information, il aurait besoin de connaitre R. Cela n'est pas possible car il est choisi par P_1.

Après le calcul de toutes les distances L^1 entre les histogrammes des transformées en ondelettes de l'image requête et ceux associés aux L images de la base de données (i.e. $\{E[d_i]\}_{i=1...L}$), P_1 doit encore besoin d'identifier les distances les plus petites pour retourner à l'utilisateur les images les plus similaires à son image requête. Pour réaliser cette tâche, nous proposons d'utiliser une méthode de comparaison basée sur le masquage. Celle-ci fonctionne comme suit. Pour comparer $E[d_i]$ et $E[d_j]$, P_1 sélectionne dans un premier temps deux valeurs aléatoires r et r' de $\mathbb{Z}_{K_p}$ telles que r' est significativement plus petite que r (c'est-à-dire $r >> r'$). Puis, P_1 calcule :

$$E[r(d_i - d_j) - r'] = (E[d_i]E[d_j]^{-1})^r \times E[r']^{-1} \tag{3.6}$$

et envoie ce résultat à P_2. P_1 déchiffre ce message, compare les données et envoie un bit i à P_1 tel que i égale à 1 si $r(d_i - d_j) - r' > 0$ ou à 0 sinon. À la réception, P_1 peut identifier la distance minimale de toutes les distances $\{E[d_i]\}_{i=1...L}$ où L est la taille de la base de données. Ensuite, P_1 retourne à l'utilisateur les images chiffrées ayant les distances minimales par rapport à l'image requête.

II.3 Analyse de complexité et de sécurité

II.3.1 Analyse de complexité

Il est important de noter que notre approche est valable pour tous les cryptosystèmes homomorphiquement additifs. Elle ne nécessite pas de communications entre l'utilisateur et les serveurs pour le calcul de la signature. Par conséquent, la complexité de calcul de l'utilisateur est bornée par $O(m \times n)$ chiffrements où $m \times n$ est la dimension de l'image.

La complexité de calcul des serveurs est liée à trois étapes. La première porte sur le calcul de l'histogramme chiffré. Elle nécessite : le chiffrement des K centres $\{T_p\}_{p=1...K}$ pour trouver S ; le calcul des différences entre les coefficients et les centres des classes S' ; le déchiffrement par P_2 pour trouver la valeur minimum de chaque vecteur S' ; la génération du vecteur $\beta_{x,y}$ pour chaque coefficient $C_u^d(x, y)$ et la multiplication des vecteurs $\beta_{x,y}$ entre eux pour chaque coefficient afin de générer l'histogramme chiffré. Puisque les complexités d'une multiplication modulaire et du calcul d'inverse modulaire sont inférieures à celle d'un chiffrement, la complexité de calcul pour la construction de l'histogramme chiffré est bornée par $O(m \times n \times K)$ chiffrements.

La deuxième étape consiste à comparer les histogrammes chiffrés qui implique le calcul de la distance L^1 entre les histogrammes. Le calcul de la distance L^1 correspond à K chiffrements, déchiffrement et calcul d'inverse modulaire pour le calcul de la valeur absolue, puis K multiplications pour le calcul de la distance L^1.

La troisième étape porte sur la comparaison des distances L^1 calculées pour trier les images les plus proches. La complexité de cette étape est bornée par $O(L)$ déchiffrements où L est la taille de la base de données. Puisque la complexité d'un déchiffrement et équivalente à celle d'un chiffrement, la complexité de calcul des serveurs de notre système est bornée par $O(m \times n \times K + L)$ chiffrements.

En termes de complexité de communication entre les serveurs P_1 et P_2, celle-ci est bornée par $O((m \times n \times K + L \times K + L)\log_2(K_p))$ en bits, où K est le nombre de classes des histogrammes et K_p est le module du cryptosystème de Paillier.

II.3.2 Analyse de sécurité

Dans notre schéma, nous avons fait l'hypothèse que les deux serveurs P_1 et P_2 (ou de manière équivalente deux fournisseurs de cloud) sont semi-honnêtes indépendants. Comme expliqué dans le chapitre 1, ce modèle d'adversaire suppose que les entités impliquées suivent les étapes du protocole mais qu'ils sont curieux et qu'ils peuvent essayer d'inférer des informations sur les données des utilisateurs. Il est approprié dans notre cas car il est difficile d'imaginer qu'un fournisseur de services cloud bien établi (e.g. Google, Amazon, Microsoft) tente de falsifier les données de ses clients. Les dommages en termes d'images et de réputation seraient désastreux économiquement.

Notre système de CBIR externalisé et sécurisé permet d'effectuer une recherche dans une base de données d'images chiffrées. Si la base de données est gérée par un serveur (P_1) le processus de recherche est lui réparti entre deux serveurs (P_1 et P_2). Ces deux serveurs, sont curieux mais ne colludent pas, et peuvent chercher à trouver : la clé privée de l'utilisateur (pour P_1), les valeurs en clair des images ou de leurs histogrammes (c.-à-d. la signature des images).

Avant d'entrer dans les détails de cette analyse, rappelons les principales étapes de notre système ainsi que toutes les informations auxquelles les clouds ont accès. Comme décrit précédemment, en plus de la clé publique de l'utilisateur, il apparait que

— P_1 possède les images chiffrées dans sa base de données et qu'il reçoit les images chiffrées de l'utilisateur en requête.

— P_2 possède la clé privée de l'utilisateur.

La confidentialité des images et de leurs transformées en ondelettes reposent sur la sécurité du cryptosystème de Paillier. Nous l'avons vu dans le chapitre 2 section IV, la clé privée de l'utilisateur (K_s) est sécurisée et ne peut pas être récupérée par P_1 à partir des données chiffrées [87].

Concernant la sécurité du calcul de la classe d'appartenance d'un coefficient d'ondelette chiffré, $E[C_u^d(x,y)]$, le serveur P_1 calcule $S' = \{E[C_u^d(x,y) - T_p]\}_{1\leq p\leq K} = \{E[C_u^d]E[T_p]^{-1}\}_{1\leq p\leq K}$. S' est la version chiffrée des différences entre C_u^d et T_p, pour $1 \leq p \leq K$ où $\{E[T_p]\}_{1\leq p\leq K}$, sont les chiffrés des centres des classes de l'histogramme. P_2 déchiffre S', accède aux différences mais ne peut pas retrouver la valeur du coefficient ou bien des centres. À noter que si P_2 peut savoir qu'il est en train de calculer la classe d'un coefficient, du fait des opérations qu'il réalise, il ne connait pas la position et la sous-bande à laquelle appartient le coefficient.

A la suite de cette étape, P_2 envoie à P_1 les vecteurs $\beta_{x,y}$ chiffrés. Leur confidentialité est assurée ainsi que celle de leur multiplication qui consiste finalement à calculer l'histogramme chiffré $E[H_{C_u^d}]$. En ce qui concerne, le calcul de la distance L^1 entre deux histogrammes chiffrés, elle est masquée par un vecteur d'aléas R choisi dans $\mathbb{Z}_{K_p}$ par P_1. Donc P_1 et P_2 n'ont aucune information sur la distance L^1 entre deux histogrammes chiffrés. Afin de trouver les distances minimales de l'ensemble $\{E[d_i]\}_{i=1..L}$ où L est la taille de la base de données, le calcul encore ici est masqué par les aléa r et r' choisi par P_1 de manière aléatoire (voir Section II.2). Ce qui ne permet pas à P_2 d'inférer des informations sur les distances $\{E[d_i]\}_{i=1..L}$ entre l'histogramme de l'image requête et les images de la base de données de P_1. En résumé, Puisque tous les calcul sont effectués sur des données chiffrées ou masquées. Donc P_1 et P_2 n'ont aucune information sur l'image requête, les images de la base de données, les distances $\{E[d_i]\}_{i=1..L}$, et les coefficients d'ondelettes.

II.4 Résultats expérimentaux

Ce schéma a été testé sur les mêmes jeux de données et dans les mêmes conditions que le système proposé dans le chapitre 2. Deux applications sont considérées : l'aide au diagnostic et l'authentification de personne par reconnaissance de visage. Dans les deux cas, les données sont stockées chiffrées dans le cloud à l'aide du cryptosystème de Paillier. L'utilisateur chiffre une image et l'envoie au serveur P_1 en requête qui lui retourne les images les plus similaires identifiées dans sa base de données.

Pour ces tests, nous avons utilisé la transformée en ondelettes $2D$ de Haar jusqu'à deux niveaux de décomposition (c.-à-d $d = 0, 1, 2$). Les performances de retrouvailles du système sont évaluées en termes de précision moyenne de CBIR de cinq, c'est-à-dire que, pour une image requête, le serveur renvoie les cinq images les plus similaires qu'il trouve dans sa base de données. Comme nous le verrons, cette mesure variera en fonction du nombre de classes considérées dans les histogrammes.

II.4.1 Performance du CBIR sécurisé avec des images médicales

La Figure 3.4 présente la performance de recherche de notre schéma en termes de précision moyenne de cinq. Nous comparons également notre système avec la même approche de CBIR dans le domaine en clair en considérant différents niveaux de décomposition d'ondelettes (i.e.$d = \{0, 1, 2\}$) et plusieurs valeurs de pas de quantification de l'histogramme (i.e. $\Delta \in \{1, 2, 4, 8, 16, 32, 64\}$). À noter également que toutes les courbes sont tracées en moyenne, en utilisant 200 images de l'ensemble de test de rétine comme images requête ; les 800 autres images constituent la base de données du serveur. Comme on peut le voir, notre système a la même performance que le système CBIR dans le domaine en clair. Un tel résultat peut être expliqué par le fait que les coefficients des filtres de la transformée en ondelettes de Haar dans les domaines chiffrés et clairs sont équivalents. L'expansion des coefficients n'a pas d'impact sur la précision de la transformée en ondelettes. Au-delà, si dans le domaine en clair, on peut travailler avec $\Delta = 1$, c'est-à-dire, en travaillant avec la signature la plus précise, en utilisant des valeurs de Δ inférieures à 16 n'affecte pas la performances de recherche. Rappelons également que si le Δ est petit alors la signature est plus précise.

II.4.2 Performance du CBIR sécurisé avec des images biométriques

De la même manière que précédemment, notre base de données d'images de test a été divisée en deux ensembles : 200 images faciales sont utilisées comme images requêtes tandis que les 200 autres sont utilisées comme images de référence par le serveur. La Figure 3.5 présente la précision moyenne en fonction de différentes valeurs de pas de quantification (c.-à-d. $\Delta \in \{1, 2, 4, 8, 16, 32, 64\}$) et des niveaux de décomposition (i.e. $d = \{0, 1, 2\}$). À nouveau, on peut voir que les moyennes de précision sont équivalentes dans le domaine en clair et chiffré.

II.5 Conclusion

Dans cette partie, nous avons proposé une nouvelle approche pour sécuriser un système de CBIR fondé sur une signature globale qui correspond aux histogrammes des coefficients d'ondelettes des images. Il permet d'effectuer une recherche dans une base de données d'images chiffrées externalisées. Contrairement au schéma de CBIR proposé dans le chapitre 2, il permet l'extraction de signature globale sécurisée à partir d'image chiffrée. Pour ce faire, il profite de deux fournisseurs de cloud entre lesquels le calcul des histogrammes chiffrés et leurs comparaisons sont partagés. Les résultats expérimentaux montrent que le fait de travailler sur des données

h

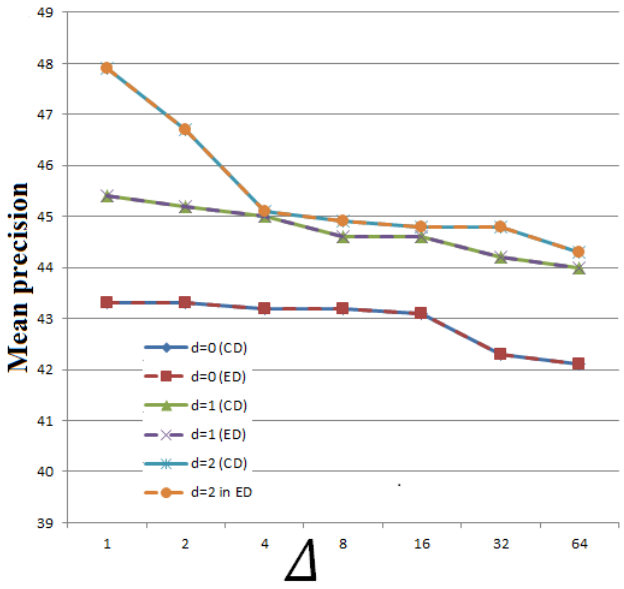


FIGURE 3.4 – Performance de recherche exprimée en précision moyenne à cinq (en %) pour notre base de données médicales en considérant différentes valeurs de pas de quantification (Δ) et différents niveaux de décomposition d. Les courbes pointillées désignent les résultats de SCBIR dans le domaine chiffré (ED "Encrypted Domain"), tandis que les courbes pleines donnent la performance du système CBIR équivalent dans le domaine en clair (CD "Clear Domain").

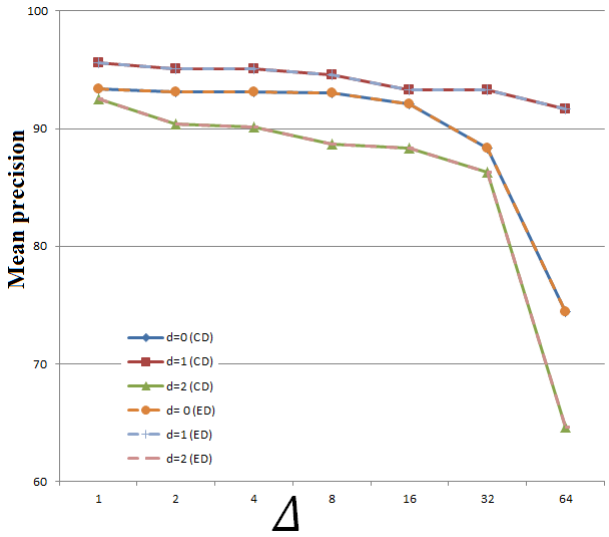


FIGURE 3.5 – La performance de recherche exprimée en précision moyenne à cinq (en %) pour notre base de données d'images biométrique en considérant différentes valeurs de pas de quantification (Δ) et niveaux de décomposition d. Les courbes en traits pointillés désignent les résultats de SCBIR dans le domaine chiffré (ED "Encrypted Domain"), tandis que les courbes pleines donnent la performance du système CBIR équivalent dans le domaine en clair (CD "Clear Domain").

chiffrées n'a pas d'impact sur la performance de recherche par rapport à celle dans le domaine en clair à la condition que la transformée en ondelette dans le domaine chiffrée équivaut celle dans le domaine en clair (i.e. pas de pertes de précision liées au processus d'expansion). Si notre solution garantit la confidentialité des données externalisées, sous l'hypothèse que les fournisseurs de cloud ne colludent pas, son utilisation dans la pratique nécessite des capacités de calcul du cloud importantes. Un autre défaut de cette solution est qu'elle nécessite deux fournisseurs de cloud ce qui peut engendrer un sur coût économique. Par ailleurs, le service proposé peut ne plus être opérationnel en cas de de pertes de communications.

III CBIR fondé sur un seul Cloud et du chiffrement homomorphe

III.1 Extraction d'un histogramme sécurisé : principes de base

Nous rappelons que la solution proposée dans le chapitre 2 permet de construire un histogramme $H_{C_u^d}$ de K classe en claire à partir d'une sous-bande chiffrée C_d^{eu}. Nous avons montré dans la partie 1 de ce chapitre, que le fait de calculer l'histogramme en clair peut fuiter de l'information sur le contenu de l'image (par exemple la forme de l'objet). Pour résoudre ce problème nous avons proposé dans la partie 1 de ce chapitre une solution qui permet de construire un histogramme chiffré d'une image chiffrée en utilisant deux fournisseurs de cloud indépendants. L'approche que nous proposons maintenant vise à construire un histogramme sécurisé $H^E_{C_u^d}$ dont les cardinalités de classe sont chiffrées (c'est-à-dire $E[H^E_{C_u^d}(k), r_k]_{1\leq k\leq K}$) et dont les centres de classe non ordonnés. Cette approches repose sur deux étapes principales : la première consiste à construire un histogramme bruité $H^N_{C_u^d}$ de la transformée en ondelettes d'image chif-

frée ; histogramme à partir duquel l'histogramme sécurisé $H^E_{C^d_u}$ sera calculé dans la deuxième étape. La construction de l'histogramme bruité a pour but de briser la répartition des coefficients d'ondelettes sur une dynamique plus grande, tout en donnant accès à un histogramme dont les cardinalités de classe sont en claire (c.-à-dire. non chiffrées). Notre solution est équivalente à l'addition d'un bruit aux coefficients de sous-bande d'ondelettes et calculer ensuite l'histogramme des coefficients bruités. Dans la suite nous expliquons en détails notre approche qui nécessite un seul fournisseur de cloud et sans extra communication avec l'utilisateur.

La solution précédente permet de calculer et de comparer de manière sécurisée des signatures extraites d'images chiffrées. Elle règle le problème de la solution proposée dans le chapitre 2 qui construit des histogrammes $(H_{C^d_u})$ de K classes en clair à partir des sous-bandes de coefficients d'ondelettes chiffrée (C^{ed}_u) ; solution qui peut faire l'objet d'une attaque spécifique comme illustré en début de ce chapitre. Cependant, cette approche fait appel à deux fournisseurs de cloud sous l'hypothèse que ces derniers sont indépendants et ne colludent pas. Cette solution est aussi sensible aux pertes de communications entre fournisseurs de cloud.

Dans la deuxième partie de ce chapitre, nous présentons une stratégie qui permet de construire un histogramme sécurisé $H^E_{C^d_u}$, i.e. dont les cardinalités des classes sont chiffrées (c'est-à-dire $E[H^E_{C^d_u}(k), r_k]_{1\leq k\leq K}$) et dont les centres de classe sont non-ordonnés, à l'aide d'un seul fournisseur de cloud et sans communications avec le client.

Considérant une seule sous-bande de coefficients d'ondelettes chiffrée, l'extraction d'un tel histogramme sécurisé repose sur deux étapes. La première consiste à construire à partir des coefficients chiffrées un histogramme « bruité » $H^N_{C^d_u}$. Comme nous le verrons, la construction de l'histogramme bruité a pour but de secrètement « casser » la distribution des coefficients d'ondelettes sur une dynamique plus grande, tout en donnant accès à un histogramme dont les cardinalités sont en clair (c.-à-dire. non chiffrées) et uniformément distribuées. Cette solution est équivalente à l'addition d'un bruit aux coefficients d'ondelettes puis calculer l'histogramme des coefficients bruités en utilisant l'approche proposée dans le chapitre 2. L'histogramme sécurisé $H^E_{C^d_u}$ sera calculé à partir de cet histogramme bruité en utilisant d'une approche de débruitage qui profite des propriétés d'homomorphie du chiffrement de Paillier.

III.2 Extraction d'une signature chiffrée avec zéro-communication

Comme évoqué précédemment, l'idée est d'ajouter un bruit aux coefficients d'ondelette, puis à calculer l'histogramme en clair de ces derniers. L'ajout du bruit a pour but de masquer la distribution des coefficients d'ondelettes et empêcher un attaquant de dérouler l'attaque présentée en Section I à partir de l'histogramme en clair.

Soit un coefficient d'ondelettes $C^d_u(x, y)$ et $N(x, y)$ un entier choisit aléatoirement dans l'intervalle $[N_{min}, N_{max}]$ tel que $N_{max} > C_{max}$, où C_{max} est la valeur maximale des coefficients d'ondelettes de la sous-bande C^d_u. $N(x, y)$ suit une distribution uniforme. L'addition de $N(x, y)$ à $C^d_u(x, y)$ conduit à une variable aléatoire $C^{dN}_u(x, y) = C^d_u(x, y) + N(x, y)$, dont la distribution est assez proche d'une variable aléatoire uniformément distribuée. Comme nous le verrons lors de l'analyse de sécurité de cette approche, du fait de ce caractère uniforme, il est difficile pour un attaquant de retrouver la distribution des coefficients à partir de cet histogramme « bruité ». La Figure 3.6 illustre le résultat d'une telle procédure dans le cas d'une variable aléatoire X gaussienne centrée à zéro et d'un écart type $\sigma = 10$ à laquelle est ajouté une variable aléatoire N uniformément distribuée dans la plage $[a, b] = [-50, 50]$. La densité de probabilité f_Y de la variable aléatoire résultante $Y = X + N$ est de la forme :

$$f_Y(t) = \frac{1}{(b-a)\sqrt{2\pi\sigma^2}} \int_a^b exp(-\frac{(x-t)^2}{2\sigma^2})dx \tag{3.7}$$

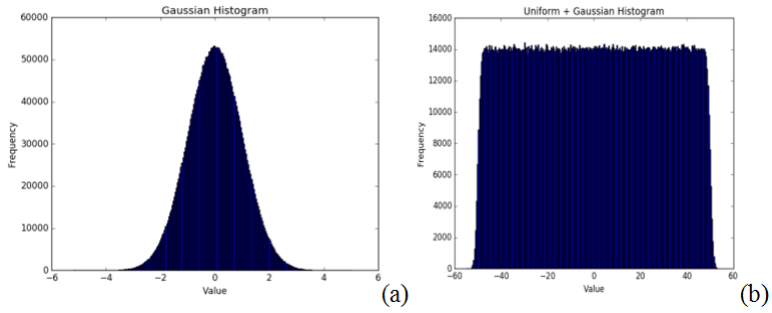


FIGURE 3.6 – Exemple d'un histogramme bruité - (a) Histogramme d'une distribution gaussienne discrète X de moyenne $\mu = 0$ et d'écart type $\sigma = 10$; (b) Histogramme de la variable aléatoire $Y = X + N$, où N est un bruit uniformément distribué dans $[-50, 50]$

Il est possible de calculer l'histogramme des coefficients $C_u^{dN}(x, y)$ dans le domaine chiffré en utilisant la solution proposée dans le chapitre 2. Supposons que la dynamique de C_u^{dN} est subdivisée en K' intervalles. Le principe de construction de l'histogramme consiste à ce que l'utilisateur envoie pour chaque coefficient les centres des classes de l'histogramme chiffrés avec le même aléa que le coefficient d'ondelette (voir chapitre 2 section III). Ici, l'utilisateur va donc envoyer avec son image chiffrée un ensemble de K' centres de classe chiffrés par coefficient d'ondelettes. Néanmoins, afin de ne pas modifier l'image chiffrée en y ajoutant le bruit et permettre son utilisation à d'autres fins, il est demandé à l'utilisateur d'ajouter $N(x, y)$ aux centres de classes d'histogramme au lieu des coefficients d'ondelettes. L'ajout du bruit aux coefficients ou aux centres est équivalent. Plus clairement, pour un coefficient $C_u^d(x, y)$, l'utilisateur envoie l'ensemble des centres de classe chiffrés $\{E[T_k + N(x, y), r_u^d]\}_{1 \leq k \leq K'}$. $H_{c_u^d}^N$ sera construit par le serveur en utilisant les ensembles de centres de classes chiffrés. Comme illustré dans l'exemple donné en Figure 3.6, on peut voir que $H_{c_u^d}^N$ est un histogramme uniforme. Il ne donne aucun indice au serveur de l'histogramme de C_u^d. L'étape suivante consiste à dériver l'histogramme chiffré $H_{C_u^d}^E$ dont le nombre de classes est K, à partir de l'histogramme bruité de classes K'. La stratégie que

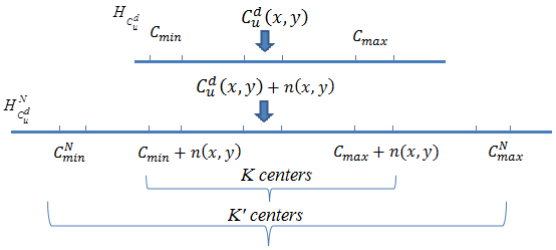


FIGURE 3.7 – Mappage entre les classes de $H_{C_u^d}^N$ et $H_{C_u^d}$ pour un coefficient d'ondelette donné $C_u^d(x, y)$ et un bruit $N(x, y)$. La dynamique de C_u^{dN} est beaucoup plus grande que celle de C_u^d où $(K' > K)$.

nous proposons pour dériver $H_{c_u^d}^E$ de $H_{c_u^d}^N$ est similaire au protocole PIR (Private Information Retrieval) [149]. Du fait de l'importance des aléas dans celle-ci, nous reprenons dans la suite de ce chapitre l'écriture des opérations de chiffrement et de déchiffrement à base d'aléas, c.à-d. $E[a, r]$ où $E[,]$ et la fonction de chiffrement et r est l'aléa associé au message a.

Considérons le calcul de la cardinalité de la $k^{ème}$ classe de l'histogramme chiffré $H^E_{C^d_u}(k)$. Dans un premier temps et pour un coefficient donné $C^d_u(x,y)$, l'utilisateur génère un vecteur $P^{T_k}_{x,y}$ de K' composantes ; composantes qui correspondent au chiffrement de la valeur $'0'$ avec des valeurs aléatoires différentes (c'est-à-dire $E[0,r_z]$) sauf sa $(k+n(x,y))^{ème}$ composante qui est fixée à $E[1,r_u]$. En fait, $P^{T_k}_{x,y}$ indique la position de la $k^{ème}$ classe en fonction des K' classes de $H^N_{C^d_u}$. Cette position est décalée en fonction du bruit ajouté aux centres des classes chiffrés. La Figure 3.7 illustre le « mapping » entre les classes des histogrammes bruité $(H^N_{c^d_u})$ et sécurisé $(H_{c^d_u})$.

De son côté, le serveur génère un vecteur $S_{x,y}$ de taille K' dont les composantes sont toutes portent la valeur 0 sauf la $l^{ème}$ qui prend la valeur 1 indiquant la classe de $H^N_{C^d_u}$ à laquelle appartient le coefficient $C^{de}_u(x,y)$. Le serveur calcule le produit scalaire entre $S_{x,y}$ et $P^{T_k}_{x,y}$:

$$S_{x,y}.P^{T_k}_{x,y} = \begin{cases} E[1,r_u] & si \quad C^d_u(x,y) \in C_k \\ E[0,r_z] & sinon \end{cases} \tag{3.8}$$

Nous rappelons que C_k est la $k^{ème}$ classe de l'histogramme de la sous bande C^d_u. Plus clairement, si $C^d_u(x,y)$ appartient à la $k^{ème}$ classe de l'histogramme, alors $S_{x,y}.P^{T_k}_{x,y}$ est égale à $E[1,r_u]$. Dans le cas contraire, le résultat du produit est $E[0,r_z]$. Sur la base de cette stratégie, nous supprimons le bruit ajouté aux centres de classe sans révéler si $C^d_u(x,y)$ appartient à la $k^{ème}$ classe de $H_{C^d_u}$ ou pas.

Pour calculer $H^E_{c^d_u}(k)$, c'est-à-dire la cardinalité chiffrée de C_k, le serveur doit simplement multiplier les résultats des produits scalaires entre eux pour une sous-bande, en profitant des propriétés d'homomorphie du cryptosystème de Paillier, c'est-à-dire

$$H^E_{C^d_u}(k) = \prod S_{x,y}.P^{T_k}_{x,y} = \prod E[q_{x,y}, r_v] = E[\sum q_{x,y}, \prod r_v] \tag{3.9}$$

où $q_{x,y} \in \{0,1\}$.

III.3 Comparaison entre deux histogrammes chiffrés de coefficients d'ondelettes

Comme montré dans la section II.2, la comparaison de deux images I^1 et I^2 stockées chiffrées dans le cloud par deux utilisateurs U_1 et U_2 repose sur le calcul de la distance L^1 entre les histogrammes de leurs sous-bandes d'ondelettes.

Soit deux histogrammes chiffrés $H^E_{C^{d(1)}_u}$ et $H^E_{C^{d(2)}_u}$ respectivement extraits des versions chiffrées de I^1 et I^2. Il est important de noter que $H^E_{C^{d(1)}_u}$ et $H^E_{C^{d(2)}_u}$ correspondent aux cardinalités $H_{C^{d(1)}_u}$ et $H_{C^{d(2)}_u}$, chiffrées avec différentes valeurs aléatoires et différentes clés publiques, celles des deux utilisateurs. Il est cependant possible de calculer $D^e_{iff}(H^E_{C^{d(1)}_u}, H^E_{C^{d(2)}_u}) = D_{iff}(H_{C^{d(1)}_u}, H_{C^{d(2)}_u})$ en utilisant la méthode de comparaison des données chiffrées proposée dans le chapitre 2.

Afin de mesurer la différence de cardinalité entre deux classes, c'est-à-dire $H^E_{C^{d(1)}_u}(k)$ et $H^E_{C^{d(2)}_u}(k)$, cette solution nécessite la comparaison de ces deux quantités avec une valeur de référence p_k ; valeur sur laquelle U_1 et U_2 se sont mis *a priori* d'accord et que chacun a chiffrée avec sa clé publique et la même valeur aléatoire, i.e., $H^E_{C^{d(1)}_u}(k) = E[H_{C^{d(1)}_u}(k), r^1_k]$ et $H^E_{C^{d(2)}_u}(k) = E[H_{C^{d(2)}_u}(k), r^2_k]$. Le problème ici est que r^1_k et r^2_k résultent de plusieurs opérations impliquées dans les calculs de $H^E_{C^{d(1)}_u}(k)$ et $H^E_{C^{d(2)}_u}(k)$. Afin de rendre cette étape possible sans introduire d'interactions entre le serveur et les utilisateurs, l'utilisateur doit envoyer d'autres

informations au serveur dont il a besoin pour d'obtenir $E[p_k, r_k^1]$ et $E[p_k, r_k^2]$. Ainsi, pour un coefficient $C_u^d(x,y)$, les deux utilisateurs génèrent un vecteur $P_{x,y}$ dont les K' composantes correspondent à la valeur p_k chiffrée avec les mêmes valeurs aléatoires que pour les composantes de $P_{x,y}^{T_k}$. Lors du calcul de $H^E_{C_u^{d(1)}}(k)$, il est demandé au serveur de calculer pour chaque coefficient le produit scalaire

$$S_{x,y}.P_{x,y} = \begin{cases} E[p_k, r_u] & si \quad C_u^d(x,y) \in C_k \\ E[p_k, r_z] & sinon \end{cases} \tag{3.10}$$

et comme pour le calcul de $H^E_{C_u^d}(k)$, de multiplier les résultats des produits scalaires pour obtenir :

$$E[\sum p_k, r_k^1] = \prod S_{x,y}P_{x,y} = \prod E[p_k, r_u] = E[\sum p_k, \prod r_u] \tag{3.11}$$

Du fait que les mêmes valeurs aléatoires ont été utilisées pour chiffrer $H^E_{C_u^{d(1)}}(k)$ et $E[\sum p_k, r_k^1]$ où $r_k^1 = \prod r_u$. En suivant la même procédure avec l'image du second utilisateur, le serveur a aussi accès à $E[\sum p_k, r_k^2]$. Supposant que les images ont les mêmes dimensions, la distance L^1 entre les cardinalités des deux classes est

$$D^e_{iff}(H^E_{C_u^{d(1)}}(k), H^E_{C_u^{d(2)}}(k)) = D_{iff}(H_{C_u^{d(1)}}(k), \sum p_k) - D_{iff}(H_{C_u^{d(2)}}(k), \sum p_k) \tag{3.12}$$

$$= H_{C_u^{d(1)}}(k) - H_{C_u^{d(2)}}(k) \tag{3.13}$$

et la distance L^1 entre les deux histogrammes chiffrés est

$$D_{iff}(H_{C_u^{d(1)}}, H_{C_u^{d(2)}}) = \sum D_{iff}(H_{C_u^{d(1)}}(k), H_{C_u^{d(2)}}(k)) \tag{3.14}$$

III.4 Système complet

Notre système de recherche par le contenu externalisé sécurisé exige qu'un utilisateur envoie avec son image chiffrée les différentes informations suivantes

- Pour que le serveur calcule les histogrammes bruités $H^N_{C_u^d}$ pour chaque sous-bandes, l'utilisateur doit fournir pour chaque coefficient, K' centres chiffrés par le cryptosystème de Paillier.
- Pour que le serveur calcule la cardinalité chiffrée de la $k^{ème}$ classe de l'histogramme chiffré $H^E_{C_u^d}(k)$, et pour la comparer à l'une des autres images, l'utilisateur envoie au serveur pour chaque coefficient
  - Un vecteur $P_{x,y}^{T_k}$ de K' composantes qui mappe la $k^{ème}$ classe de l'histogramme dans le domaine en clair avec les classes de l'histogramme bruité, $P_{x,y}^{T_k}$ sera exploité par le serveur afin de calculer $H^E_{C_u^d}(k)$.
  - Un vecteur $P_{x,y}$ qui contient une valeur de référence choisi par les différents utilisateurs qui est de taille K' et inconnu pour le serveur. Le serveur utilisera ce vecteur pour calculer la distance L^1 entre les histogrammes chiffrés d'images différentes.

Une fois que toutes les données ont été envoyées, le serveur peut calculer les différents histogrammes chiffrés sans besoin d'interagir avec un tiers de confiance ou l'utilisateur.

Pour conclure, dans le cas d'une transformée en ondelettes séparables dyadiques de d niveaux de décomposition et d'images de dimension égale à $n \times m$. En considérant que l'histogramme de sous-bande dans le domaine en clair (c-à-dire. $H_{C_u^d}$) est constitué de K classes et que l'histogramme bruité se compose de K' classes, alors le nombre de données chiffrées que l'utilisateur doit envoyer avec l'image est

$$m.n.K' + 2.K'.K[3\sum_{i=1}^{d} \frac{m}{2^i}\frac{n}{2^i} + \frac{m}{2^d}\frac{n}{2^d}] = m.n.K'.(2K+1) \tag{3.15}$$

III.5 Analyse de complexité et de sécurité

Avant d'analyser la sécurité et la complexité de cette solution, précisons que cette solution a également été testée dans les mêmes contextes applicatifs que les méthodes précédentes (i.e. l'aide à la décision en santé et l'authentification de personnes par reconnaissance faciale). En utilisant l'ondelette de Haar, les performances de retrouvailles obtenues en termes de précision moyenne sont identiques à celles-obtenues précédemment. La différence entre les trois solutions réside dans leur complexité (de calcul et de communication) et également l'analyse de sécurité.

III.5.1 Analyse de complexité

Dans le cas d'une transformée en ondelettes dyadiques et d'une image de $n \times m$ pixels, l'utilisateur doit chiffrer $n \times m + n \times m \times K \times (2K + 1)$ données (l'images et les données auxiliaires) et les envoyer au serveur. Ces données sont stockées dans la base de données du serveur. La complexité de calcul associée à l'utilisateur est bornée par $O(m \times n \times K' \times (2K+1))$ chiffrement. Cette complexité est très importante, mais les données sont envoyées une fois. Il faut noter également que toutes ces informations peuvent être utilisées à d'autres fins que SOCBIR ou pour le calcul de SOCBIR avec différentes familles d'ondelettes, par exemple.

Du côté du serveur, la complexité de calcul repose sur : le calcul de la transformée en ondelettes des images dans le domaine chiffré, la construction des différents histogrammes et le calcul de la distance L^1 entre les signatures. La complexité de calcul des deux derniers correspond en fait : au nombre de comparaisons que le serveur doit faire et qui est borné par $O(m \times n \times \log_2(K'))$; et le nombre de multiplications modulaires entre les résultats des produits scalaires de l'histogramme qui est lui borné par $O(m \times n \times K)$ chiffrements. Le serveur doit également effectuer $O(m \times n \times K)$ multiplications pour calculer la cardinalité d'une classe de l'histogramme chiffré. Pour calculer la distance L^1 entre les signatures, il faut $O(L \times K)$ soustractions, où L est la taille de la base de données. En ce qui concerne le calcul de la transformée en ondelettes, ceci impose de faire $m \times n \times \log_2(K') + 2 \times m \times n \times K$ multiplications et additions modulaires. En conséquence, la complexité de calcul du serveur est bornée par $O(m \times n \times \log_2(K') + 2 \times m \times n \times K)$ chiffrements. Rappelons que la complexité d'un chiffrement est plus élevée que celle de la multiplication modulaire ou de calcul d'inverse modulaire. La complexité de communication de cette approche est bornée par $O((m \times n \times K' \times (2K + 1)) \log_2(K_p))$ en bits où K_p est la clé publique du cryptosystème de Paillier. Cette complexité est très élevée par rapport aux approches proposées précédemment, cependant la solution proposée permet d'effectuer les calculs de CBIR sur un seul serveur cloud en assurant la confidentialité des données et sans interactions avec l'utilisateur ou un tiers de confiance.

III.5.2 Analyse de sécurité

Notre système de CBIR sécurisé externalisé permet d'effectuer une recherche d'image similaires dans une base de données d'images chiffrées gérée par un seul fournisseur de cloud. Dans ce qui suit, nous discutons la sécurité de ce système en termes de confidentialité des données en considérant un serveur de cloud semi-honnête.

Avant d'entrer dans les détails de cette analyse, rappelons les principales étapes de notre système de CBIR ainsi que les données auxquelles le cloud a accès. En plus de la clé publique de l'utilisateur, le cloud :

— Accède à l'image chiffrée envoyée en requête et aux images chiffrées de sa base de données.

— Pour chaque coefficient d'ondelette, il accède à : K' centres de classe chiffrés utilisés par le cloud pour calculer les histogrammes bruité $H^N_{C^d_u}$; un vecteur $P^{T_k}_{x,y}$ de K' composantes

chiffrées exploitées par le cloud afin de calculer les histogrammes chiffrés, c'est-à-dire $H^E_{C^d_u}$. À noter que, dans notre système, le chiffrement des K' centres de classe est synchronisé en termes d'aléas avec celui des coefficients d'ondelettes chiffrés à quantifier.

La confidentialité des images et de leurs transformées en ondelettes dépend entièrement de la sécurité du cryptosystème de Paillier. L'analyse de sécurité de ce cryptosystème a été étudiée dans [87] en regardant différents modèles d'attaques : Attaque à texte chiffré seulement, Attaque à texte clair connu, Attaque à texte clair choisi et Attaque à texte chiffré choisi. Sous tous ces modèles d'attaque, la clé privée de l'utilisateur K_s est sécurisée. Elle ne peut pas être récupérée [87]. En ce qui concerne la confidentialité de l'histogramme bruité, le cloud ne doit pas pouvoir identifier la valeur d'un coefficient d'ondelette $C^d_u(x,y)$ ou des K' centres de classes associées $\{T_k\}_{1\leq k\leq K'}$ à partir du calcul de leurs différences dans le domaine chiffré. Hsu *et al.* ont montré dans [1] que le calcul de la différence entre deux messages chiffrées avec la même valeur aléatoire est sécurisée sous l'attaque à texte chiffré seulement.

Même si le cloud est capable de connaître le pas de quantification entre ces centres à travers leurs différences, il ne sera pas capable de les « mapper » avec la dynamique des coefficients d'ondelettes. En effet, pour un coefficient, les K centres de classe de l'histogramme en clair (c.-à-d. $H_{C^d_u}$) sont associés aux K' centres de l'histogramme bruité en ajoutant une valeur aléatoire provenant d'un bruit uniforme généré par l'utilisateur. Cette procédure permet de déplacer de manière aléatoire les centres de classe de l'histogramme en clairs de $H_{C^d_u}$ sur la dynamique de l'histogramme bruité. La connaissance de la classe d'un coefficient $C^{de}_u(x,y)$ dans $H^N_{C^d_u}$ informe simplement le serveur que $H_{C^d_u}$ a été déplacé autour de cette position. Ce changement étant différent d'un coefficient à l'autre, le serveur ne peut pas identifier exactement la classe de $C^d_u(x,y)$. La probabilité de deviner la classe correcte d'un coefficient $C^{de}_u(x,y)$ de K' classe de $H^N_{C^d_u}$ est $\frac{1}{2K}$. En conséquence, la probabilité de deviner tous les coefficients d'une image de taille $m\times n$, est donnée par $(\frac{1}{2K})^{mn}$. La Figure 3.8 illustre le résultat de l'attaque de reconstruction représentée dans la section 1, où le cloud remplace les coefficients chiffrés qui se trouvent dans les mêmes centres d'histogramme par la même valeur arbitraire. Comme on peut le voir, l'image reste inintelligible. Même si le serveur a une connaissance *a priori* de la distribution des sous-bandes de coefficients d'ondelettes, il ne peut déduire aucune information sur l'histogramme bruité. Pour conclure sur ce point, le cloud ne peut pas identifier la valeur des coefficients et les centres de classe ou même déduire si deux coefficients sont identiques.

L'histogramme chiffré est dérivé de l'histogramme bruité en utilisant une stratégie similaire au PIR (Private Information Retrieval) [149] sur la base d'un ensemble de vecteurs chiffrés $\{P^{T_k}_{x,y}\}$ (un vecteur par coefficient) dont les composantes indiquent la correspondance entre les deux histogrammes. Ces composantes sont chiffrées, le cloud n'a aucun moyen de trouver le « mapping ». Plus clairement, le serveur ne peut pas identifier la classe $H^E_{C^d_u}$ d'un coefficient chiffré $C^{de}_u(x,y)$. À la fin de cette procédure, les cardinalités de l'histogramme chiffré sont aussi chiffrées. Le degré de sécurité est donc celui du cryptosystème de Paillier.

IV Comparaison entre les trois solutions proposées

Jusqu'ici, nous avons proposées trois approches permettant d'externaliser de manière sécurisée une méthode de recherche par le contenu. Ces trois approches ont des avantages et inconvénients sur lesquels nous souhaitons revenir.

Concernant les approches qui extraient des signatures globales sécurisées, la solution à base d'un seul serveur a une complexité de calcul pour l'utilisateur $K'\times(2K+1)$ fois supérieure à celle de l'approche qui s'appuie sur deux fournisseurs de cloud dont la complexité en termes de

(a) (b) (c) (d)

FIGURE 3.8 – Exemple d'attaque de reconstruction de l'image (a) à partir de ses coefficients d'ondelettes chiffrés en considérant deux niveaux de décomposition et différents pas de quantification : (b) $\Delta = 32$, (c) $\Delta = 64$, (d) $\Delta = 128$.

Les approches	Chapitre 2	Chapitre 3, partie 1	Chapitre 3, partie 2
Complexité de calcul d'utilisateur	$O(m \times n)$	$O(m \times n)$	$O(m \times n \times K' \times (2K+1))$
Complexité de calcul serveur(s)	$O(m \times n)$	$O((m \times n \times K + L \times K + L)\log_2(K_p))$	$O(m \times n \times \log_2(K') + 2 \times m \times n \times K)$

TABLE 3.1 – Complexité de calcul d'utilisateur et de serveur dans les trois approches

calcul est $O(m \times n)$ chiffrements. Pour les serveurs, la différence de complexité est $\log_2(K') + K$ fois supérieure sachant que cette complexité est $m \times n \times K + L \times K + L$ pour l'approche à deux serveurs. Il est intéressant de voir que pour la solution qui extrait une signature en clair (voir chapitre 2), sa complexité de calcul est $O(m \times n)$ chiffrement pour l'utilisateur et identique à l'approche qui extrait une signature sécurisée avec l'aide de deux clouds. Par contre, cette dernière à une complexité de chiffrement pour le serveur K fois supérieure (i.e. $O(K \times m \times n)$ chiffrement – voir table 3.1).

Pour conclure, l'approche proposée dans le chapitre 2 a une complexité de calcul et de communication plus faible par rapport à celles proposées dans le chapitre 3, mais les signatures extraites des images sont en clair permettant l'attaque décrite en début de ce chapitre. Les deux approches proposées dans ce chapitre ne sont pas sujettes à cette attaque, mais elles sont de complexité très différentes en fonction de si elles s'appuient ou non sur deux fournisseurs de cloud. Elles sont complètement sécurisées, ne fuitent aucune information mais un compromis est à trouver entre complexité de communications et complexité de calculs. Un autre point, nous n'avons pas eu le temps d'étudier, porte sur les performances de retrouvailles qui sont intimement liées à la possibilité d'exprimer la transformée en ondelette dans le domaine chiffré. Dans toutes nos expérimentations, l'ondelette de Haar a été utilisée. Son expression dans le domaine chiffré est sans erreur sur la base d'un facteur d'expansion bien choisi (voir section II.5). L'utilisation d'autres ondelettes peut poser problème. Il faudra alors établir un compromis entre la complexité de calcul, les performances de retrouvailles et la confidentialité ou la sécurité des données.

V Conclusion

Dans ce chapitre, nous avons montré comment il est possible d'attaquer un schéma de CBIR sécurisé qui calcul des signatures en clair et découvrir le contenu de l'image. Nous avons ensuite présenté deux nouvelles méthodes de recherche par le contenu sécurisées, qui permettent de résoudre ce problème. Ces dernières externalisent l'ensemble d'un processus de recherche d'images dans une base de données d'images chiffrées maintenue par un serveur. L'originalité de ces approches réside dans le fait que toutes deux extraient des signatures chiffrées d'images chiffrées et comparent ses signatures sans divulguer d'information sur le contenu de ces signatures. Nous avons fait l'analyse de sécurité et de la complexité de ces deux approches. Ces deux solutions ne permettent pas à un cloud honnête mais curieux de pouvoir déduire des informations concernant les données des utilisateurs.

Si aucune de ces deux approches n'impliquent de communications supplémentaires avec l'utilisateur, elles sont de complexité de calcul et de communication très différentes. La première, qui s'appuie sur deux fournisseurs de services est de plus faible complexité de calcul mais est dépendante de communications entre les deux serveurs qui ne doivent surtout pas colluder. En effet, un des deux serveurs connait la clé privée de l'utilisateur. La seconde approche n'exploite qu'un seul serveur. Son originalité réside d'une part dans le calcul d'un histogramme « bruité » à partir duquel le cloud ne peut déduire aucune information et, d'autre part, dans l'utilisation d'un schéma similaire au PIR (Private Information Retrieval) qui permet de construire un histogramme chiffré à partir de cet histogramme bruité. La complexité de cette solution est importante mais a l'avantage d'être indépendante de toute communication avec un tiers de confiance (e.g. un autre cloud). Les résultats expérimentaux montrent que notre système SOCBIR réalise les mêmes performances que l'approche CBIR dans le domaine en clair.

Chapitre 4

Apprentissage automatique sécurisé

Nous avons vu dans les deux chapitres précédents comment, en utilisant le chiffrement homomorphe combiné avec du calcul multipartite sécurisé, on peut sécuriser un système de recherche d'images par le contenu qui s'appuie sur des signatures d'images dont les calculs est connu *a priori*. Dans ce chapitre, un autre type d'approches qui permet de trouver des images semblables sur la base de techniques d'apprentissage automatique (dites de " machine learning"). Le déploiement de ces systèmes passe par une phase d'apprentissage supervisée ou non, phase à l'issue de laquelle un système a appris à reconnaitre des objets, des formes, des textures, des pathologies, etc. Cette dernière phase s'appelle l'étape de classification. Comme nous le verrons dans ce chapitre, bon nombre de tels systèmes ont été sécurisés. Un enjeu cependant est la sécurisation d'approches fondées sur les réseaux de neurones ; réseaux qui sont à l'origine notamment de l'apprentissage profond (ou "deep Learning") et qui particulièrement d'actualité avec les données massives (ou "big data") et dont seule la phase de classification a été sécurisée aujourd'hui.

Dans ce chapitre, nous présentons un système d'apprentissage automatique fondé sur un perceptron multicouches, un type de réseau de neurones, que nous avons sécurisé sur la base du chiffrement homomorphe et de deux fournisseurs de Cloud indépendants, honnêtes mais curieux, sous la contrainte de zéro-communication entre l'utilisateur et le système externalisé. Toute la difficulté est de sécuriser un tel réseau de manière à ce que sa phase d'apprentissage converge alors qu'il prend en entrées des données chiffrées homomorphiquement tout en considérant que les paramètres du réseaux sont eux aussi confidentiels. En effet, ce sont ses paramètres qui constituent la valeur ajoutée du système d'apprentissage automatique et qui seront ensuite utilisés pour classer automatiquement de nouvelles données. À notre connaissance, cet objectif n'a pas encore été atteint dans la littérature.

Dans la première partie de ce chapitre, nous revenons sur les différentes méthodes d'apprentissage automatique avec un intérêt particulier pour les réseaux de neurone. Puis dans la deuxième partie, nous présenterons comment nous avons sécurisé le perceptron multicouche (MLP). Pour que la phase d'apprentissage converge alors que le traitement comme les données sont sécurisées, i.e. chiffrées de manière homomorphe, il convient de pouvoir sécuriser un certain nombre de traitements élémentaires qui ne sont pas linéaires (e.g. division, comparaison) et pour lesquels nous avons développé des outils cryptographiques spécifiques. Comme nous le verrons, notre approche fonctionne avec deux serveurs indépendants et n'a pas besoin de communiquer avec l'utilisateur pour entrainer le réseau ou classer les données. Nous avons également conduit l'analyse de sécurité de cette approche qui a été testée sur une base de données binaire pour apprendre la fonction binaire Et-Logique.

I Méthode d'apprentissage automatiques

Comme nous l'avons vu dans le chapitre 1 section I.2, on peut distinguer ces méthodes en fonction de si leur phase d'apprentissage est supervisée ou non. Un apprentissage supervisé

consiste à entrainer la méthode de machine learning sur un jeu de données d'apprentissage ; jeu pour lequel on connait la classe d'appartenance ou l'étiquette de chaque donnée du jeu. En ce qui nous concerne, notre intérêt porte sur les réseaux de neurones ou perceptron multicouches avec un apprentissage supervisé.

I.1 Machine learning par apprentissage supervisé

Mathématiquement, ce type de méthode suppose qu'il existe une fonction $G : X \longrightarrow Y$ qui met en lien l'espace d'entrée X (l'espace des données), avec l'espace de sortie Y (l'espace des étiquettes, labels ou classes). Même la forme de la fonction G n'est pas connue, une base de données, $BD = \{(x_n, y_n)/x \in X, y \in Y, n = \{0....N\}\}$, qui contient des échantillons d'entrées et de sorties de la fonction G est disponible. Cette base est une des composantes si ce n'est la composante la plus importante du schéma d'apprentissage car celle-ci est représentative de la fonction liant ses entrées à ses sorties. Plus cette base est grande et représentative de ces différents liens, meilleure sera la phase d'apprentissage de la fonction G. Cette phase a pour but d'estimer la fonction inconnue G à l'aide d'un ensemble d'hypothèses $H = \{h_m : X \longmapsto Y/m = \{0...M\}\}$. Ces hypothèses sont par exemple des fonctions linaires ou non-linéaires qui dépendent du choix de la méthode d'apprentissage et qui combinées permettent d'approcher la fonction G. Les nombres d'hypothèses et de données d'apprentissage sont des éléments clés dans la qualité de l'apprentissage. De ces nombres dépend le taux d'erreur ou de mauvaises classifications du système après l'apprentissage selon l'inégalité de Vapnik-Chervonenkis.

À partir des données d'entraînement et les fonctions de l'ensemble d'hypothèses, un algorithme d'apprentissage produit l'hypothèse finale $F : X \longrightarrow Y$. Cette fonction F doit être une bonne approximation de la fonction visée G. Dans le cadre du "Machine learning", les algorithmes les plus utilisés pour apprendre la fonction sont les suivants :

- **Machine à vecteurs de support (SVM "Support Vector Machine ")** : Les SVM sont des classifieurs qui reposent sur deux idées clés. Ils sont capables de traiter des problèmes de discrimination non-linéaire en reformulant le problème de classification comme un problème d'optimisation quadratique. La première idée clé est la notion de marge maximale. La marge est la distance entre la frontière de séparation et les échantillons les plus proches qui sont appelés vecteurs supports. La frontière de décision est choisie comme celle qui maximise la marge. La recherche de la frontière séparatrice optimale, à partir d'un ensemble d'apprentissage est faite en formulant le problème comme un problème d'optimisation quadratique, pour lequel il existe des algorithmes connus. La deuxième idée clé est de transformer l'espace de représentation des données d'entrées en un espace de dimension plus grande afin de gérer les cas où les données ne sont pas linéairement séparables. Ceci est réalisé grâce à une fonction noyau qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace.
- **Fonctions de base radiale ("Radial Basis Functions")** : Ce sont des fonctions dont la valeur ne dépend que de la distance par rapport au centre de la fonction. Elles ont la propriété suivante $\phi(x) = \phi(||x||)$. Les fonctions de base radiales sont typiquement utilisées pour construire une approximation de la fonction à apprendre de la forme

$$F(x) = \sum_{n=1}^{N} w_n \phi(\|x - x_n\|) \tag{4.1}$$

où la fonction d'approximation $F(x)$ est représentée comme une somme de N fonctions radiales de base, chacune associée à un centre x_n différent et pondérée par un coefficient w_n

approprié. Les poids w_n peuvent être estimés à l'aide de la méthode des moindres carrés en utilisant les échantillons de la base de données.

- Réseau de neurones (Neural Networks) : Ce sont des réseaux complexes d'unités élémentaires de calcul inter-connectées appelées neurones ou perceptrons qui permettent d'apprendre une fonction G à partir d'une base de données. Comme on le verra dans la sous-section suivante, il existe des algorithmes d'apprentissage pour trouver les paramètres optimaux du réseau et assurer que la fonction apprise est une bonne approximation.

I.2 Des réseaux de neurones à l'apprentissage sur des données massives

Les réseaux de neurones sont devenus un outil important dans de nombreux domaines (santé, militaires, multimédia et au grand publique) pour des fonctions telles que la reconnaissance d'objets, la classification d'images et la recherche par le contenu. La disponibilité plus de données et de puissance de calcul ont conduit aujourd'hui à l'émergence de l'apprentissage profond ("deep learning").

Les réseaux de neurones sont des structures composées de neurones ou perceptrons qui sont normalement organisés en couches. Dans cette section, nous revenons sur la structure mathématique d'un neurone, les paramètres qui définissent son comportement et son impact global sur le réseau. Nous aborderons ensuite les différentes architectures de réseaux dont la pertinence dépend de l'objectif du réseau.

I.3 Le perceptron : un neurone numérique

La structure mathématique d'un neurone ou d'un perceptron est donnée en Figure 4.1. Les entrées du neurone $\{x_i\}_{i=1...N}$ sont pondérées à l'aide de poids synaptiques $\{w_i\}_{i=1...N}$ puis sommées avec un biais b, c.-à-d. :

$$z = b + \sum_{i=1}^{N} x_i.w_i \tag{4.2}$$

De manière vectorielle, on peut écrire :

$$z = X^T.W \tag{4.3}$$

où $X = (x_0, x_1, \cdots, x_N)$ et $W = (w_0, w_1, \cdots, w_N)$, avec $x_0 = 1$ et $w_0 = b$.

Le résultat est ensuite placé en entrée d'une fonction d'activation, une fonction non-linéaire, dont le résultat constitue la sortie du neurone. L'intérêt de cette fonction d'activation est de pouvoir permettre l'approximation de fonction d'apprentissage G non-linéaires.

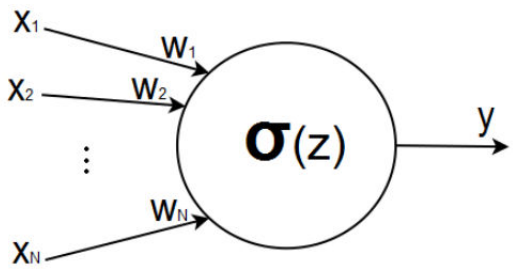


FIGURE 4.1 – La structure d'un neurone

Il existe différente fonction d'activation. Le choix de la fonction dépend de l'objectif applicatif. Parmi toutes ces fonctions, celles qui sont les plus couramment utilisées sont la fonction échelon $y_1(.)$, la fonction sigmoide $y_2(.)$ et la fonction ReLu $y_3(.)$ (Unité de rectification linéaires) :

$$y_1(z) = \begin{cases} 1 & \text{si} \quad z \geq 0 \\ 0 & \text{autrement} \end{cases} \tag{4.4}$$

$$y_2(z) = \frac{1}{1+e^{-z}} \tag{4.5}$$

$$y_3(z) = \begin{cases} z & \text{si} \quad z \geq 0 \\ 0 & \text{autrement} \end{cases} \tag{4.6}$$

Dans le cas d'une fonction d'activation ReLu, on peut assimiler un neurone à un classifieur qui sépare l'espace des données en entrées suivant l'hyperplan défini par les poids synaptiques et le biais (le biais évite que l'hyperplan passe par zéro).

I.4 Le perceptron multicouches "MLP"

Un neurone seul ne permet d'approcher que des fonctions très simples. Dans le cas de fonctions complexes, comme la classification d'images, il faut trouver une architecture appropriée qui associe plusieurs neurones. Toute la question est de trouver cette architecture. À noter également qu'en général, plus le nombre de neurones est important, plus la capacité d'apprentissage du réseau augmente.

Le perceptron multicouches (MLP) est une des architectures les plus utilisées, car une des plus efficaces dans de nombreux problèmes pratiques, qui organise les neurones en couche. Comme illustré en Figure 4.2, c'est une structure "feed-forward" dans laquelle les neurones ne sont connectés que dans un sens. Un réseau MLP possède des neurones organisés dans des couches distinctes. D'une manière générale, la couche d'entrée sert simplement à introduire les valeurs des variables d'entrées. Les neurones de la couche cachée et de la couche de sortie sont connectés chacun à toutes les neurones de la couche précédente.

Lors de l'exécution du réseau, les couches sont progressivement exécutées en ordre séquentiel (voir Figure 4.2). Chacun de ces neurones va calculer une valeur d'activation de la somme pondérée entre les sorties de la couche précédente et les poids synaptiques du neurone. Cette valeur d'activation est placée en entrée des neurones de la couche suivante. Dans le cas d'un MLP pour classer des données en N classes, chacune des N sorties du réseau donnera la probabilité d'appartenance d'une donnée placée en entrée du réseau à l'une des N classes (voir Figure 4.2 pour $N = 2$). Il est possible de représenter mathématiquement un MLP sous forme matricielle. Prenons pour exemple le MLP illustré en Figure 4.2.

- Puisque les couches sont entièrement connectées, leurs perceptrons respectifs ont le même nombre de poids, ces derniers peuvent donc être rangé dans une matrice $w^{(l)}$, l'indice l indique le numéro de la couche.
- On notera $z^{(l)}$ le vecteur qui regroupe les sorties de la couche l avant le passage par la fonction seuil (on dira avant l'activation) et $y^{(l)}$ le vecteur regroupant les sorties de la couche après l'activation.
- La sortie du système sera notée $y^{(max)}$ et l'entrée $y^{(0)}$.
- Notant $w_1^{(l)}, \cdots, w_{n_l}^{(l)}$ les vecteurs poids des perceptrons de la couche l (les lignes de $w^{(l)}$).
- f désignera la fonction d'activation, on a $y_i^{(l)} = f(z_i^{(l)})$ pour tout $i \in \{1, \cdots, n_l\}$, qu'on écrira sous forme matricielle : $y_l^{(l)} = f(z_l^{(l)}) = f(w^{(l)}.y^{(l-1)})$

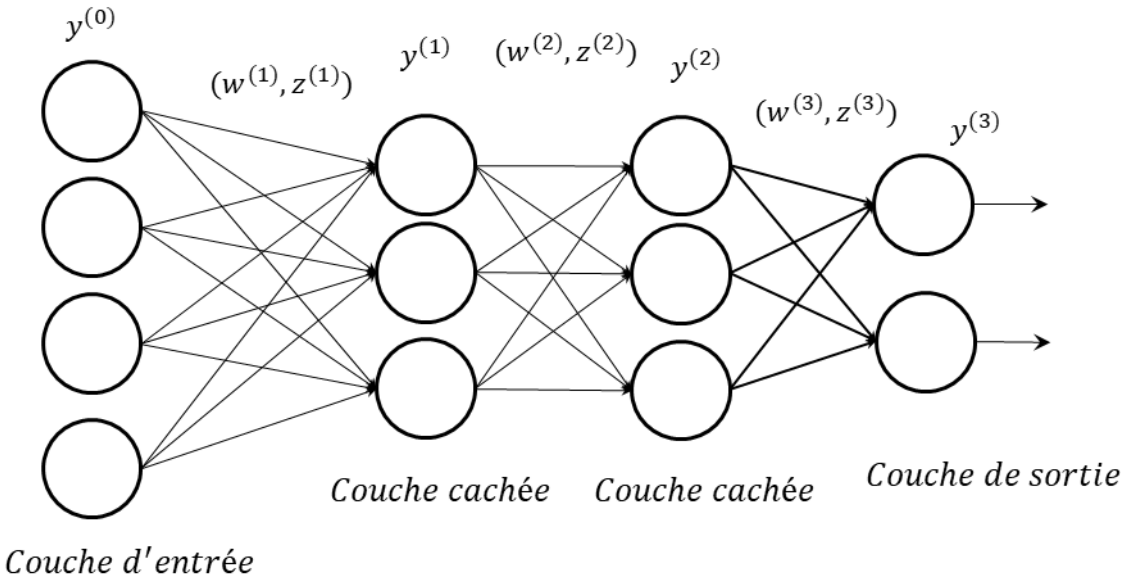


FIGURE 4.2 – Architecture d'un réseau de neurones multi-couches.

I.5 L'apprentissage d'un réseau MLP

L'objectif de cette phase d'apprentissage lorsqu'elle est supervisée est de déterminer les valeurs des poids synaptiques des neurones qui minimisent la distance entre les sorties du réseau et les étiquettes ou les labels associés aux données d'apprentissage placées en entrée du réseau.

Plusieurs mesures de distances peuvent être utilisées. Comme pour les fonctions d'activation, le choix de la mesure dépend de l'application visée. Parmi les distances les plus courantes, on utilise :

— Distance L^1

$$e(y,t) = |y - t| = \sum_{i=1}^{N} |y_i - t_i| \tag{4.7}$$

où le vecteur t est le label associées d'une entrée x, le vecteur y est la sortie du réseau pour la même entrée x et N est le nombre de composantes des vecteurs.

— Distance L^2

$$e(y,t) = \|y - t\|^2 = \sum_{i=1}^{N} (y_i - t_i)^2 \tag{4.8}$$

— Distance d'entropie croisée

$$e(y,t) = -\sum_{i=1}^{N} t_i.log(y_i) \tag{4.9}$$

L'apprentissage d'un réseau consiste donc à trouver la fonction $F(.)$ reliant les entrées et les sorties qui minimise cette fonction de coût e parmi l'ensemble d'hypothèses H , soit :

$$F(x) = arg \min_{F' \in H} \sum_{(x,t) \in BD} e(y,t) \tag{4.10}$$

Dans le cas des réseaux de neurones, il s'agit de trouver les poids synaptiques et les biais de tous les neurones. C'est un problème d'optimisation qui n'est pas simple à résoudre car il est

non-linéaire et non convexe. Une solution simple et largement utilisée est de s'appuyer sur une stratégie itérative : la méthode de descente de gradient (cf. Figure 4.3). L'algorithme de descente de gradient procède par améliorations successives pour minimiser la fonction d'erreur. À chaque étape, i.e. après avoir fait passer le jeu de données ou une donnée à travers le réseau, l'objectif est de déplacer dans la direction opposée le gradient de l'erreur (i.e. différence d'erreur entre deux étapes) par mise à jour des poids synaptiques et des biais, de manière à faire décroître la fonction de coût. La mise à jour des poids correspond à une étape dite de rétro-propagation du gradient. La rétro-propagation peut s'appliquer de différentes manières. Supposons que le jeu

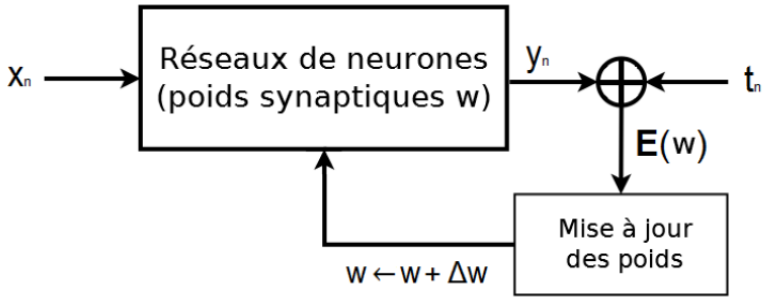


FIGURE 4.3 – Principe de la méthode de la descente de gradient et de la rétro-propagation de l'erreur pour un réseau de neurones multicouches

d'apprentissage possède N données. La technique de l'apprentissage total consiste à appliquer l'étape de rétro-propagation après avoir utilisé la totalité des données pour minimiser l'erreur :

$$e = \sum_{n=1}^{N} e(y_n, t_n) \tag{4.11}$$

L'apprentissage en ligne consiste lui à faire la mise à jour des poids synaptiques pour chaque donnée du jeu. L'apprentissage par lot, qui est le plus utilisé, divise la base d'apprentissage N en lots de M de données. Du fait que le nombre de données utilisées à chaque itération est réduit, le temps de calcul est réduit aussi. On parlera également de méthode de descente de gradient stochastique. C'est cette méthode que nous avons considérée dans ces travaux. Il est important dans ce cas de créer des lots qui sont représentatifs de la base de données de telle sorte que la direction du gradient de la fonction d'erreur soit, pour un lot de données, i.e. à chaque itération, semblable à celle obtenue sur l'ensemble de la base. Mathématiquement parlant :

$$\frac{\nabla e_{BD}}{\|\nabla e_{BD}\|} \cong \frac{\nabla e_{Lot}}{\|\nabla e_{Lot}\|} \tag{4.12}$$

avec $e_{BD} = \sum_{n=1}^{N} e(y_n, t_n)$ et $e_{Lot} = \sum_{m=1}^{M} e(y_m, t_m)$ Pour un perceptron multicouches, une fois l'erreur calculée pour un lot de données, l'étape de rétro-propagation du gradient a pour objectif de mettre à jour les poids de chaque neurone afin de minimiser l'erreur. On parle de rétro-propagation parce qu'on commence par calculer l'erreur à la sortie du réseau pour mettre à jour les neurones en allant vers l'entrée du réseau.

Reprenant les notations vectorielles d'un MLP vue en section I.4, la mise à jour des poids s'appuie sur la procédure suivante. Pour une entrée x (un vecteur) d'étiquette t (un autre vecteur), la fonction de coût est donnée par :

$$e_t = ||y^{(max)} - t||^2 \tag{4.13}$$

La mise à jour des poids de la dernière couches est donnée par la formule matricielles suivantes :

$$\frac{\partial e_t}{\partial w^{(max)}} =^t \delta^{z(max)} y^{(max-1)} \tag{4.14}$$

où $\delta^{z(max)} = (y^{(max)} - t) \bullet f'(y^{(max)})$, f est la fonction d'activation et l'opérateur $(\bullet)$ telle que $u \bullet v = (u_1 v_1, \cdots, u_n v_n)$ où u et v sont des vecteurs de même taille.

Pour mettre à jour les poids des neurones des couches précédentes, on profite de la connectivité entre les couches. Le calcul du gradient des couches cachées se fait par récurrence en exploitant la règle de chaine des dérivées partielles. La relation de récurrence est la suivante :

$$\frac{\partial e_t}{\partial w^{(l)}} =^t \delta^{z(l)} y^{(l-1)} \tag{4.15}$$

où $\delta^{z(l)} = f'(y^{(l)}) \bullet ({}^t\delta^{z(l+1)}.{}^t w^{(l+1)})$

En pratique, et dans le cas la descente de gradient stochastique, l'algorithme de rétro-propagation est donné par l' algorithme 1. L'algorithme de rétro-propagation est une version de l'algorithme 1 (voir algorithme 1).

Algorithm 1 Algorithme de rétro-propagation

1: Initialise la matrice de poids $w^{(l)}$ de chaque couche par des valeurs aléatoires
2: Pour $n \in \{1...N_{iter}\}$
3: Pour tout couple $(x = y^{(0)}, t)$ de la base d'apprentissage :
4: Calcule les sorties $y^{(1)}, ..., y^{(max)}$ des différentes couches
5: $\delta^{z(max)} = (y^{(max)} - t) \bullet f'(y^{(max)})$
6: $w^{(max)} = w^{(max)} - \eta.{}^t\delta^{z(max)} y^{(max-1)}$
7: pour $l \in \{(max-1), ..., 1)\}$
8: $\delta^{z(l)} = f'(y^{(l)}) \bullet ({}^t\delta^{z(l+1)}.{}^t w^{l+1})$
9: $w^{(l)} = w^{(l)} - \eta.{}^t\delta^{z(l)}.y^{(l-1)}$

I.5.1 Convergence et sur-apprentissage

Sur la base du jeu d'apprentissage, le MLP peut converger c'est-à-dire trouver les poids synaptiques et les biais des neurones qui minimisent la fonction d'erreur. Il existe cependant le danger du sur-apprentissage qui apparaît lorsque le comportement du réseau est trop adapté aux données d'entraînement. Plus clairement, le réseau classifie bien les données de la base d'entraînement mais est incapable de traiter correctement de nouvelles données qui lui sont inconnues. En fait, le réseau appris n'est pas généralisable au-delà de données d'entraînement. Des solutions à ce problème existent. Parmi elles, une stratégie consiste à augmenter les données d'apprentissage. Une autre passe par la réduction de la complexité du réseau. Cependant, du fait que souvent la base de données comporte un nombre limité d'échantillons et que les grands réseaux sont plus puissants que les petits réseaux, d'autres méthodes ont été proposées. Elles cherchent à détecter le sur-apprentissage et à en limiter l'impact. Les trois méthodes les plus répandues sont : l'arrêt prématuré, la régularisation et le momentum.

L'arrêt prématuré Pour détecter le sur-apprentissage, la base de données est divisée en deux sous-ensembles : un ensemble d'entraînement et un ensemble de validation. L'ensemble d'entraînement est utilisé par l'algorithme d'apprentissage pour faire la mise à jour de poids synaptiques du réseau. Les données de l'ensemble de validation ne sont pas utilisées pour l'apprentissage mais

pour vérifier le comportement du réseau avec des données qu'il ne connait pas. Si l'erreur de prédiction du réseau sur l'ensemble d'entraînement diminue alors que l'erreur sur la validation augmente de manière significative, il y a un sur-apprentissage. Le réseau améliore ses performances sur les données d'entraînement mais perd son pouvoir de prédiction pour de nouvelles données. L'apprentissage est arrêté dès qu'une divergence est observée entre les courbes d'erreurs d'apprentissage et de validation comme illustré en Figure 4.4.

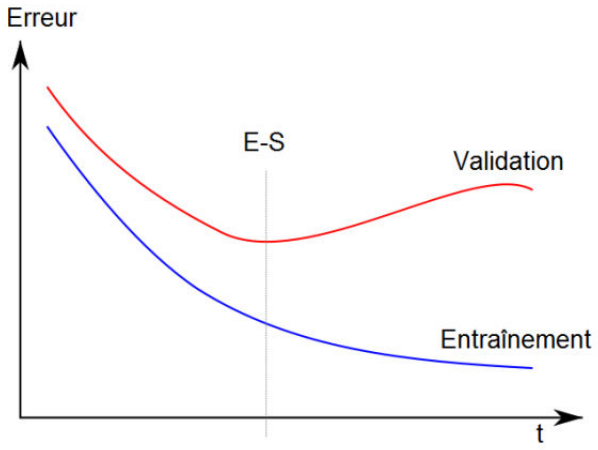


FIGURE 4.4 – Arrêt Prématuré : Courbes d'erreurs calculées sur les données d'apprentissage et de validation. E-S est l'abscisse de divergence des courbes ou encore le nombre d'itérations au-delà duquel l'apprentissage ne doit pas se poursuivre

Régularisation La régularisation consiste à ajouter un terme à la fonction de coût afin d'imposer une contrainte sur la forme de la fonction à apprendre. En temps normal, la fonction de coût à minimiser est donnée par :

$$e = \sum_{(x,t)\in BD} e(y,t) \tag{4.16}$$

On rajoute un terme de régularisation, $\rho(F(.))$, pondéré par un nombre réel, λ qui sert à déterminer le degré de régularisation.

$$e_R = e + \lambda\rho(F(.)) \tag{4.17}$$

Une des techniques de régularisation les plus utilisées est la dégradation des poids ("weight decay") où le terme de régularisation dépend de la magnitude des poids synaptiques qui relient les neurones entre eux.

$$e_R = e + \lambda \sum_i w_i^2$$

Intuitivement, cette régularisation fait que le réseau préfère apprendre des petits poids. Les grands poids ne seront autorisés que s'ils améliorent considérablement le premier membre de la fonction de coût. Cette régularisation cherche à trouver compromis entre la recherche de petits poids et la minimisation de la fonction de coût classique.

Momentum Le momentum est une technique qui aide l'algorithme d'apprentissage à sortir des minimums locaux de la fonction de coût. Il s'appuie sur l'ajout d'un terme d'inertie dans la mise à jour des poids afin de stabiliser la direction de déplacement au cours de l'apprentissage.

La direction de déplacement est donnée par le gradient de la fonction de coût et un terme qui correspond à la moyenne des gradients précédents pondérée par un facteur α. À chaque itération, le changement de poids conserve les informations des changements précédents. Cet effet de mémoire permet d'éviter les oscillations et accélère l'optimisation du réseau.

$$\Delta w^k = \alpha \Delta w^{k-1} - \eta \nabla e(w) \tag{4.18}$$

$$w^{k+1} = w^k - \Delta w^k \tag{4.19}$$

où $\nabla e(w)$ est le gradient de la fonction de coût et η est le taux d'apprentissage qui permet de contrôler la vitesse de l'apprentissage. Le critère d'arrêt consiste à interrompre la mise à jour si $\|\nabla e(w)\| < \epsilon$ avec ϵ une quantité réelle suffisamment petite (lorsque on s'approche du minimum de la fonction, le gradient devient nul)

Ses paramètres sont identifiées à la suite d'une phase d'apprentissage sur des jeux de données, dans le cas supervisé. Il faudra faire attention au sur-apprentissage.

II Méthodes d'apprentissage sécurisé

II.1 Réseaux de neurones sécurisé : L'existant

Comme nous l'avons vu en introduction de ce chapitre, il existe des versions sécurisées du MLP. Une première classe de méthodes s'appuient sur le partage de secret additif [150–152]. Cependant, ce dernier ne permet pas de manipuler les fonctions d'activation du fait de leur non-linéarité. Ces solutions contournent le problème en approchant ces fonctions par des polynômes ou des fonctions linéaires. Une conséquence immédiate est une perte de précision dans la classification ou, pire, une phase d'apprentissage qui ne converge pas. C'est pourquoi, ces stratégies n'externalisent pas la phase d'apprentissage qui est conduite chez l'utilisateur. Une seconde classe d'approches tire avantage du chiffrement homomorphe. À notre connaissance, toutes les approches proposées sécurisent la phase de classification d'un réseau de neurones. Par exemple, [153] propose trois classifieurs sur la base de trois cryptosystèmes homomorphes (Paillier, Goldwasser-Micali et BGV). Dans [154], les auteurs ont développé un réseau de neurones convolutif sécurisé. Cependant, à chaque fois, le réseau est entrainé dans le domaine en clair avant d'être externalisé. Dans [155], les auteurs montrent théoriquement qu'un réseau de neurones peut être entrainé avec des données chiffrées à l'aide des cryptosystèmes homomorphes, en suggérant d'approcher les fonctions d'activation par des polynômes, sans cependant démontrer expérimentalement les capacités de convergence de tels réseaux.

Sur cette base, nous nous sommes donc intéressés au problème de la sécurisation de la phase d'apprentissage d'un réseau perceptron multicouches (MLP) externalisé. La solution proposée s'appuie sur le chiffrement homomorphe et le calcul multipratite sécurisé à l'aide de deux fournisseurs de cloud indépendants. Le SMC est nécessaire notamment du fait de la non-linéarité des fonctions d'activation. Nous avons également fait le choix d'utiliser la fonction d'activation de l'unité linéaire rectifiée (ReLU) du fait de sa large utilisation, sa précision [156], sa rapidité [156] et sur le fait qu'elle peut être sécurisée avec un cryptosystème homomorphe et deux fournisseurs de clouds. Notre solution n'utilise pas d'approximation de la sortie du perceptron telle que proposée par les méthodes évoquées ci-dessus et, comme nous le verrons, peut converger.

Comme nous le verrons, notre MLP sécurisé peut être complètement externalisé et ne nécessite pas d'extra-communications avec l'utilisateur. Ce dernier doit simplement envoyer ses données chiffrées de façon homomorphe au serveur qui va entrainer le réseau de manière sécurisée puis classer les données avec ce réseau sans pouvoir déduire des informations sur les paramètres du MLP sécurisé, les données d'utilisateur ou le résultat de la classification.

II.2 Scénario d'externalisation du MLP

Le scénario d'usage que nous considérons dans ce travail est donné en Figure 4.5, où l'utilisateur externalise ses données chiffrées par le cryptosystème de Paillier afin d'entrainer un perceptron multicouches (MLP) situé sur le serveur P_1 d'un fournisseur de cloud ou pour classifier une requête. Dans notre système, le résultat de la classification ainsi que tous les paramètres du MLP doivent être inconnus du cloud, qui est considéré comme honnête mais curieux (voir chapitre 1 section III). Il tentera d'inférer des informations sur les données, les résultats de classification ainsi que les paramètres du MLP. Comme nous le verrons dans la suite, nous profiterons d'un deuxième serveur P_2 qui sera utilisé afin de mener certaines opérations non-linéaires nécessaires à l'entrainement ou la classification du MLP. En particulier, si le fonctionnement d'un neurone s'appuie sur des multiplications et des additions, l'utilisation de la fonction ReLu implique le calcul de la fonction max (cf. section I.3). La phase d'apprentissage fondée sur la descente de gradient, nécessite le calcul de dérivées, donc des opérations élémentaires comme la division, impossible à réaliser avec le chiffrement homomorphe. Dans la sous-section qui suit, nous présentons les fonctions sécurisées qui nous permettent d'implémenter un réseau MLP sécurisé sur des données chiffrées par le cryptosystème de Paillier.

Comme précédemment, nous faisons l'hypothèse que les fournisseurs de cloud sont semi-honnêtes, ou de manière équivalente, honnêtes mais curieux. Ils ne colludent pas. Pair ailleurs, nous ferons l'hypothèse que P_2 connait la clé secrète de l'utilisateur et que P_1 a les données d'entrainement et qu'il exécute et entraine le MLP sécurisé.

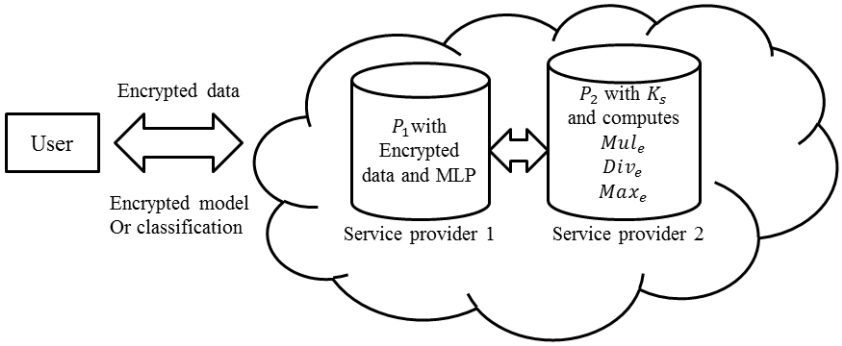


FIGURE 4.5 – Scénario d'usage sécurisé

II.3 Fonctions sécurisées

Comme indiqué dans le chapitre 2 section III.1, le cryptosystème de Paillier permet d'effectuer des opérations linéaires sur les données chiffrées. Pour calculer les opérateurs de multiplication, de division et de comparaison sur des données chiffrées, il est nécessaire de passer par deux serveurs P_1 et P_2.

- **Multiplication entre données chiffrées à l'aide du cryptosystème de Paillier opérateurs** $Mul_e(.;.)$ **et** $MUL_e(.;.)$

Considérons deux messages a et b ainsi que leurs chiffrés $E[a]$ et $E[b]$ détenus par P_1 et générés sur la base de la clé publique de l'utilisateur (K_p). Pour calculer $E[a \times b]$, nous proposons d'exploiter le " masquage" qui s'appuie sur un deuxième serveur P_2 qui réalisera l'opération de multiplication. L'idée est la suivante :

1. P_1 ajoute des nombres aléatoires aux messages a et b :

$$a' = E[a] \times E[r_a] = E[a + r_a] \tag{4.20}$$
$$b' = E[b] \times E[r_b] = E[b + r_b] \tag{4.21}$$

où r_a et r_b sont des nombres aléatoires choisis uniformément dans $\mathbb{Z}_{K_p}$ et connus uniquement de P_1. Ensuite, P_1 envoie a' et b' à P_2.

2. P_2 connaissant la clé secrète K_s de l'utilisateur déchiffre a', b' et multiplie le résultat

$$M = (a + r_a)(b + r_b) \tag{4.22}$$

envoie à P_1 : $E[M]$.

3. Pour obtenir $E[a \times b]$, P_1 doit simplement supprimer les facteurs aléatoires comme suit :

$$E[a \times b] = E[M] \times E[b]^{-r_a} \times E[a]^{-r_b} \times E[-r_a \times r_b] \tag{4.23}$$

Sur la base de cette procédure, P_1 accède à $E[a \times b]$ sans qu'aucune information sur a et b ne soient révélés à P_1 et P_2. Dans la suite, cette procédure sera notée par l'opérateur :

$$Mul_e(E[a], E[b]) = E[a \times b] \tag{4.24}$$

L'opérateur $MUL_e(.;.)$ a pour objectif de permettre une multiplication entre une matrice et un vecteur. Soit $A = \{a_{i,j}\}_{i=1\cdots m, j=1\cdots n}$ une matrice de taille $m \times n$ et $\{v_i\}_{i=1\cdots n}$ un vecteur de taille $n \times 1$. Dans le domaine en clair, le produit $w = Av = \{w_i\}_{i=1\cdots n}$ est un vecteur de taille m :

$$w_i = \sum_{j=1}^{n} a_{k,j} v_j \qquad i = 1 \cdots m \tag{4.25}$$

Considérant maintenant la matrice et le vecteur chiffrés élément par élément : $E[A]$) et $E[v]$. Leur produit dans le domaine chiffré est noté $E[w] = MUL_e(E[A], E[v])$ tel que

$$MUL_e(E[A], E[v]) = E[w_i] = \prod_{j=1}^{n} Mul_e(E[a_{k,j}], E[v_j]) \qquad i = 1, \cdots, m \tag{4.26}$$

- **Multiplication de deux vecteurs u et v élément par élément dans le domaine chiffré de Paillier : $MUL_e^{\bullet}(.;.)$**

Cette multiplication entre deux vecteur u et v de la même taille n a été définie dans le domaine en clair par l'opérateur $(\bullet)$ en section I.5. Son implémentation dans le domaine chiffré avec les chiffrés des vecteurs $E[u]$ et $E[v]$ est tel que

$$E[u \bullet v] = MUL_e^{\bullet}(E[u], E[v]) = (Mul_e(E[u_1], E[v_1]), \cdots, Mul_e(E[u_n], E[v_n])) \tag{4.27}$$

- **Division entre des données chiffrées par le cryptosystème de Paillier : $Div_e(.;.)$ et $DIV_e(.;.)$**

Différentes manières ont été proposées pour calculer la division dans le domaine chiffré à l'aide de deux serveurs [157]. Celle que nous utilisons fonctionne comme suit : Supposons que P_1 a un message chiffré $E[a]$ et qu'il veut diviser a par d qui peut être ou non chiffré. De son côté, P_2 connait la clé de déchiffrement. Le calcul $E[a/d]$ à partir de $E[a]$ et d est également basé sur le masquage :

1. P_1 choisit de manière aléatoire un nombre r dans $\mathbb{Z}_{K_p}$ et calcule $E[z] = E[a+r] = E[a]E[r]$. Il envoie $E[z]$ à P_2.
2. P_2 déchiffre $E[z]$, calcule $c = z/d$, et chiffre le résultat ($E[c]$) qu'il envoie à P_1
3. P_1 calcule $E[a/d]$ tel que $E[a/d] = E[c]$ x $E[-r/d]$.

A l'issue de cette procédure, on peut voir que P_1 et P_2 n'apprennent aucune information sur a.

De la même manière nous aurons besoin de calculer la division des composantes d'un vecteur par un scalaire dans le domaine chiffré. Pour ce faire nous définissons l'opérateur $DIV_e(.;.)$ tel que :

$$DIV_e(E[v], a) = (Div_e(E[v_1], a), \cdots, Div_e(E[v_n], a)) \tag{4.28}$$

- **Le calcul du max de données chiffrées par le cryptosystème de Paillier :** $Max_e(.;.)$

La fonction $Max(a, b)$ est un opérateur de base dans nombreuses applications, en particulier, elle est au coeur de la fonction d'activation ReLU (unité linéaire rectifiée). Différentes solutions ont été proposées afin de comparer en toute confiance les données chiffrées [1, 4, 158, 159]. La plupart d'entre elles sont fondées sur le masquage et deux serveurs indépendants. Cependant, à la fin du processus, le résultat de la comparaison est renvoyé en clair au serveur qui fait la requête de comparaison. Plus précisément, si P_1 demande à P_2 de comparer $E[a]$ et $E[b]$, P_1 saura si a est ou non supérieur à b. Une fuite d'information se produira si P_1 a une connaissance sur la valeur de a ou de b. Dans ce travail, nous proposons une alternative où le résultat de la fonction Max est envoyé à P_1 sous forme chiffrée. La procédure correspondante est la suivante :

1. P_1 sélectionne deux valeurs aléatoires r et $r^{'}$ de $\mathbb{Z}_{K_p}$ telles que $r^{'}$ est significativement plus petit que r (c'est-à-dire $r >> r^{'}$) et calcule

$$E[r(a-b) - r'] = (E[a]E[b]^{-1})^r \times E[r']^{-1} \tag{4.29}$$

   Il envoie le résultat à P_2.
2. P_2 déchiffre le message, compare les données et envoie un bit chiffré $E[i]$ à P_1 tel que i égale à 1 si $r(a-b) - r^{'} > 0$ ou à 0 sinon.
3. Pour obtenir le résultat de la fonction $Max(a, b)$ chiffré, P_1 calcule

$$\begin{aligned} Max_e(E[a], E[b]) &= E[max(a, b)] \\ &= Mul_e(E[a]E[b]^{-1}, E[i]) \times E[b] \end{aligned} \tag{4.30}$$

$$= E[i(a-b)+b] = \begin{cases} E[a] & if \quad i = 1 \\ E[b] & if \quad i = 0 \end{cases} \tag{4.31}$$

A l'issue de cette procédure P_1 accède ainsi à la version chiffrée de la valeur maximale entre deux nombres entiers. P_1 et P_2 n'obtiennent aucune information sur a et b ni sur le résultat de la fonction $Max(a, b)$. Cette procédure qui n'a jamais été proposée par ailleurs, et qui s'appuie sur la fonction Mul_e jouera un rôle majeur dans la sécurisation du MLP.

Soit deux vecteur chiffrés $E[v]$ et $E[u]$ et de même taille n. Le vecteur $MAX_e(E[u], E[v])$ est tel que

$$MAX_e(E[u], E[v]) = (Max_e(E[u_1], E[v_1]), \cdots, Max_e(E[u_n], E[v_n])) \tag{4.32}$$

Ces opérateurs nous servirons pour sécuriser un réseau MLP pendant la phase de "feed-forward" et de "rétro-propagation".

II.4 MLP sécurisé

La sécurisation d'un perceptron multicouches consiste à implémenter les phases de "feed-forward" et de "rétro-propagation" de manière sécurisée et sur des données chiffrées. Notre objectif est de pouvoir faire qu'un tiers, honnête mais curieux, exécute ces phases sur des données externalisées sécurisées sans qu'il ne découvre les paramètres du réseaux en cours d'apprentissage ou de classification, ou n'obtienne d'information sur les résultats de classification ou sur les données en entrées et les données des utilisateurs (i.e. clé privé). Pour atteindre cet objectif nous proposons d'utiliser les opérateurs sécurisé précédents qui s'appuient sur deux serveurs indépendants et des données chiffrées à l'aide du cryptosystème de Paillier.

Le MLP que nous nous proposons de sécuriser, tant dans sa phase d'apprentissage que de classification, s'appuie sur des neurones dont la fonction d'activation est la fonction ReLU et utilise comme fonction de coût l'erreur quadratique moyenne (EQM). Nous l'expérimenterons dans le cas de l'apprentissage de la fonction Et-Logique. Comme nous le verrons la phase d'apprentissage de ce réseau converge.

II.4.1 la phase de "feed-forward" sécurisé

Cette phase correspond pour un réseau appris à la phase de classification. C'est également la première phase de l'apprentissage à l'issue de laquelle sera calculée la fonction de coût à minimiser par la suite. Avant de décrire l'ensemble de la procédure sécurisée, intéressons-nous dans un premier temps à la sécurisation d'un seul perceptron (ou neurone).

Comme décrit dans la section I, un perceptron effectue une somme pondérée de ses entrées dont le résultat z est fourni à la fonction d'activation la sortie de laquelle constitue la réponse du neurone. Considérant le vecteur d'entrée X, nous avons

$$z = W.X = \sum_{i=1}^{n} x_i w_i$$

où $W = \{w_i\}_{i=1\cdots n}$ sont les poids synaptiques du perceptron. Utilisant la fonction d'activation ReLU, la réponse du perceptron est alors :

$$y = max(0, z) \tag{4.33}$$

Considérant la contrainte que tous les éléments du perceptron (i.e. ,$\{x_i\}_{i=1\cdots n}$, $\{w_i\}_{i=1\cdots n}$, z et y) sont confidentiels, ils sont chiffrés à l'aide du cryptosystème de Paillier. P_1 a seulement accès à $\{E[x_i]\}_{i=1\cdots n}$, $\{E[w_i]\}_{i=1\cdots n}$,$E[z]$ et $E[y]$. Du fait de la présence de multiplications et de calcul de la fonction max, P_1 interagira avec le serveur P_2 en utilisant les opérateurs sécurisés décris précédemment.

Deux autres contraintes importantes sont aussi à considérer. La première réside dans le fait que le cryptosystème de Paillier fonctionne que avec des entiers positifs dans $\mathbb{Z}_{K_p}$ (où $\mathbb{Z}_{K_p}$ est le domaine en clair). Toutes les données et les paramètres du neurone doivent être représentés par des nombres entiers. Pour répondre à cette contrainte, nous appliquons à l'entrée du réseau une opération de quantification couplée à un facteur d'expansion :

$$\Theta = [Q\theta] \tag{4.34}$$

où θ est un réel et Θ son équivalent en nombre entier, [] est l'opérateur d'arrondi et Q est un facteur d'expansion. Dans ce qui suit, et pour des questions de simplicité, nous considérerons travailler sur des entiers. Cette opération d'expansion et de quantification est implicite à l'entrée de chaque neurone. Nous reviendrons sur l'impact de cette opération sur la convergence d'un MLP dans la partie expérimentation (cf. section IV). La seconde contrainte est liée au fait que,

même si les paramètres de MLP sont des nombres entiers, leur traitement peut conduire à des valeurs négatives. Pour représenter ces valeurs dans $\mathbb{Z}_{K_p}$, nous subdivisons cet espace en deux parties. Les valeurs entières supérieures à $(K_p + 1)/2$ correspondront aux valeurs négatives et les autres aux valeurs positives.

La version sécurisée d'un perceptron exécutée par le serveur P_1 est la suivante :

1. P_1 calcule la somme pondérée chiffrée $E[z]$ telle que

$$E[z] = \prod_{i=0}^{n} Mul_e(E[x_i], E[w_i]) \tag{4.35}$$

2. La sortie du perceptron sécurisé $E[y]$ est obtenue en utilisant la version sécurisée de la fonction d'activation ReLU dans le domaine chiffré, c.-à-d. :

$$E[max(0, z)] = Max_e(E[0], E[z]) \tag{4.36}$$

qui est équivalent à $E[y] = E[max(0, z)]$.

Dans le cas du perceptron multicouches, où les neurones de deux couches successives sont entièrement connectés, la phase de classification ou de "feedforward" au niveau de la $l^{ème}$ couche du réseau est dans la représentation matricielle de la forme :

$$y^{(l)} = max(z^{(l)}, 0) = max(w^{(l)}.y^{(l-1)}, 0) \tag{4.37}$$

où $z^{(l)}$ et $y^{(l)}$ sont les vecteurs de sortie de la $l^{ème}$ couche avant et après la fonction d'activation ReLu. $w^{(l)}$ est la matrice de poids associée à la $l^{ème}$ couche. La version sécurisée de cette phase "feedforward " profite de nos opérateurs sécurisés MUL_e et MAX_e de la manière suivante :

$$E[y^{(l)}] = MAX_e(MUL_e(E[w^{(l)}], E[y^{(l-1)}]), E[0]) \tag{4.38}$$

La version sécurisée de la phase "feed-foward" d'un MLP de (max) couches est de la forme :

$$E[y^{(l)}] = MAX_e(MUL_e(E[w^{(l)}], E[y^{(l-1)}]), E[0]) \tag{4.39}$$

pour $l = 1...(max)$.

II.4.2 La phase de rétro-propagation sécurisée

Dans l'apprentissage, cette phase suit la phase de "feed-forward", qui utilise des données labellisées, et a pour objectif de mettre à jour les poids du réseau minimisant la fonction de coût.

Dans notre cas, la fonction de coût est l'erreur quadratique moyenne (EQM) entre les labels (t) des données d'entrées utilisées et les sorties du MLP obtenues lors la phase feed-forward ($y^{(max)}$ pour un MLP de (max) couches) :

$$e_t = MSE(y^{(max)}, t) = (y^{(max)} - t)^2 \tag{4.40}$$

Dans le domaine chiffré, l'EQM peut être calculée à l'aide de l'opérateur $Mul_e(.;.)$ comme suit :

1. P_1 qui détient $E[y^{(max)}]$ et $E[t]$, calcule $E[y^{(max)} - t] = E[y^{(max)}]E[t]^{-1}$
2. P_1 interagit avec P_2 pour calculer $E[e_t] = Mul_e(E[y^{(max)} - t], E[y^{(max)} - t])$

Une fois la fonction de coût calculée se pose la question de l'étape de retro-propagation qui doit être sécurisée.

Cette étape prévois le calcul de dérivée et le passage d'une couche à une autre. On commence par propager l'erreur de la sortie à travers les couches précédentes comme on a expliqué en section I.5.

La version sécurisée de l'algorithme de rétro-propagation est représentée dans l'algorithme 2

Algorithm 2 Algorithme de rétro propagation sécurisé

1: Initialise la matrice de poids $E[w^{(l)}]$ de chaque couche par des valeurs aléatoires
2: Pour $n \in \{1...N_{iter}\}$
3: Pour tout couple $(E[x] = [y^{(0)}], E[t])$ de la base d'apprentissage :
4: Calcule les sorties $E[y^{(1)}], ..., E[y^{(max)}]$ des différentes couches
5: $\delta^{z(max)} = E[(y^{(max)} - t) \bullet f'(y^{(max)})] = MUL_e^{\bullet}(E[y^{(max)}]E[t]^{-1}, E[f'(y^{(max)})])$
6: $E[w^{(max)}] = E[w^{(max)}]DIV_e(E[MUL_e({}^t\delta^{z(max)}, y^{(max-1)})], \eta^{-1})$,
7: pour $l \in \{(max-1), ..., 1)\}$
8: $E[\delta^{z(l)}] = MUL_e^{\bullet}(E[f'(y^{(l)})], (MUL_e(E[{}^t\delta^{z(l+1)}], E[{}^t w^{l+1}]))$
9: $E[w^{(l)}] = E[w^{(l)}]DIV_e(E[MUL_e({}^t\delta^{z(l)}, y^{(l-1)})], \eta^{-1})$,

La dérivé de la fonction *Relu* est la fonction échelon :

$$f'(x) = 1_{x>0} = \begin{cases} 1 & si \quad x > 0 \\ 0 & sinon \end{cases} \tag{4.41}$$

Cette fonction peut être calculée dans le domaine chiffré en utilisant l'opérateur $Max_e(,)$ car c'est une fonction fondée sur la comparaison.

III Discussion et analyse de sécurité

III.1 Gestion des "overflows"

Comme nous l'avons vu, le système de chiffrement de Paillier travaille avec des entiers dans l'ensemble $\{0, \cdots, K_p - 1\}$, où K_p est la clé publique de l'utilisateur. Dans le même temps, nous avons choisi de représenter les nombres en clair positifs dans l'ensemble $\{0, \cdots, (K_p - 1)/2\}$ et les négatifs dans $\{(K_p + 1)/2, \cdots, K_p - 1\}$.

Pour pouvoir exprimer un MLP avec des entiers, nous utilisons une opération d'expansion couplée avec une quantification. Pour une valeur réelle $x \in \mathbb{R}$, sa version entière est telle que :

$$x^* = [xQ] \simeq xQ \tag{4.42}$$

où Q est le facteur d'expansion et [.] est l'opérateur d'arrondi. Pour minimiser la perte d'information qui se traduira comme une baisse de précision dans les calculs, il faut Q suffisamment grand. Idéalement, nous souhaitons avoir $x^* = [xQ] = xQ$. Cependant en pratique nous obtenons $[xQ] \simeq xQ$. Néanmoins, sous cette hypothèse la première propriété d'homomorphie de Paillier est toujours vérifiée, c'est-à-dire :

$$x_1^* + x_2^* = [x_1Q] + [x_2Q] \simeq x_1Q + x_2Q = (x_1 + x_2)Q \tag{4.43}$$

Cela permet au cloud d'effectuer un nombre arbitraire de sommes à partir des messages chiffrés. Par contre, la deuxième propriété de Paillier qui concerne la multiplication pose un problème. En effet :

$$x_1^* \times x_2^* = [x_1Q] \times [x_2Q] \simeq x_1x_2Q^2 \tag{4.44}$$

La présence du facteur Q^2 a une conséquence importante : la taille du message chiffré augmente de façon exponentielle avec le nombre de multiplications. Cela peut conduire à des « overflows », c'est-à-dire que le résultat du produit est plus grand que $(K_p + 1)/2$. Des nombres qui devraient positifs vont apparaître négatifs. Ce résultat est en particulier vérifié dans la phase d'apprentissage du MLP, qui engendre de nombreuses opérations successives. Cela sera aussi le cas d'un MLP avec de nombreuses couches.

Pour résoudre ce problème, deux possibilités s'offrent à nous. La première consiste à choisir une clé publique très grande, ce qui aura pour effet en contre partie d'augmenter les temps de calcul. La seconde est celle que nous avons adoptée, elle consiste à faire des divisions après chaque multiplication par le facteur d'expansion Q. Par conséquent, quelque soit le nombre d'opérations n appliquées au chiffré, celui-ci gardera toujours un facteur d'expansion Q au lieu de Q^{2n}. On verra dans la section des résultats expérimentaux que cette solution contribue à la convergence du réseau MLP sécurisé.

III.2 Analyse de sécurité

Cette analyse considère le modèle d'attaquant semi-honnête. Du fait que l'apprentissage d'un MLP est composée de deux phases, la phase de "feed-forward" et la phase de rétro-propagation, la sécurité d'un MLP repose sur la sécurité de ces deux phases. À noter également que l'étape de classification de nouvelles données fonctionne de la même manière que la phase "feed-forward", les poids du MLP appris, nous n'analyserons pas sa sécurité.

Dans notre système, toutes les données, paramètres du modèle MLP compris, sont chiffrées par le cryptosystème Paillier dont la sécurité est prouvée dans [86] sous le modèle semi-honnête. La sécurité des phases de feed-forward et de rétro-propagation repose sur la sécurité des opérations élémentaires $Mul_e(.)$, $Div_e(.)$ et $Max_e(.)$. Nous verrons que si ces dernières sont sûres alors l'ensemble du système est sécurisé.

- **La sécurité de** Mul_e - Comme défini en section II.4, $Mul_e(E[a], E[b]) = E[a \times b]$ repose sur une opération de masquage de données et des interactions entre deux serveurs P_1 et P_2. Pour ce faire,P_1 génère deux valeurs aléatoires r_a et r_b de $\mathbb{Z}_{K_p}$ et calcule $E[a + r_a]$ et $E[b+r_b]$ qui les envoie à P_2. P_2 déchiffre ces derniers pour obtenir $A = a+r_a$ et $B = b+r_b$. Puisque r_a et r_b sont choisis aléatoirement dans $\mathbb{Z}_{K_p}$ et qu'ils sont seulement connus de P_1, A et B sont uniformément répartis dans $\mathbb{Z}_{K_p}$ du point de vue de P_2. Par conséquent, A et B ne fuitent pas d'information concernant a et b à P_2.
- **La sécurité de** $Div_e(.;.)$ - elle a été entièrement prouvée par Thijs Veugen dans [157] dans le cadre du modèle semi-honnête.
- **La sécurité de** $Max_e(.)$ - Ici, P_1 possède le couple $(E[a], E[b])$ et veut calculer $E[max(a, b)]$. Comme décrit en section II.3, P_1 calcule $E[r(a - b) - r']$ où r et r' sont choisis de manière aléatoires dans $\mathbb{Z}_{K_p}$ et tels que $r >> r'$. De son côté, P_2 déchiffre cette valeur dont il ne peut déduire aucune information sur a et b ou leur différence car il ne connaît pas r et r'. Dans notre schéma, pour éviter que P_1 ne connaisse le résultat de la comparaison (i.e. la valeur $max(a, b)$), P_2 renvoie le chiffré de la valeur d'un bit i qui vaut 1, si $r(a-b)-r > 0$, ou 0, sinon. A la réception P_1 calcule $E[i(a - b) + b] = E[max(a, b)]$. Ici P_1 et P_2 n'ont aucune idée sur a et b ou $max(a, b)$.

L'ensemble des calculs impliqués dans les fonctions du MLP en mode classification ou d'apprentissage (e.g. calcul de l'EQM, de la dérivée de l'erreur) sont basés sur des données chiffrées. Également, étant donné que tous les opérateurs précédents impliqués dans les phases « feed-forward » et de rétro-propagation sont sécurisés et qu'ils produisent eux aussi des données chiffrées, selon le théorème séquentiel de composition [160], notre approche MLP est également

x_1	x_2	t
0.0285	0.5696	0
0.7846	0.9251	1
0.2145	0.4875	0

TABLE 4.1 – Extrait de la base d'entraînement pour la fonction ET-Logique. x_1 et x_2 sont des nombres réels et les entrées de la fonction. t correspond au label et y est la sortie de la fonction appliquée au couple (x_1, x_2).

sécurisée sous le modèle semi-honnête. En conséquence, si P_1 et P_2 ne colludent pas (P_2 connait la clé privé K_s), aucune information liée aux données des utilisateurs ou au modèle MLP n'est divulguée. Rappelons que la collusion est une hypothèse interdite selon le modèle d'attaquant semi-honnête.

IV Résultats expérimentaux

Nous avons expérimenté l'approche précédente dans le cas d'un perceptron multicouches visant à modéliser la fonction ET-Logique sur un mode d'apprentissage supervisé. Cette fonction prend en entrée deux nombres réels x_1 et x_2 dans l'intervalle $[0, 1]$ et sa sortie est une valeur binaire telle que :

$$y = [x_1]ET[x_2] \tag{4.45}$$

où [] est l'opérateur d'arrondi qui retourne l'entier le plus proche.

IV.1 La base de données et l'architecture du réseau

Les données d'entraînement consistent en un ensemble de 10000 lignes de trois colonnes chacune, chaque ligne représentant un exemple d'entrainement. Les deux premières colonnes contiennent un couple de deux valeurs réelles (x_1, x_2) comprises dans l'intervalle $[0, 1]$ et la troisième le label t, une valeur binaire, qui correspond au résultat de la fonction ET-Logique eq. (4.45) appliquée au couple. La Table 4.1 donne trois échantillons de cette base de données d'entraînement. Comme le montre la Figure 4.6, l'architecture proposée pour le réseau MLP est constituée d'une couche d'entrée comportant deux neurones, de deux couches cachées avec deux neurones et d'une couche de sortie avec un neurone. Le réseau a été produit en utilisant comme fonction d'activation la fonction ReLU et la méthode de descente de gradient stochastique. L'entrainement de notre MLP sécurisé a été effectué avec un apprentissage par lot de taille 10, un taux d'apprentissage de 0,005 et 100 itérations (voir section I.4). À noter que dans le domaine en clair, sur un ensemble de tests composé de 2000 exemples, la précision maximale trouvée à l'aide de la fonction ReLU est de $98,3\%$.

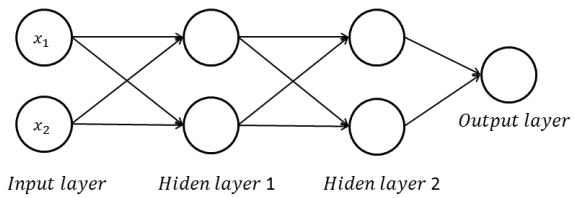


FIGURE 4.6 – Architecture du MLP proposé pour apprendre la fonction logique "ET"

Facteur d'expansion Q	10^3	10^4	10^5	10^6
Convergence & Précision maximale	non	oui/ 72%	oui/ 83%	oui/ $91,1\%$

TABLE 4.2 – La convergence et la précision du réseau MLP sécurisé en fonction du facteur d'expansion

FIGURE 4.7 – La précision du MLP en fonction du facteur d'expansion

IV.2 La performance du MLP sécurisé

Les performances de notre MLP sécurisé, qui s'expriment en termes de précision de classification et aussi de convergence, dépendent de plusieurs paramètres : le taux d'apprentissage et les paramètres de la fonction d'expansion.

- **Impact de le fonction d'expansion** Comme évoqué dans la section III.1, les nombres réels ne peuvent pas être chiffrés directement à l'aide du cryptosystème de Paillier. Pour ce faire, nous utilisons un facteur d'expansion sur les données d'entrée pour convertir les réels en nombres entiers (cf. section III.1 eq. (4.42)). Appliquée aux données d'entrées du MLP (i.e., les couples de valeurs (x_1, x_2), cette fonction expansion consiste en une simple multiplication par un facteur d'expansion Q, une puissance de 10, suivie de l'arrondi à l'entier plus proche. Ce facteur d'expansion a un impact direct sur la convergence du MLP sécurisé. Nous donnons dans la table 4.2, la précision et convergence de ce dernier pour différentes valeurs du facteur d'expansion Q dans l'intervalle $[10^3, 10^6]$. On peut voir que la précision, et donc la convergence du MLP, augmente avec Q jusqu'à atteindre les meilleurs performances pour (précision de $91,1\%$). Cependant, continuer à augmenter Q peut conduire à l'instabilité du système. Pour $Q > 10^7$ le système ne converge pas. Un tel résultat peut s'expliquer par la valeur des nombres alors à manipuler tant pour les données en entrées que pour les poids synaptiques du réseau, dans la phase de rétro-propagation. Pour généraliser notre MLP à d'autres applications, il conviendra de trouver un compromis entre précision et facteur d'expansion. À noter également, et comme évoqué en section III.1, notre MLP sécurisé applique après chaque multiplication une division par le facteur d'expansion Q cela pour éviter l'introduction « d'overflow », i.e. des résultats de valeurs supérieures à $(K_p + 1)/2$. Même si la division utilisée n'est pas équivalente à la division en clair (on travaille avec des nombres entiers), notre MLP converge avec une précision maximale de 91.1%. Sans cette division, nous avons vérifié que le MLP ne converge jamais. Il suffit de 3 à 4 itérations pour que les « overflows » apparaissent.
- **Impact du taux d'apprentissage** Le taux d'apprentissage, qui influe sur le pas de modification dans la mise à jour des poids (i.e. voir principes de la descente de gradient cf. section I.5) dans la phase de rétro-propagation, joue aussi un rôle critique dans la convergence du réseau. Nous avons testé plusieurs valeurs de taux d'apprentissage η pour un facteur d'expansion $Q = 10^6$. Nous donnons dans le tableau 4.3, la variation de la précision de notre MLP sécurisé pour des taux dans l'intervalle $[10^{-12}; 10^{-4}]$. Il est possible de voir que le MLP converge pour des petites valeurs. Pour des grandes valeurs de η le sys-

η	10^{-12}	10^{-10}	10^{-8}	10^{-6}	10^{-4}
Convergence	oui	oui	oui	non	non

TABLE 4.3 – La convergence de notre MLP sécurisé en fonction du taux d'apprentissage

tème diverge. À noter par ailleurs, que le fait de choisir des poids synaptiques initiaux de grandes valeurs est plus robuste. En effet, des petites valeurs couplées à taux d'apprentissage élevé n'a pas de sens. En conséquence, nous conseillons de prendre des poids initiaux de manière uniformément répartie dans l'intervalle $[10^{-5}, 10^{5}]$, et un taux d'apprentissage de valeur de $\eta = 10^{-10}$.

- **Complexité du MLP sécurisé** Pour conclure, et comme le montrent les tableaux 4.3 et 4.2, la précision de notre MLP sécurisé est toujours inférieure mais proche de celle du MLP appris en clair de 91.3%. Quoiqu'il en soit, ces premiers travaux, bien que modestes, montrent qu'il est possible de déployer la phase d'apprentissage d'un MLP de manière externalisée

V Conclusion

Dans ce chapitre, nous avons vu comment on peut sécuriser un réseau de neurones tant dans sa phase d'apprentissage que dans sa phase de classification. Avec notre approche, la confidentialité des paramètres du réseau comme les données de l'utilisateur et les réponses du réseau est assurée. Pour ce faire, nous tirons avantage du chiffrement homomorphe et deux fournisseurs de cloud indépendants et semi-honnêtes. Nous avons une nouvelle version sécurisée de la fonction $Max(.,.)$. Contrairement aux solutions actuelles, pour laquelle le résultat de notre fonction est chiffré, comme ses entrées.

Nous avons montré que notre MLP sécurisé converge notamment grâce à la gestion des « overflows » dans la fonction d'expansion. Si cette fonction permet de transformer des nombres réels en nombre entier, et de conduire les phases d'apprentissage et de classification de manière équivalente, elle est source d'augmentation de la taille des données. Pour résoudre ce problème et assurer la convergence du réseau, nous proposons d'ajouter après chaque multiplication une division par le facteur d'expansion Q. Il conviendra cependant de trouver un compromis entre facteur expansion et précision de classification en fonction du contexte applicatif.

Pour aller au-delà de ces premiers résultats, l'implémentation de cette solution sur les images nécessite le développement de cryptosystèmes homomorphes rapides, car en termes de complexité de calcul la phase d'apprentissage d'un MLP peut prendre plusieurs jours dans le domaine en clair. Les problèmes que nous avons rencontrés quant à la manipulation de nombre entiers, montre également l'intérêt pour des cryptosystèmes fonctionnant avec des nombres réels. Il n'y aurait plus alors de pertes d'information liées à la fonction d'expansion.

Pouvoir exploiter ce type d'approches de manière externalisée sur la base de la base de données mutualisées, soulève la question de comment partager les données chiffrées externalisées entre plusieurs utilisateurs. En effet, les systèmes de recherche par le contenu sécurisé et d'apprentissage automatique sécurisé que nous avons proposés doivent pouvoir fonctionner directement sur les dossiers des patients externalisés par plusieurs médecin. C'est un enjeu sur lequel nous nous sommes penchés et qui fait l'objet du chapitre suivant.

Chapitre 5

Partage sécurisé et intégrité des données sécurisées

Comme nous avons pu le voir, la réutilisation de données externalisées soulève la question du partage ou de l'accès à des données chiffrées avec des clés de chiffrement différentes et aussi le besoin de tracer les données ou de s'assurer de leur fiabilité (cf. Chapitre 1). En effet, les solutions d'aide au diagnostic fondées sur des méthodes sécurisées de CBIR ou de machine learning externalisées (cf. Chapitres 2, 3 et 4) sont efficaces à la condition d'accéder de manière exhaustives aux différentes formes de pathologies ; exhaustivité que l'on trouve dans l'ensemble des dossiers des patients et dont l'accès est régulé par la loi. Rappelons que seuls les professionnels de santé en charge d'un patient ont le droit d'accéder aux données de ce patient. S'ils externalisent ces données, ils les chiffrent avec leurs clés publiques. Une solution serait d'unifier la clé de chiffrement pour l'ensemble des utilisateurs. Cela pose cependant le problème de la confidentialité des données, car chaque médecin pourrait accéder aux données de tous les patients. Une autre solution est, quand cela est nécessaire, que l'utilisateur rapatrie les données qu'il a externalisées pour les ré-externaliser chiffrées avec la clé publique du destinataire. Cette approche de sens lorsqu'il s'agit de traiter des données massives. L'utilisateur aurait besoin d'une forte puissance de calcul. La dernière solution qui nous intéresse dans ce chapitre, est le partage de données externalisées directement au sein du cloud. L'objectif est de permettre au cloud de convertir un message chiffré avec la clé d'un utilisateur en un message chiffré par la clé d'un autre utilisateur.

Les mécanismes cryptographiques qui permettent le partage de données chiffrées sont identifiés sous le terme « Proxy Re-Encryption (PRE) » (ou « re-chiffrement par proxy »). Comme nous le verrons, cette notion est une généralisation du chiffrement à clé publique. Les algorithmes de chiffrement et de déchiffrement sont modifiés de manière à permettre le changement de clé. Les clés de chiffrement/déchiffrement ne sont pas modifiées mais le récepteur a une information supplémentaire (clé de déchiffrement) qui lui permet de déchiffrer le message partagé par l'émetteur. Des solutions ont été proposées suivant les mêmes principes pour le chiffrement homomorphe. Dans ce chapitre, nous proposons une alternative qui ne nécessite pas de modifier intrinsèquement les algorithmes de chiffrement et de déchiffrement, d'utiliser une autre clé en plus des clés publiques/privées des utilisateurs tout en permettant le post-traitement des données sur le cloud.

En ce qui concerne les problèmes de traçabilité et de l'intégrité des données externalisées, nous proposons d'utiliser le tatouage de données. Cette technologie, lorsqu'elle est appliquée à l'image, permet de dissimuler un message par modifications imperceptibles des niveaux de l'image. Tel que définit le message est accessible indépendamment du format de stockage de l'image (le message est caché dans les niveaux de gris) et peut servir différents objectifs de sécurité comme le contrôle d'intégrité de l'image ou de la tracer en tatouant les identifiants du destinataire et de l'émetteur. C'est une protection *a posteriori*, la donnée est accessible mais

maintenue protégée par le message caché, contrairement au chiffrement qui est une protection *a priori* (la donnée est protégée tant qu'elle est chiffrée). Dans notre contexte, nous nous sommes intéressés au tatouage d'images chiffrées pour à assurer des services d'intégrité et de traçabilité que les images soient ou non chiffrées.

Ce chapitre s'organise en 2 parties. Dans la première, nous rappelons la définition et les propriétés d'un schéma PRE classique avant de présenter un nouveau concept de partage de données chiffrées dont l'originalité repose sur la génération de séquences aléatoires chiffrées et le calcul de différences entre données chiffrées que nous avons proposé dans le chapitre 2 section III.1. En plus de l'analyse de sécurité, nous montrons comment généraliser ce concept à plusieurs algorithmes de chiffrement homomorphes (« partial », « somewhat » et « fully »). Notre schéma a été implémenté dans le cas du partage d'images non compressées stockées dans le cloud, et offre des propriétés « unidirectionnalité » et de résistance à la collusion, des propriétés essentielles à un schéma de PRE.

La deuxième partie de ce chapitre porte sur un algorithme de tatouage dont le principe permet l'insertion de certains attributs de sécurité dans une image chiffrée ; attributs disponibles dans les deux domaines chiffrés et spatiaux (i.e. en clair) pour vérifier la fiabilité de l'image. Par rapport aux autres méthodes, notre approche chiffre entièrement l'image ; et les opérations de tatouage et de chiffrement/déchiffrement sont indépendantes. Plus clairement, les processus d'insertion et d'extraction des messages ne nécessitent pas la connaissance de la clé de déchiffrement.

I Proxy re-encryption (PRE)

Dans cette section, nous revenons sur la définition et les propriétés d'un schéma PRE que l'on trouve classiquement dans la littérature avant d'introduire notre schéma de partage de données chiffrées qui s'en distingue notamment par l'absence du besoin de la clé de re-chiffrement.

I.1 Définition et propriétés d'un schéma PRE

La notion de PRE a été introduite pour la première fois par Blaze *et al.* [161] en 1998, mais n'a été définie qu'en 2005 par Ateniese et Hamburger [162]. Cette définition est précisée par un ensemble de fonctions et de propriétés.

I.1.1 Définition

Le partage de données chiffrées se réfère généralement au proxy re-encryption (PRE), où Alice (le délégateur ou l'émetteur) veut partager avec Bob (le délégué ou le destinataire) certaines données chiffrées qu'elle a précédemment externalisées dans le Cloud (le Proxy). Lorsque le chiffrement asymétrique est utilisé, l'objectif du proxy est de re-chiffrer le texte d'Alice, chiffré avec sa clé publique, dans un texte chiffré qui peut être déchiffré avec la clé privée de Bob.

Classiquement et pour ce faire, Alice (A) génère une clé de re-chiffrement $rk_{A->B}$ qui sera utilisée par le proxy pour convertir un texte chiffré c_A sous la clé publique pk_A de Alice en un autre texte chiffré c_B sous la clé publique pk_B de Bob (B). c_A et c_B sont les chiffrés du même message m. Bob sera capable d'obtenir le message en clair m avec sa clé privée sk_B. Lors de l'exécution d'un schéma PRE sécurisé, un attaquant (par exemple, le proxy) ne doit pas pouvoir déduire des informations sur m ou les clés privées (sk_A ou sk_B).

En prenant la définition proposée dans [162], un PRE est définie par un ensemble de cinq fonctions :

— KeyGen (k) : Algorithme de génération de clé - il prend en entrée le paramètre de sécurité k, et retourne en sortie une paire de clés publique/privé (pk, sk).

— ReKey (pk_A, sk_A, pk_B, sk_B) : Algorithme de génération de la clé de re-chiffrement - il prend en entrée les clés de l'émetteur et parfois, la clé privée du destinataire, il retourne en sortie une clé de re-chiffrement $rk_{A->B}$. Cette fonction est exécutée par Alice (émetteur) $rk_{A->B}$

— Encrypt (m, pk) : Algorithme de chiffrement - cette fonction permet de chiffrer le message m par la clé publique pk pour générer un message chiffré c.

— ReEncrypt($c_A, rk_{A->B}$) : Algorithme de re-chiffrement - qui prend en entrée le message chiffré de Alice c_A et la clé de re-chiffrement $rk_{A->B}$. Cette fonction est exécutée par le Proxy pour transformer le message chiffré c_A en c_B via la clé de re-chiffrement.

— Decrypt(c, sk) : Algorithme de déchiffrement : Cette fonction prend en entrée un message chiffré c et une clé secrète sk, elle retourne un message en clair m

Afin de partager des données, ces fonctions sont utilisées comme suit :

$$Decrypt(sk_A, Encrypt(pk_A, m)) = m \tag{5.1}$$

$$Decrypt(sk_B, ReEncrypt(ReKey(pk_A, sk_A, pk_B, sk_B), Encrypt(pk_A, m))) = m. \tag{5.2}$$

Comme on peut le voir, un schéma PRE est dérivé d'un algorithme de chiffrement à clé publique dont il possède deux fonctions supplémentaires : ReEncrypt et ReKey.

I.1.2 Propriétés

Il est attendu qu'un schéma PRE possède les propriétés suivantes [163]

— Propriété Unidirectionnelle ou Bidirectionnelle - Un PRE est considéré comme unidirectionnel si le proxy ne peut convertir que le message chiffré du délégateur en un message chiffré sous la clé publique du délégué mais pas l'inverse. Plus clairement, Si Alice transmet un fichier à Bob via le cloud, le cloud n'est pas capable de transmettre un fichier de Bob à Alice bien qu'il connaisse la clé de re-chiffrement. Un système PRE bidirectionnel permet quant lui au proxy équipé de la clé de re-chiffrement de transformer non seulement le message chiffré du délégateur en un message chiffré sous la clé du délégué, mais aussi l'inverse. En fait, le cloud peut transmettre des fichiers d'Alice à Bob et aussi des fichiers de Bob à Alice avec la même clé de re-chiffrement. Une différence notable entre un système PRE unidirectionnel et un système bidirectionnel repose sur l'utilisation ou non de la clé secrète du délégué dans l'algorithme de génération de la clé de re-chiffrement (ReKey). Dans un système unidirectionnel, le type de système est le plus récent, seule la clé privée sk_A du délégateur est impliquée pour générer la clé de re-chiffrement $rk_{A->B}$. Dans un système bidirectionnel, le délégateur (Alice) et le délégué (Bob) fournissent leurs clés secrètes sk_A et sk_B pour générer $rk_{A->B}$.

— Propriété « multi-usage/ single-use » - Dans un système PRE à usage multiples, les messages chiffrés générés par l'algorithme Encrypt ou l'algorithme ReEncrypt peuvent être pris comme entrée dans ReEncrypt pour être re-chiffré une autre fois. En revanche, dans un schéma PRE à usage unique, le message chiffré obtenu ne peut pas être re-chiffrer après l'opération de re-chiffrement par ReEncrypt. Une conséquence est que l'on ne peut pas partager un fichier déjà partagé.

— Propriété de transparence - Un schéma PRE est transparent s'il est impossible pour le délégué de distinguer entre un chiffré qui résulte du chiffrement d'un message avec la fonction encrypt ou d'un re-chiffrement par le proxy en utilisant la fonction ReEncrypt

— Propriété « Key-optimal » - Un utilisateur (le délégué ou le délégateur) ne doit stocker et protéger qu'un nombre constant de données secrètes (e.g. pour un délégateur les clés privées des délégués) quel que soit le nombre de délégations qu'il délègue ou qu'il accepte. Dans le même temps, la taille et le nombre de clés que le proxy doit sauvegarder doivent également rester constants. Le but de cette propriété est de minimiser le coût de stockage chez chaque entité.

— Propriété de non-interactivité : Si la clé secrète sk_B du délégué (Bob) n'est pas requise dans l'algorithme de génération de clés de re-chiffrement ReKey, le schéma PRE est considéré comme non interactif. C'est-à-dire, une clé de re-chiffrement $rk_{A->B}$ peut être générée avec la paire de clés privée / publique du délégateur (sk_A, pk_A) et la clé publique pk_B du délégué. C'est-à-dire que la clé privée sk_B du délégué n'est pas requise comme entrée de l'algorithme ReKey.

— Propriété de non-transitivité : on dit un schéma PRE est non-transitif si les droits de déchiffrement ne peuvent pas être redéfinis par le proxy. D'une manière formelle, il est impossible pour le proxy de calculer $rk_{A->C}$ de $rk_{A->B}$ et $rk_{B->C}$.

— Temporary : Pour traiter le cas où le délégateur doit révoquer les droits de déchiffrement du délégué, il est souhaitable d'équiper le PRE de la propriété temporaire de tel sorte que le droit de re-chiffrement pour le proxy et le droit de déchiffrement pour le délégué puisse être supprimé selon la demande de délégateur. Cela signifie que le délégateur a toujours le pouvoir de révoquer le droit du délégué en mettant à jour les paramètres globales du PRE ou en donnant des instructions appropriées au proxy.

— Collusuion-resistant : Dans un système PRE résistant à la collusion, même la collusion du proxy avec le délégué, ni l'un ni l'autre ne peut récupérer la clé privée du délégateur.

II État de l'art des schémas PRE

Blaze *et al.* [161] sont les premiers à avoir proposé un schéma PRE dans le cas d'un proxy semi-honnête. Cette approche est fondée sur le cryptosystème ElGamal et sur la clé de re-chiffrement générée par le délégateur. Ce dernier doit l'envoyer au proxy afin de modifier le chiffré d'Alice pour qu'il soit déchiffrable par Bob (c.-à-d. les données chiffrées avec la clé publique de Bob). Un problème important de cette approche, remarquée par Ateniese *et al.* [122], est que le schéma de Blaze *et al.* est bidirectionnel. La clé de re-chiffrement permet de convertir des textes chiffrés d'Alice à Bob et également les les textes chiffrés de Bob sous la clé publique d'Alice. Ceci n'est pas acceptable pour Bob. L'origine de ce problème est que la clé de re-chiffrement dépend de la clé privée de Bob. Afin de le résoudre obtenir la propriété approche unidirectionnelle, différentes approches ont été proposées. La première classe de méthodes repose sur des cryptosystèmes classiques de chiffrement asymétrique. Par exemple, [164] profite d'un protocole basé sur le partage de données entre des proxys distribués. Chacun d'eux possède une partie des données d'Alice mais reçoit une clé de re-chiffrement indépendante de la clé privée de Bob. Cependant, avec cette approche, la sécurité de la clé privée d'Alice est sûre tant que les proxys sont honnêtes. Deux proxys suffisent à casser le système. Une alternative proposée dans [165], fonctionne avec un seul proxy où la clé de re-chiffrement fournie par Alice est divisée en deux parties : une pour le proxy et l'autre pour Bob. Malheureusement, parce que [165] utilise un cryptosystème symétrique, les données d'Alice ne sont pas re-chiffrées avec la clé publique de Bob. La deuxième classe regroupe les méthodes appelées re-chiffrement de proxy par l'identité (IBPRE pour "Identity-based proxy re-encryption"), une notion introduite par Green et Ateniese [166]. Une telle méthode mélange les principes du PRE avec la cryptographie fondée

sur l'identité (ID-based cryptography, "IBC"). Dans IBC, la clé de chiffrement publique d'un utilisateur est dérivée de son identité (par exemple, son adresse électronique). En la combinant avec le PRE, l'émetteur et le proxy ont juste besoin de connaître l'identité des délégués au lieu de vérifier leurs certificats. Dans ce cas, la propriété unidirectionnelle est atteinte en raison du fait que la clé de re-chiffrement dépend de l'identité du délégué. Cependant, il faut savoir que IB-PRE souffre du problème de la clé d'engagement (voir [165] pour plus de détails).

La plupart de ces systèmes reposent également sur des cryptosystèmes basés sur le couplage ou "pairing" [167–171] , une application considérée comme très coûteuse en termes de complexité de calcul par rapport à la multiplication ou l'exponentiation modulaire [172]. Pour surmonter ce problème, Deng *et al.* [173] ont proposé un système PRE basé sur les algorithmes de chiffrements asymétrique au lieu du couplage.

Au-delà, si les approches ci-dessus permettent à l'utilisateur de partager des données avec un autre, ils ne permettent pas le traitement de données chiffrées, par le cloud ou le proxy. Nous l'avons vu, cette capacité est accessible avec les cryptosystèmes homomorphes. La première tentative d'un PRE homomorphe (HPRE) a été proposée par Bresson *et al.* dans [79] sur la base du cryptosystème de Paillier [86]. Cependant, même si leur solution permet le partage de données, elle ne peut pas être considérée comme un schéma de PRE pur. En effet, les données ne sont pas re-chiffrées avec la clé publique du délégué. Si celui-ci veut demander au cloud de traiter les données qu'il reçoit d'Alice, il doit : i) d'abord télécharger les données d'Alice, ii) les déchiffrer en fonction de certaines informations secrètes fournies par Alice ; iii) les chiffrer avec sa clé publique et les renvoyer dans le cloud. Il existe donc toujours un besoin pour PRE homomorphe (HPRE). Dans ce qui suit, nous profitons des propriétés du chiffrement homomorphe pour proposer un nouveau concept de PRE.

III Un nouveau concept de partage de données chiffrées : Principe de base

L'objectif que nous nous sommes fixés, est de partager des données chiffrées chez un fournisseur de cloud entre plusieurs utilisateurs sous la contrainte : 1) qu'ils n'aient à les télécharger et les « reuploader » sur le cloud ; 2) que ces données puissent traitées à volonté sur le cloud ; 3) les claculs liés au partage sont tous réalisés par le Cloud sans interactions avec les utilisateurs. La solution à laquelle nous sommes arrivée est le premier schéma PRE fondé sur le chiffrement homomorphe.

Par définition, un PRE se compose de cinq fonctions : KeyGen, Rekey, Encrypt, Rencrypt et Decrypt. Un PRE est dérivé d'un cryptosystème asymétrique et s'en diffère diffère par les deux fonction : Rekey, qui permet de générer la clé de re-chiffrement ; et Rencrypt, qui permet de re-chiffrer un message.

L'approche que nous proposons fonctionne sur la base de cryptosystèmes asymétriques homomorphes et n'utilise pas le pairing. Par rapport à un PRE classique, les deux fonctions Rekey et Rencrypt n'existe pas. Nous utilisons d'autres fonctions qui sont les suivantes :

- La fonction D_{iff} qui permet de calculer la différence entre deux message chiffrés par le même cryptosystème.
- La fonction *GenRand* qui permet de générer une séquence pseudo-aléatoire dans le domaine chiffré. Plus clairement, cette fonction génère une séquence de nombres chiffrés.

Le principe de notre schéma de partage est le suivant. Supposons qu'Alice (le délégateur ou l'émetteur) veuille partager avec Bob (le délégué ou le destinataire) certaines données chiffrées par un cryptosystème homomorphe ; données qu'elle a précédemment externalisées dans le Cloud

(le Proxy). Pour ce faire, Bob et Alice se mettent d'accord sur une clé jetable, sans passer par le cloud. Alice chiffre cette clé et l'envoie au cloud. Le cloud utilise cette clé chiffrée pour paramétrer la fonction GenRand et générer une séquence aléatoire chiffrée. Une fois cette séquence générée, le cloud utilise la fonction D_{iff} pour calculer la différence entre cette séquence et les données chiffrées par Alice et qui sont à partager avec Bob. Comme nous le verrons par la suite, cette fonction tire avantage de la méthode de comparaison de données chiffrées présentée en Chapitre 2 section III.3. Le cloud chiffre les différences calculées avec la clé publique de Bob. De son côté, Bob doit simplement demander au Cloud d'éliminer le « bruit » qui a été ajouté aux données chiffrées pour accéder aux données qu'Alice veut partager avec lui et les traiter d'une manière externalisée, s'il le veut. À noter que dans cette approche tous les calculs sont effectués par le cloud sans interactions avec Alice et Bob.

Tel que présenté, la définition de notre schéma s'appuie sur un ensemble de cinq fonctions :

- KeyGen(k) - L'algorithme de génération de clé qui prend en entrée le paramètre de sécurité k, et qui retourne en sortie une paire de clés publique/privée (pk, sk).
- GenRand($pk_A, E[seed]$) - L'algorithme de génération d'une séquence aléatoire chiffrée. Il prend en entrée la clé publique de l'émetteur et le chiffré d'une « graine », nous verrons qu'il s'agit de la clé de partage, pour retourner en sortie une séquence pseudo aléatoire chiffrée.
- Encrypt(m, pk) - C'est la fonction de chiffrement qui permet de chiffrer le message m par la clé publique pk pour générer un message chiffré c.
- $D_{iff}(E[a], E[b])$ - Algorithme de calcul de différences entre deux messages chiffrés. Cette fonction calcule la différence entre a et b à partir de $E[a]$ et $E[b]$.
- Decrypt(c, sk) - Fonction de déchiffrement qui prend en entrée un message chiffré c et une clé secrète sk, et retourne le message en clair m.

Dans la section suivante, nous présentons une mise en oeuvre possible de ce HPRE sur la base d'un générateur congruentiel linéaire combiné (SCLCG « Secure combined linear congruetial generator ») dans le domaine chiffré de Paillier. Ce SCLCG fournit une séquence de nombres aléatoires chiffrés.

Dans la section IV, nous reviendrons sur l'analyse de sécurité de cette approche. Nous démontrerons en particulier que lors de l'exécution de notre schéma, un proxy honnête mais curieux ne peut pas déduire d'informations sur le message m ou les clés privées des utilisateurs (sk_A ou sk_b).

IV Une mise en oeuvre avec le chiffrement de Paillier

IV.1 Générateur pseudo aléatoire sécurisé

Comme indiqué ci-dessus, dans notre schéma, il revient au Cloud de générer de manière sécurisée une séquence aléatoire d'entiers chiffrés. Le cloud ne doit pas connaître les paramètres du générateur ni les valeurs aléatoires en clair.

Le générateur que nous proposons de sécuriser est CLCG [174] (Générateur Congruentiel Linéaire Combiné). Celui-ci est donné par la somme de deux générateurs congruentiels linéaires. Un générateur congruentiel linéaire (LCG) est basé sur la congruence et une fonction linéaire :

$$X_{n+1} = aX_n + c \mod m \tag{5.3}$$

où X_n est la $n^{ème}$ valeur de la séquence LCG ; a est le multiplicateur ; c est l'incrément ; m est le modulo ; et, X_0 présente le terme initial, également appelé le germe ou encore la clé secrète du LCG.

La génération d'une séquence aléatoire CLCG résulte de la combinaison de deux séquences LCG générées telles que :

$$Z_{n+1} = X_{n+1} + Y_{n+1} \mod m = a_X X_n + c_X + a_Y Y_n + c_Y \mod m \tag{5.4}$$

où Z_n est la $n^{ème}$ valeur de la séquence aléatoire généré par CLCG ; a_X et a_Y sont les multiplicateurs du LCG ; c_X et c_Y sont les incréments du CLCG ; m est le modulo ; X_0 et Y_0 sont les termes initiaux du $CLCG$ pour les deux séquences X_i et Y_i, respectivement.

Il est important de noter que la sécurité d'un CLCG se situe sur les germes X_0 et Y_0. La connaissance des paramètres (a_x, a_y), (c_x, c_y) et m ne met pas en danger sa sécurité [175]. Sous la contrainte que le modulo, i.e m, dans l'équation 5.3 égale la clé publique K_p de Paillier, l'implémentation du CLCG dans le domaine chiffré de Paillier (i.e. $\{E[Z_n, r_n]\}_{n=0...N-1}$) s'exprime de la manière suivante :

$$E[Z_{n+1}, r_{Z,n+1}] = E[X_{n+1}, r_{X,n+1}]E[Y_{n+1}, r_{Y,n+1}] \tag{5.5}$$

avec

$$E[X_{n+1}, r_{X,n+1}] = E[X_n, r_{X,n}]^{a_X} E[c_X, r_{c_X}] = E[a_X X_n + c_X, r_{X,n}^{a_X} r_{c_X}] \tag{5.6}$$

$$E[Y_{n+1}, r_{Y,n+1}] = E[Y_n, r_{Y,n}]^{a_Y} E[c_Y, r_{c_Y}] = E[a_Y Y_n + c_Y, r_{Y,n}^{a_Y} r_{c_Y}] \tag{5.7}$$

Ce générateur CLCG sécurisé (SCLCG) génère donc une sequence de valeurs entières pseudo-aléatoires chiffrées. À noter par ailleurs que le période du SCLCG est très longue du fait que $m = K_p$. Par exemple, dans le cas où K_p est codé sur 1024 bits, la période SCLCG est $\frac{(K_p-1)^2}{2} \simeq 2^{2047}$ [176].

Si les incréments et tous les termes de la séquence sont chiffrées (y compris les clés du SCLCG), ce n'est pas le cas des multiplicateurs (a_X, a_Y) (la raison réside dans les propriétés du cryptosysteme de Paillier, voir chapitre 2 section III.1, eq. (5.25)). Cela ne réduit cependant pas la sécurité de notre système car les paramètres (a_X, a_Y) ne sont pas censés être secrets [175].

Il est important de souligner qu'il existe une relation récursive entre les aléas r_Z, de notre SCLCG. Ces aléas assurent la sécurité sémantique du cryptosystème de Paillier. Cette relation de récurrence est liée aux relations (5.6) et (5.6), et est de la forme :

$$r_{Z,n+1} = r_{X,n+1} r_{Y,n+1} = r_{X,n}^{a_X} r_{c_X} r_{Y,n}^{a_Y} r_{c_Y} \tag{5.8}$$

où r_{c_X} et r_{c_Y} désignent les variables aléatoires utilisées comme aléas pour chiffrer les incréments. Cette relation récursive jouera un rôle important dans l'échange de données entre deux utilisateurs différents.

IV.2 Schéma global

Dans ce qui suit, nous définissons d'abord notre schéma d'échange de données et les hypothèses de sécurité que nous avons considérées avant de présenter notre schéma HPRE.

Comme indiqué précédemment, notre objectif est de permettre aux utilisateurs de partager leurs données sous la contrainte que le délégué ne doit pas les télécharger pour les rechiffrer avec la clé publique du délégateur et les télécharger à nouveau dans le Cloud. Nous voulons également que l'échange de données soit effectué par le cloud (proxy) sans lui donner la clé secrète du délégué ainsi qu'avec très peu voire aucune communications entre le délégué, le proxy et le délégateur. Dans notre solution, si un utilisateur veut partager des données avec plusieurs utilisateurs à la fois, tous devront s'entendre sur un seul secret avec le délégué : une clé de partage. Une dernière hypothèse est que toutes les communications sont protégées à l'aide du cryptosystème de Paillier. Les attaques par écoutes ne peuvent pas donc inférer les messages transmis.

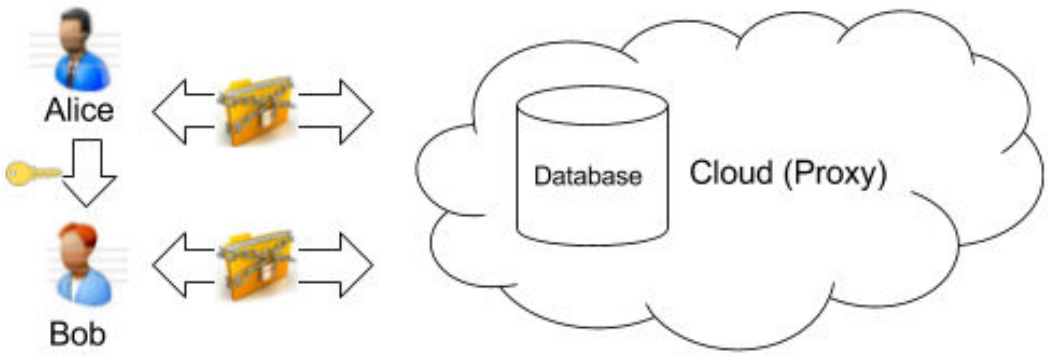


FIGURE 5.1 – Schéma général du partage de données dans un environnement Cloud.

IV.2.1 Partage de données externalisées sécurisé

Considérons qu'Alice (le délégué) veut partager avec Bob (le délégateur) un ensemble de données dont elle est la propriétaire. Ces données peuvent être un ensemble de valeurs entières, comme par exemple une image en niveau de gris I de N pixels $I = \{I_i\}_{i=0...N-1}$, dont les pixels sont codés sur b bits. On suppose qu'Alice a déjà externalisé dans le Cloud une image chiffrée par le cryptosystème de Paillier. Plus clairement, les pixels de l'image sont chiffrés de manière indépendante avec la clé publique K_{p1} d'Alice, tel que (voir l'étape d'externalisation des données en Figure 5.2)

$$I_i^e = E_{K_{p1}}[I_i, r_i] \tag{5.9}$$

où r_i est la valeur aléatoire associée au $i^{ème}$ pixel I_i de I, I_i^e est la version chiffrée de I_i.

Comme nous le verrons par la suite, notre procédure HPRE impose une contrainte sur la façon dont Alice génère les valeurs aléatoires $\{r_i\}_{i=0..N-1}$. Ceux-ci doivent satisfaire la relation de récurrence (5.8) et, pour un fichier qu'Alice stocke dans le Cloud, elle doit mémoriser les valeurs aléatoires r_{c_X}, r_{c_Y} et r_0 ; valeurs qu'elle a utilisées pour chiffrer le premier pixel de l'image : $I_0^e = E_{K_{p1}}[I_0, r_0]$.

Pour partager cette image chiffrée avec Bob, ou plus clairement la rechiffrer avec la clé publique de Bob K_{p2}, nous proposons la procédure HPRE suivante qui s'appuie sur quatre phases (voir la Figure 5.2) :

— **Entente des utilisateurs** - Dans cette étape, Bob et Alice s'accordent sur les paramètres du SCLCG pour l'échange : les clés secrètes (X_0, Y_0), les multiplicateurs (a_X, a_Y) et les incréments (c_X, c_Y).

— **Génération de séquence aléatoire secrète** Alice chiffre (X_0, Y_0) et (c_X, c_Y) avec sa clé publique K_{p1} et les envoie au Cloud. Notez que Z_0 est chiffré avec le même aléa que le premier pixel de son image : $r_0 = r_{Z,0} = r_{X,0}r_{Y,0}$. Pour simplifier les notations, nous noterons dans la suite r_i l'aléa $r_{Z,i} = r_{X,i}r_{Y,i}$. Ainsi, sur la base de ces informations, le Cloud génère la séquence aléatoire secrète avec (5.8) $Z^e = \{Z_i^e = E_{K_{p1}}[Z_i, r_i]\}_{i=0..N-1}$.

— **Chiffrement des données pour le délégateur** - Cette procédure repose sur différentes étapes :

1. *Calcul des différences entre les données chiffrées d'Alice* (I^e) *et la séquence aléatoire secrète* (Z^e) : Pour ce faire, nous utilisons la fonction D_{iff}^e que nous avons définie

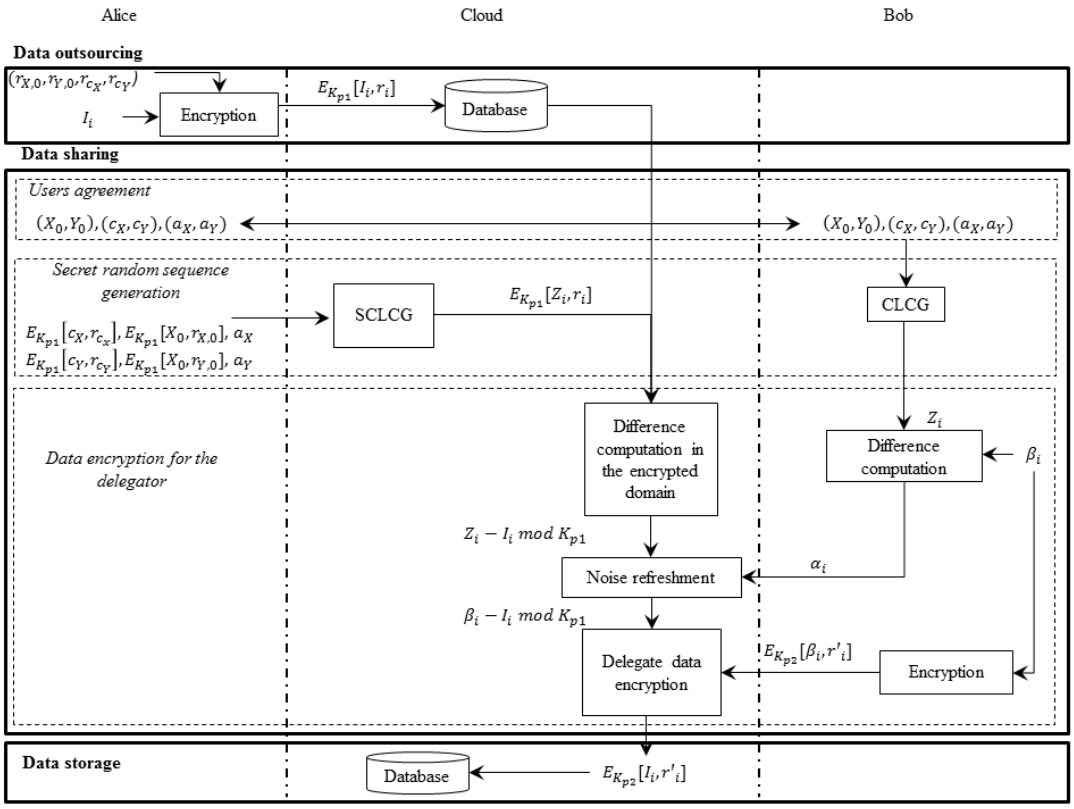


FIGURE 5.2 – Les étapes principales de notre schéma pour partager une image

dans le chapitre 2, section III. Comme Z_i^e et I_i^e sont chiffrés avec la même clé publique K_{p1} et les mêmes aléas r_i, le Cloud peut calculer leurs différences D_i comme suit :

$$\begin{array}{rcl} D_i & = & Dif f(Z_i, I_i) = Dif f^e(E_{K_{p1}}[Z_i, r_i], E_{K_{p1}}[I_i, r_i]) \\ D_i & = & Z_i - I_i \mod K_{p1} \end{array} \tag{5.10}$$

Il est important de souligner que les valeurs D_i ne sont pas chiffrées. Ce sont des valeurs en clair.

2. *Chiffrement des différences avec la clé publique de Bob* : Le chiffrement de la séquence de différences $D = \{D_i\}_{i=1...N}$, avec la clé publique de Bob avec la garantie de ne pas avoir d'erreurs dans les données déchiffrés, n'est pas possible que si la contrainte suivante est vérifiée : $D_i \mod K_{p1} = D_i \mod K_{p2}$, ou bien $0 < D_i < \min(K_{p1}, K_{p2})$. Cette contrainte est cependant difficile à satisfaire en raison de l'amplitude de la sortie du SCLCG qui ne peut pas être contrôlée de manière simple. Pour résoudre ce problème, notre schéma comprend une "une étape de rafraîchissement du bruit" appliquée avant le chiffrement de D. Comme illustré en Figure 5.2, au départ de cette procédure, Bob génère de son côté la séquence $\{Z_i\}_{i=0..N-1}$ en utilisant un CLCG paramétré comme le SCLGC du Cloud. Il produit également un second bruit $\{\beta_i\}_{i=0..N-1}$ tel que :

$$2^b - 1 < \beta_i < \min(K_{p1}, K_{p2}) \tag{5.11}$$

où b est le nombre de bits sur lequel un pixel d'image est codé. Sous cette contrainte, nous nous assurons que : $\beta_i \mod K_{p1} = \beta_i \mod K_{p2}$ et $\beta_i - I_i \mod K_{p1} = \beta_i - I_i \mod K_{p2}$.

Bob envoie au Cloud le chiffré de ce second bruit, i.e $\{E_{K_{p2}}[\beta_i, r'_i]\}_{i=0...N-1}$ où r'_i est une valeur aléatoire définie par Bob accompagnée de la séquence de différence entre ce bruit et la sequence de bruit original en clair, i.e $\{\ \alpha_i = \beta_i - Z_i \mod K_{p1}\}_{i=0..N-1}$. À la réception de ces informations , le cloud ou le proxy peut supprimer le bruit $Z = \{Z_i\}_{i=0...N-1}$, , des données $D = \{D_i\}$ de Bob en calculant

$$G_i = \alpha_i + D_i \mod K_{p1} = \beta_i - I_i \mod K_{p1} \tag{5.12}$$

Puis, le proxy chiffre $\{G_i\}_{i=0..N-1}$ avec la clé publique de Bob.

$$\{E_{K_{p2}}[G_i, r''_i] = E_{K_{p2}}[\beta_i - I_i, r''_i]\}_{i=0..N-1} \tag{5.13}$$

De cette manière, on garantit qu'il n'y aura pas d'erreurs au déchiffrement.
Enfin, pour supprimer le bruit β_i des données de Bob, le serveur calcule

$$E_{K_{p2}}[I_i, r'_i r''^{-1}_i] = E_{K_{p2}}[\beta_i, r'_i] E_{K_{p2}}[\beta_i - I_i, r''_i]^{-1} \tag{5.14}$$

À la fin de cette procédure, Bob possède sur le cloud l'image d'Alice chiffrée avec sa propre clé publique. Il est possible de remarquer que l'accès aux données partagées est fondé sur la connaissance de la clé secrète du SCLCG, i.e Z_0, générée par Alice en accord avec Bob.
Afin de donner à Bob la possibilité de partager cette image avec un autre utilisateur, sans devoir la télécharger et la re-externaliser l'image, le Cloud doit simplement prendre $r''_i = 1$ dans (5.13). En effet, dans ce cas, Bob obtient l'accès à $E_{K_{p2}}[I_i, r'_i]$ où r'_i est sous son contrôle. C'est lui génère ces aléas en utilisant (5.8).
Ce système permet donc l'échange de données entre Alice et Bob, sans communications supplémentaires entre le Cloud et Alice, et le téléchargement de données par Bob. Comme il s'appuie sur le chiffrement homomorphe, les données peuvent être traitées par le cloud sans compromettre leur confidentialité.

IV.3 Amélioration : un schéma HPRE zéro communication

Un des défauts de la solution précédente est la nécessité pour Bob de régénérer les séquences de bruit $Z = \{Z_i\}_{i=1...N}$ et $\beta = \{\beta_i\}_{i=1...N}$ qu'il envoie chiffrée au Cloud. Ces deux séquences sont nécessaire pour la procédure de rafraichissement qui garantit l'absence d'erreur dans le déchiffrement. Une alternative, est que cette étape de rafraichissement soit elle aussi réalisée par le cloud. Nous rappelons que dans l'étape de chiffrement de la séquence de différences $D = \{D_i\}_{i=1...N}$, avec la clé publique de Bob avec la garantie de ne pas avoir d'erreurs dans les données déchiffrés, n'est pas possible que si la contrainte suivante est vérifiée : $D_i \mod K_{p1} = D_i \mod K_{p2}$, ou bien $0 < D_i < \min(K_{p1}, K_{p2})$. Pour résoudre ce problème, nous proposons la solution suivante :

— Lors de la génération de la séquence chiffrée $Z^e = \{E[Z_i, r_i]\}_{i=1...N}$ par le cloud, ce dernier il ajoute un message de la forme 2^{b+1} à la séquence Z comme :

$$Z_i^{e'} = E[Z_i + 2^{b+1}, r_i] = E[Z_i, r_i] \times (1 + 2^{b+1} K_p)\ mod K_p^2 \tag{5.15}$$

pour tout $i \in \{1, ..., N\}$, b est le nombre de bits auquel les pixels de l'image à partager I est codé.
Dans ces cas, après le calcul de la différence entre $Z^{e'}$ et I^e : $D'_i = D^e_{iff}(Z^{e'}, I^e)$ vérifie la condition $D'_i \mod K_{p1} = D'_i \mod K_{p2}$. Et par conséquence, notre schéma n'a plus besoin de l'étape de rafraichissement

L'avantage de cette solution est la réduction de la complexité de calcul de Bob parce que nous n'avons plus besoin de générer le vecteur β_i.

IV.4 Analyse de sécurité

Le schéma proposé permet à deux utilisateurs, Alice (le délégué) et Bob (le délégateur), de partager des données stockées dans le cloud (le proxy) que nous considérons comme semi-honnête. Dans ce qui suit, nous discutons de sa sécurité en termes de confidentialité des données, d'unidirectionnalité et de collusion. Avant de rentrer dans les détails, rappelons les différents éléments que le cloud connaît ou a accès correspondent :

— aux clés publiques d'Alice (K_{p1}) et de Bob (K_{p2})

— les données chiffrées d'Alice : $\{E_{K_{p1}}[I_i, r_i]\}_{i=0...N-1}$

— les multiplicateurs (a_X, a_Y)

— les incréments chiffrés ($E_{K_{p1}}[c_X, r_{c_X}]$, $E_{K_{p1}}[c_Y, r_{c_Y}]$)

— La séquence chiffrée générée par le SCLCG : $\{E_{K_{p1}}[Z_i, r_i]\}_{i=0...N-1}$

La sécurité de notre schéma repose principalement sur SCLCG qui joue un rôle important dans notre système. Si le Cloud accède à la séquence aléatoire $\{Z_i\}_{i=0...N-1}$, il peut déchiffrer toutes les données que Alice et Bob veulent partager. Notre SCLCG est paramétré avec ($E_{K_{p1}}[c_X, r_{c_X}]$, $E_{K_{p1}}[c_Y, r_{c_Y}]$), (a_X, a_Y) et ($E_{K_{p1}}[X_0, r_{X,0}]$, $E_{K_{p1}}[Y_0, r_{Y,0}]$) pour générer $\{E_{K_{p1}}[Z_i, r_i]\}_{i=0...N-1}$. Même si a_X et a_Y sont en clairs, donc connus au cloud, ce dernier ne peut pas inférer d'information sur c_X, c_Y, X_0 et Y_0 à partir de $\{Z_i\}_{i=0...N-1}$, car tous les deux ne contribuent pas à la sécurité du CLCG (voir section IV).

Comme indiqué auparavant, les données d'Alice $\{I_i\}_{i=0...N-1}$ et la séquence aléatoire $\{Z_i\}_{i=0...N-1}$ sont chiffrées par le cryptosystème de Paillier avec les mêmes valeurs aléatoires $\{r_i\}_{i=0...N-1}$ sous la contrainte (5.8). Cette relation ne met pas en danger la confidentialité des données. L'analyse de sécurité du cryptosystème Paillier a été étudiée dans [86] pour différents modèles d'attaques : Attaque à texte chiffré seulement ("ciphertext-only attack"), Attaque à texte clair connu ("known-plaintext attack"), Attaque à texte clair choisi ("chosen-plaintext attack") et Attaque à texte chiffré choisi ("chosen-ciphertext attack"). Ces attaques correspondent à différents scénarios, considérant l'accès ou non à certaines connaissances *a priori* qui pourraient aider à découvrir la clé secrète de l'utilisateur ou à déchiffrer des données sans la clé secrète. Il a été démontré que si le cryptosystème de Paillier n'est pas sécurisé sous l'attaque à texte chiffré choisi, comme tous les cryptosystèmes homomorphes réels, il est sécurisé contre les trois autres attaques. Notre HPRE atteint les mêmes performances même s'il existe une relation récursive entre les valeurs aléatoires utilisées pour le chiffrement des pixels consécutifs d'une image. Plus clairement, du fait que les valeurs aléatoires de l'image chiffrée sont générées à partir de l'équation (5.13), un attaquant peut tenter de se synchroniser en termes d'aléas et calculer la différence entre deux pixels de l'image en utilisant la fonction D_{iff}. Prenons le cas de deux pixels consécutifs. Le Cloud peut essayer de synchroniser l'aléa d'un pixel chiffré $E[I_i, r_i]$ avec l'aléa r_{i+1} du pixel suivant en utilisant de la récursivité entre aléas, c.-à-d :

$$r_{i+1} = r_{X,i+1} r_{Y,i+1} = r_i r_{X,i}^{a_X - 1} r_{c_X} r_{Y,i}^{a_Y - 1} r_{c_Y} \tag{5.16}$$

Dans notre schéma, le cloud ne connait pas les aléas en clair et n'accède qu'aux données chiffrées suivantes : $\{E[X_i, r_{X,i}]\}_{i=0...N-1}$, $\{E[Y_i, r_{Y,i}]\}_{i=0...N-1}$, $E[c_X, r_{c_X}]$, $E[c_Y, r_{c_Y}]$, a_X, a_Y et $\{E[Z_i, r_i]\}_{i=0...N-1}$). Pour se synchroniser en termes d'aléas, il faudrait que le cloud puisse calculer $E[I_i, r_a]$ et $E[I_{i+1}, r_a]$.

Sur la base des données à sa disposition, il peut tout au plus calculer

$$E[I_i+X_i(a_X-1)+c_X+Y_i(a_Y-1)+c_Y] = E[I_i,r_i]E[X_i,r_{X,i}]^{a_X-1}E[c_X,r_{c_X}]E[Y_i,r_{Y,i}]^{a_Y-1}E[c_Y,r_{c_Y}] \tag{5.17}$$

et obtenir la différence

$$D_{i+1} - D_i = Z_{i+1} - Z_i - I_i - I_{i+1} \tag{5.18}$$

à partir de laquelle il ne peut déduire aucune information sur I_i car il ne connait pas Z_{i+1} et Z_i dont il ne peut déduire aucune information sur I_i.

Au delà, si le cloud connaît la valeur aléatoire initiale r_0 associée au chiffrement de la clé du SCLCG et du premier pixel de l'image d'Alice, il pourra déchiffrer I_0 et Z_0. Il ne pourra cependant pas déchiffrer le reste de l'image. En effet, il faudrait pour cela qu'il accède aux valeurs (r_{c_X}, r_{c_Y}). Dans notre schéma, le Cloud a seulement accès à $E_{K_{p1}}[X_0, r_{X,0}], E_{K_{p1}}[Y_0, r_{Y,0}])$ et $(E_{K_{p1}}[c_X, r_{c_X}], E_{K_{p1}}[c_Y, r_{c_Y}])$ dont il ne peut dériver aucune information.

Une fois que le cloud a calculé la séquence aléatoire sécurisée $\{E_{K_{p1}}[Z_i, r_i]\}_{i=0...N-1}$, il calcule les différences entre cette séquence et les données d'Alice $\{E_{K_{p1}}[I_i, r_i]\}_{i=0...N-1}$, et obtenir à la séquence en clair $D = \{D_i = I_i - Z_i\}_{i=0...N-1}$. D correspond à une version masquée de I par la séquence générée à partir du CLCG. Dans le cas d'une image codée sur b bits et de N pixels, il faudra utiliser une attaque de force brute pour récupérer l'image, c'est-à-dire $2^{b.N}$ opérations. Appliqué à l'image de la Figure 5.3, l'attaquant a besoin $2^{8\times122\times92}$ opérations, ce qui n'est pas faisable.

Un schéma PRE unidirectionnel assure que le proxy (le Cloud) ne peut convertir qu'un texte chiffré sous la clé publique d'Alice en un texte chiffré sous la clé publique de Bob mais pas l'inverse. Plus clairement, le proxy ne doit pas pouvoir chiffrer certaines données de Bob sous la clé publique d'Alice. Notre système a cette propriété. Le partage de données est fondé sur le calcul des différences entre les données chiffrées en synchronisant les valeurs aléatoires (c'est-à-dire, les aléas r_i) suivies d'une procédure de rafraîchissement du bruit sous le contrôle de Bob uniquement. Une conséquence de ce rafraîchissement est la modification des valeurs aléatoires utilisées pour chiffrer les données d'Alice. Il n'est donc pas possible pour le Cloud de réutiliser la séquence aléatoire SCLCG paramétrée par Alice afin de reconvertir les données que Bob a reçues sans son accord. Bob devra se mettre à nouveau d'accord avec Alice sur de nouveaux paramètres d'échange. En résumé, il est impossible pour le cloud de convertir les données de Bob sans son accord. C'est également le cas même si Alice et le Cloud colludent, car la procédure de rafraichissement du bruit n'est que sous le contrôle de Bob. C'est analyse est aussi valable pour la version améliorée de notre schéma.

La résistance à la collusion est une autre propriété de sécurité importante dans le partage de données. Si Bob et le Cloud colludent, ayant ainsi la connaissance de l'ensemble des paramètres du système ainsi que des données partagées sous forme chiffrée et en clair, ils ne doivent pas pouvoir trouver la clé privée d'Alice (K_{s1}). Une telle situation correspond au modèle d'attaque CPA. Du fait que nous utilisons le cryptosystem de Paillier [86], notre système satisfait cette propriété. En conséquence, le Cloud et Bob ne peuvent pas récupérer la clé privée d'Alice à partir de sa clé publique et un ensemble de données d'Alice chiffrées et en clair. Enfin, si Alice veut partager des données avec un autre utilisateur que Bob, il faut qu'elle change la clé de partage (X_0, Y_0) (la graine du SCLCG), sinon le cloud et Bob pourront accéder à ces données sans permission.

En résumé, notre schéma est unidirectionnel, résistant à la collusion et empêche un proxy semi-honnête (c'est-à-dire le Cloud) d'apprendre : la clé privée de Bob et Alice ; la version en clair des données d'Alice ou Bob sans leurs clés privées. En plus, notre solution peut être appliquée à toute sorte de documents (par exemple, fichiers PDF ou Word).

FIGURE 5.3 – Exemples d'images de visage de la base de données

IV.5 Résultats expérimentaux

Cette solution a été expérimentée dans le cas du partage d'images non compressées. On a utilisé 400 images tests issues de la base de données du Laboratoire de recherche Olivetti de Cambridge (Royaume-Uni). Elles sont encodées sur 8 bits et de taille 92×112 pixels (voir Figure 5.3). Les images ont été chiffrées avec des clés publiques de plus de 1024 bits afin de fournir un haut niveau de sécurité. La performance de notre système est évaluée en termes de complexité de stockage et de calcul. Notre schéma a été mis en oeuvre en C/C++ avec la bibliothèque GMP et toutes les expériences ont été menées à l'aide d'une machine équipée de quatre coeurs, de 23 Go de RAM et fonctionnant sur Ubuntu 14.04 LTS.

Complexité de stockage : dans le cas où les images sont chiffrées avec une clé de 1024 bits, une image chiffrées représente $2,7Mo$ dans le Cloud. Pour une image qu'Alice externalise, elle doit conserver de son côté les valeurs aléatoires $(r_{X,0}, r_{Y,0}, r_{c_X}, r_{c_Y})$. Pendant un échange d'images, Alice envoie la graine chiffrée $E_{K_{p1}}[Z_0, r_0]$, les incréments chiffrés $(E_{K_{p1}}[c_X, r_{c_X}], E_{K_{P1}}[c_Y, r_{c_Y}])$ et les multiplicateurs (a_X, a_Y). Cette quantité de données est bornée par $O(log_2(K_{p1}^2))$. De son côté, Bob doit stocker (X_0, Y_0), la clé secrète CLCG, mais seulement pour un échange.

Complexité de calcul : Du côté d'Alice, la complexité de calcul est limitée au chiffrement des paramètres du SCLCG (c.-à-d. X_0, Y_0, c_X et c_Y). En ce qui concerne le Cloud, il doit calculer la séquence aléatoire secrète, la différence entre les données chiffrées, le bruit de rafraîchissement, la version chiffrée des différences rafraichies et l'élimination du bruit. Pour une image de N pixels, la génération de séquence aléatoire secrète est équivalente à N chiffrements. Il en est de même pour le calcul des différences $\{D_i\}_{i=0...N-1}$. Comme la procédure de rafraîchissement du bruit consiste en des additions modulaires, sa complexité est négligeable par rapport aux opérations de chiffrement. Le chiffrement des différences $\{G_i\}_{i=0...N-1}$ est constitué de N chiffrements. Pour résumer, la complexité de calcul pour le Cloud est bornée par $O(3 \times N)$ chiffrement. La complexité de calcul de Bob repose sur la procédure de rafraîchissement du bruit. Il doit générer une séquence aléatoire du CLCG (c'est-à-dire $\{Z_i\}_{i=0...N-1}$), une tâche dont la complexité est négligeable par rapport aux N chiffrements du bruit(C'est-à-dire $\{\beta_i\}_{i=0...N-1}$) qu'il produit également et qu'il envoie ensuite au Cloud. La complexité de calcul de Bob est donc de N chiffrements et N soustractions modulaires. Nous fournissons dans le tableau 5.1 la quantité de données que chaque entité doit stocker ainsi que le temps de calcul requis dans le cas de partage d'image de notre base de données. Notre schéma prend environ $1'30$ minutes afin de partager une image avec un ordinateur standard.

En ce qui concerne la version améliorée de notre schéma HPRE, la complexité de calcul de Bob est N chiffrements sans les N soustractions. Par contre la complexité de calcul cloud augmente de N multiplications modulaire.

V Généralisation à d'autres cryptosystèmes

L'extension de notre schéma HPRE à d'autres cryptosystèmes dépend de leur capacité de : i) permettre le calcul de différence entre des données chiffrées avec les mêmes aléas ou bruit, ii)

entités	Délégateur (Alice)	Proxy (Cloud)	Délégué (Bob)
Temps de calcul (sec)	0.002	90	30
Taille de données chiffrées (bits)	0	22986753	2048

TABLE 5.1 – Quantité d'informations stockées (en bits) ainsi le temps de calcul correspondant à chaque entité (Alice, Bob et le Cloud) pour partager une image de 92×122 pixels.

implémenter de manière sécurisée le générateur congruentiel linéaire combiné, ou tout équivalent. C'est le cas du cryptosystème Damgärd-Jurik et du cryptosystème BGV. Nous montrons dans ce qui suit comment une différence peut être calculée avec ces cryptosystèmes, et comment implémenter le SCLCG, leurs autres propriétés permettant de les intégrés facilement dans notre HPRE.

V.1 Damgärd-Jurik

Le cryptosystème de Damgärd-Jurik est une généralisation du cryptosystème de Paillier. Il utilise des calculs modulo K_p^n au lieu de K_p, où K_p est la clé publique du cryptosystème de Paillier et $n \in \mathbb{N}^*$ est un entier naturel. La clé privée est la même pour les deux cryptosystèmes.

Comme nous l'avons évoqué dans le chapitre 1 section III.2, chaque cryptosystème homomorphe est définie par quatre fonctions ($KeyGen()$, $Enc()$, $Dec(.)$ et $Eval(.)$). La description de ces fonctions pour le cryptosystème de Damgärd-Jurik est la suivante :

- $KeyGen()$: On choisit deux nombres premiers distincts de grande taille p et q, dont on calcule le produit $K_p = pq$ et $K_s = PPCM(p-1, q-1)$. On choisit un entier $g = (1+K_p)^j x$ où j est une variable connue premier avec K_p et $x \in \mathbb{Z}^*_{K_p^n}$.
- $Enc(m)$: Pour un message $m \in \mathbb{Z}_{K_p^n}$, on choisit aléatoirement un nombre $r \in \mathbb{Z}^*_{K_p}$, le chiffrement est le suivant :

$$c = E[m, r] = g^m r^{K_p^n} \mod K_p^{n+1} \tag{5.19}$$

  Il est important de noter que l'entier r rend le cryptosystème de Damgärd-Jurik probabiliste ou sémantiquement sûr. Comme introduit dans [87], il est possible d'obtenir une version rapide de ce chiffrement (5.19) en fixant $g = 1 + K_p$ sans réduire la sécurité de l'algorithme. Par conséquent, le chiffrement de m n'exige qu'une seule exponentiation modulaire et deux multiplications modulaires

$$c = E[m, r] = (1 + mK_p) r^{K_p^n} \mod K_p^{n+1} \tag{5.20}$$

- $Dec(c)$: Pour un message chiffré $c \in \mathbb{Z}^*_{K_p^{n+1}}$, le déchiffrement passe par deux étapes. La première consiste à calculer

$$c^{K_s} = (1 + K_p)^{mK_s} \mod K_p^{n+1} \tag{5.21}$$

  et la seconde cherche à accéder à la valeur mK_s à partir de c^{K_s}. Pour ce faire, Damgärd-Jurik propose un algorithme itératif pour trouver m à partir de $(1 + K_p)^m \mod K_p^n$. Cette procédure s'appuie sur le théorème binomial et une fonction $L(.)$ définie telle que $L(b) = \frac{b-1}{K_p}$ et utilisée comme suit. En prenant en entrée la valeur $a = (1 + K_p)^m$

mod K_p^{n+1}, cet algorithme calcule d'abord $L(a \mod K_p^2)$ qui donne accès à $m_1 = L(a \mod K_p^2) = m \mod K_p$. Puis en calculant itérativement m_j jusqu'à $j = n$,

$$m_j = L(a \mod K_p^{j+1}) - (C_2^{m_{j-1}} K_p + ... + C_j^{m_{j-1}} K_p^{j-1}) \mod K_p^j \tag{5.22}$$

À la fin l'algorithme arrive donc à $m_n = m \mod K_p^n$. Cette procédure que nous notons comme la fonction $F(.)$ est décrite dans l'algorithme 3.

Algorithm 3 Damgärd-Jurik algorithm

```
1: procedure F(a)
2:     m ← 0
3:     for j ← 1, n do                                   ▷ m = m_{j-1}
4:         t1 ← L(a mod K_p^{j+1})
5:         t2 ← m
6:         for k ← 2, j do                               ▷ t2 = m(m-1)...(m-k+2)
7:             m ← m − 1
8:             t2 ← t2 * m mod K_p^j
9:             t1 ← t1 − (t2*K_p^{k-1})/k! mod K_p^j     ▷ t1 = t1 − C_k^i K_p^{k-1}
10:        end for
11:        m ← t1
12:    end for
13:    return m mod K_p^n
14: end procedure
```

le déchiffrement d'un message chiffré c en m est

$$m = F(c^{K_s}) K_s^{-1} \mod K_p^n \tag{5.23}$$

- Propriétés d'homomorphies ($Eval(c_1 = E[m_1, r_1], c_2 = E[m_2, r_2])$) : Soit c_1 et c_2 deux messages chiffrés de m_1 et m_2 lors les propriétés d'homomorphie du cryptosystème de Damgäard-Jurik sont les suivantes :

$$E[m_1, r_1] \times E[m_2, r_2] = E[m_1 + m_2, r_1 r_2] \tag{5.24}$$

$$E[m_1, r_1]^{m_2} = E[m_1 m_2, r_1^{m_2}] \tag{5.25}$$

  Le cryptosystème de Damgärd-jurik est donc, homomorphiquement additif.

- Calcul de la différence de données chiffrées

Considérons une relation client-serveur où le serveur a deux messages chiffrés $E_{K_p}[a, r]$ et $E_{K_p}[b, r]$ par le client, où a et b sont deux entiers dont nous voulons calculer la différence $d = a - b$ à partir de leurs chiffrés. La solution que nous proposons suppose que les deux messages sont chiffrés avec le même aléa r. Sous cette contrainte, nous pouvons dériver directement la différence d de $E_{K_p}[a, r]$ et $E_{K_p}[b, r]$ en exploitant la fonction itérative $F(.)$ du cryptosystème

de Damgärd-Jurik et l'hypothèse de simplification sur g, i.e. $g = 1 + K_p$, comme suit :

$$\begin{aligned} d &= D_{iff}(a,b) = D^e_{iff}(E_{K_p}[a,r], E_{K_p}[b,r]) \\ d &= F(E_{K_p}[a,r]E_{K_p}[b,r]^{-1} \mod K_p^{n+1}) \mod K_p^n \\ d &= F(g^a r^{K_p^n} g^{-b} r^{-K_p^n} \mod K_p^{n+1}) \mod K_p^n \\ d &= F(g^{a-b} \mod K_p^{n+1}) \mod K_p^n \\ d &= F((1+K_p)^{a-b} \mod K_p^{n+1}) \mod K_p^n \\ d &= a - b \mod K_p^n \end{aligned} \tag{5.26}$$

- Générateur congruentiel linéaire combiné sécurisé.

La génération d'une séquence aléatoire sécurisée par le cryptosystème de Damgârd-jurik est équivalente à celle que nous avons présentée dans la section IV.2 car le cryptosystème de Damgärd-jurik est une généralisation du cryptosystème de Paillier.

Sur les bases des mêmes notations utilisées dans la section IV.2, nous limitons ici la sécurisation d'un LCG. En, effet, un CLCG sécurisé combine deux SLCG. Ainsi, un LCG securisé dans le domaine chiffré de Damgärd-Jurik est tel que

$$E_{K_p}[X_n + 1; r_n + 1] = E_{K_p}[X_n, r_n]^a E_{K_p}[c, r_c] = E_{K_p}[aX_n + c, r_n^a r_c] \tag{5.27}$$

où X_0 représente la graine, a est le multiplicateur et c est l'incrément. La relation récursive qui existe entre les entiers aléatoires r_n est la même que celle pour le cryptosystème de Paillier, c'est-à-dire $r_{n+1} = r_n^a r_c$ où r_c et r_0 sont les aléas utilisées pour chiffrer l'incrément c et la graine X_0, respectivement.

V.2 Cryptosystème BGV

Le cryptosystème BGV appartient à la famille FHE et sa sécurité est basée sur RLWE (Ring learning with errors) [177]. Avec ce cryptosystème, les données en clairs et chiffrées prennent la forme de polynômes. Considérons un anneau $R = \mathbb{Z}[x]/f(x)$, où $f(x) = x^d + 1$ est un polynôme cyclotomique et d est une puissance de 2. En passant du domaine en clair $R_t = \mathbb{Z}_t[x]/f(x)$ au domaine chiffré $R_q = \mathbb{Z}_q[x]/f(x)$ où q est un nombre premier tel que $q = 1 \mod 2d$. R_q peut être vu comme le domaine de tous les polynômes de degré d sur $\mathbb{Z}_q$.

Les opérations dans R_q sont les additions et les multiplications modulo $f(x)$. Les coefficients des polynômes dans R_q appartiennent à $]-q/2, q/2]$. L'espace des messages en clair est $R_t = \mathbb{Z}_t[x]/f(x)$, avec $t < q$. Les clés du cryptosystème BGV (K_p, K_s) sont dérivées d'une distribution d'erreur gaussienne centrée en zéro $\mathcal{N} = N(0, \sigma)$ et d'écart-type σ sur R. Les clés de ce cryptosystème sont définies comme étant : $K_p = (\alpha_0 = (\alpha_1 s + te), \alpha_1)$ et $K_s = s$ où e et s sont deux éléments choisis aléatoirement dans $\mathcal{N}$ et α_1 est un élément aléatoire de R_q. C'est la sélection des paramètres t, q, d et σ qui garantit la sécurité de ce schéma, et également, que le chiffrement et le déchiffrement sont corrects.

Le chiffrement d'un message $m \in R_t$ (un polynôme de degré d et de coefficients dans $\mathbb{Z}_t$) est donné par

$$c = (c_0, c_1) = (\alpha_0 u + tg + m, \alpha_1 u + tf) = E[m, u, g, t, f] \tag{5.28}$$

où u, f et g sont des éléments aléatoires de $\mathcal{N}$

Le déchiffrement d'un message chiffré $c = (c_0, c_1)$

$$m = (c_0 + sc_1 \mod q) \mod t. \tag{5.29}$$

Dans le cas où $c = (c_0, c_1)$ et $c' = (c'_0, c'_1)$ sont les versions chiffrées de deux polynômes m et m', les propriétés homomorphes de BGV sont telles que

$$C_{add} = (c_0 + c'_0, c_1 + c'_1) \tag{5.30}$$
$$C_{mul} = (c_0c'_0, c_0c'_1 + c_1c'_0, c_1c'_1) = (c''_0, c''_1, c''_2) \tag{5.31}$$

Afin de réduire le nombre de composantes de la multiplication, il suffit d'utiliser la procédure de la re-linéarisation décrite dans [178].

- Calcul de la différence entre données chiffrées

La différence d entre deux messages m et m' chiffrés en c et c' sous le même paramétrage (i.e. les mêmes valeurs de u, t, g, f et α_1) $c = (c_0, c_1) = (\alpha_0 u + tg + m, \alpha_1 u + tf)$ et $c' = (c'_0, c'_1\alpha_1 u + tf)$ est donné comme :

$$d = c_0 - c'_0 \mod t = m - m' \mod t \tag{5.32}$$

- Générateur congruentiel linéaire combiné sécurisé.

Comme précédemment, nous nous en tiendrons à sécuriser un générateur LCG. Cela peut se faire en profitant des propriétés d'homomorphie du cryptrosystème BGV comme suit :

$$E[X_{n+1}, g_{n+1}, u_{n+1}, t, f_{n+1}] = E[X_n, g_n, u_n, t, f_n].a + E[c, g_c, u_c, t, f_c] \tag{5.33}$$

où le module du LCG est égal au paramètre de la clé publique t. La relation entre les aléas du LCG est donnée par :

$$[g_{n+1}, u_{n+1}, f_{n+1}] = [ag_n + g_c, au_n + u_c, af_n + f_c] \tag{5.34}$$

VI Traçabilité dans le domaine chiffré

Comme évoqué dans le chapitre 1 section III.2, le tatouage ou le marquage de données offre une solution complémentaire au chiffrement de données. Son objectif est de protéger le document (image, vidéo, musique ...) dans lequel le message est dissimulé à des fins, par exemple, de protection de la propriété intellectuelle. Pour les images, le message est inséré par modification aussi imperceptible que possible des niveaux de gris de l'image. Le signal de différence entre l'image originale et sa version tatouée est ce que l'on appelle la " marque" qui est associée au message tatoué. Telle que définit, la protection assurée par le tatouage est indépendante du format de stockage des données (la protection est dans la donnée elle-même). C'est une protection *a posteriori*, la donnée peut être utilisée tout en étant protégée par la marque. Par définition, si le tatouage ne permet pas d'assurer la confidentialité des données, il offre cependant des services de sécurité comme le contrôle d'intégrité et la traçabilité. C'est une solution intéressante dans le cas de l'externalisation de données. Par exemple, avant d'externaliser les données, l'utilisateur peut les tatouer avec l'identifiant du fournisseur de cloud à qui il confie ses données. Il pourra par la suite l'identifier s'il retrouve ses données distribuées de manière illégale

Aujourd'hui, il y a un intérêt à pouvoir combiner le tatouage avec le chiffrement de données pour accéder à des services de tatouage à partir de données chiffrées [6]. Nous présentons ici une approche de tatouage de données chiffrées de manière homomorphe compatible avec les solutions de CBIR et de machine sécurisées. Avant de rentrer dans les détails de celle-ci, nous revenons sur les objectifs du tatouage dans le domaine chiffré et les principes d'un schéma de tatouage dans le domaine en clair sur la base de la modulation par quantification d'index (QIM - "Quantization Index Modulation").

VI.1 Objectifs de tatouage

Les services de sécurité que le tatouage peut proposer dans notre contexte, seul ou de manière combinée, sont les suivants :

— **Le contrôle d'intégrité** : Ici, le message tatoué sert à détecter des modifications des données. Ce message caché peut être une signature numérique de l'image ou un résumé de celle-ci. Au moment de la vérification, le système compare le message tatoué au message recalculé. Toutes différences indiquera une perte d'intégrité de l'image voire la position des modifications, les parties de l'image encore exploitables, la nature de la modification (e.g. global ou local) et une idée de l'origine de la modification (e.g. une erreur de transmission, un acte malveillant). Pouvoir vérifier l'intégrité de données sans les déchiffrer a un sens dès lors que l'on souhaite donner accès à cette fonctionnalité à des utilisateurs non-autorisés à accéder aux données en clair.

— **La traçabilité** : Un objectif peut être de tracer une image, chiffrée ou non, au sein d'un réseau. Pour cela, chaque noeud ou utilisateur est invité à insérer son identifiant dans l'image. À tout moment, on peut alors déterminer le parcours de l'image au cours de son existence. En ce qui nous concerne, l'avantage d'insérer une marque dans une image chiffrée est une optimisation en termes de temps de traitements. Il n'est pas nécessaire de déchiffrer les données pour les tatouer. Également, cela offre à une entité non-autorisée la possibilité de tatouer une image.

Dans un environnement externalisé, tel que décrit dans la Figure 5.4, où un ensemble d'utilisateurs partage des données via plusieurs fournisseurs de cloud, supposons que l'utilisateur 5 (U5) récupère une image externalisée précédemment par l'utilisateur 1 (U1), sans en faire une demande explicite à ce dernier (i.e. U5 ne fait une requête de données à U1). Grâce au tatouage, U5 pourra connaître la provenance de l'image ainsi que les cloud où l'image a transité. Pour cela, il faut que les clouds puissent tatouer les images sous forme chiffrée. De la même manière, toutes les entités dans ce schéma d'externalisation doivent être capable de contrôler l'intégrité des données, qu'elles soient chiffrées ou non. Un message tatoué dans le chiffré doit pouvoir être lu dans le clair.

VI.2 Tatouages des images chiffrées

Plusieurs solutions pour combiner et le chiffrement de manière profiter d'une protection *a priori* (chiffrement) et *a posteriori* (le tatouage). Le tatouage peut être effectué avant l'étape du chiffrement, lors du chiffrement (tatouage/chiffrement conjoint) ou du déchiffrement (tatouage/déchiffrement conjoint), ou après le chiffrement. On pourra cependant distinguer ces méthodes en fonction de la disponibilité du message dans le domaine spatial et/ou chiffré :

- **Message disponible dans le domaine spatial (MS)** : Memon *et al* [179], propose de tatouer des images chiffrées hommomorphiquement. Le processus d'insertion est exprimé dans le chiffré, et la marque accessible uniquement lorsque l'image est déchiffrée. Dans [180], Zhang *et al.* insèrent le message dans les trois plans de bits de poids faible de l'image chiffrée à l'aide d'un stream-cipher. La détection de la marque est réalisée dans le domaine en clair, sur la base d'une mesure de corrélation spatiale.
- **Message disponible dans le domaine chiffré (MC)** : On peut trouver deux méthodes de références. La première est celle de Puech *et al* [181], où l'insertion d'un bit du message se fait en substituant un LSB d'un bloc de pixels chiffrés. À la réception de l'image, la position du bit substitué étant connu, le message est lu, et le système déchiffre deux fois

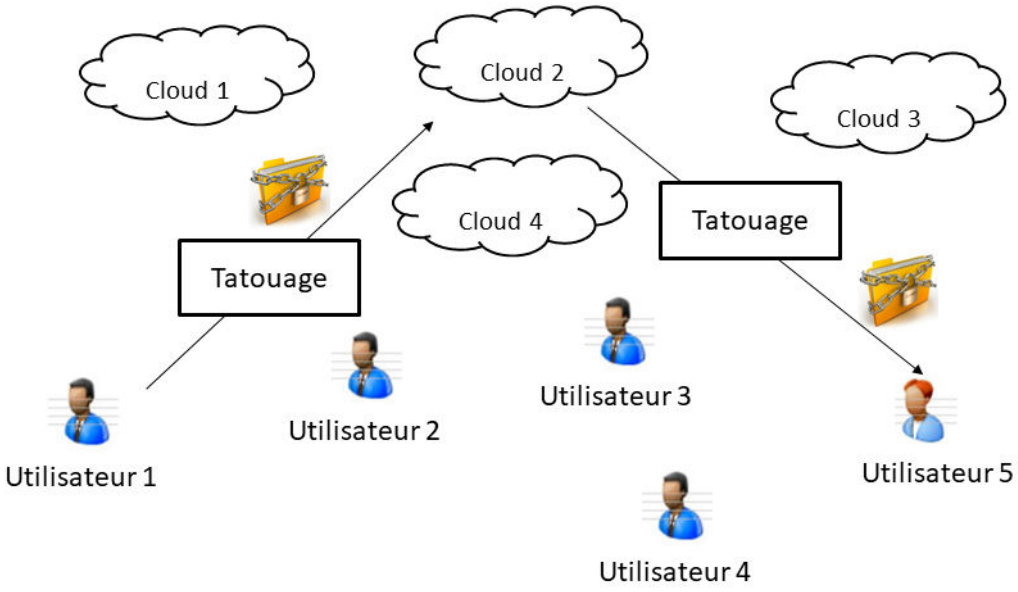


FIGURE 5.4 – Exemple d'application du tatouage des images chiffrées

le bloc en testant les deux valeurs possible du bit original. Le bloc déchiffré avec l'écart type le plus faible correspond au bloc de pixel orignal. Le deuxième schéma de tatouage a été proposé par Zhang *et al.* dans [182] qui consiste à faire une compression sans perte des LSBs de l'image chiffrée et d'insérer le message dans le gain d'espace obtenu. Là aussi, la lecture du message n'est possible que dans le domaine chiffré.

- **Message disponible dans les domaines spatial et chiffré (MSC)** : Ces solutions sont fondées sur le chiffrement partiel [183], sur du chiffrement "invariant" [184] ou sur l'insertion d'une pré-marque [185]. Dans le cas des deux premières techniques, seule une partie de l'image sont chiffrées, laissant l'autre partie disponible pour l'insertion du message. Dans [186], l'image est chiffrée en permutant la position des pixels. Les niveaux de gris de l'image sont en clair et les pixels peuvent être tatoués. D'autres stratégies [187, 188] consiste à appliquer un tatouage réversible de l'image pour libérer un espace d'insertion qui pourra être exploité pour rendre un message accessible dans les deux domaines. Le problème de ces approches est qu'elles impliquent une réorganisation des données (i.e. des pixels avant le chiffrement). C'est le cas de [187] qui compresse les bits de poids faible de l'image, et qui place les bits "libres" en début de flux avant l"opération de chiffrement par un stream cipher de type RC4. Ces bits peuvent être modifiés sans aucuns problèmes pour permettre l'insertion du message.

  Plus récemment, Bouslimi *et al.* [189] ont proposé d'utiliser une pré-marque. L'idée est d'insérer dans l'image une pré-marque, un message qui ne tient que des zéros, par exemple, avant de la chiffrer. L'image chiffrée peut être tatouée classiquement. Si le message dans le chiffré peut alors être lu facilement, ce sont les erreurs de déchiffrement sur la pré-marque qui vont permettre la lecture du message dans le domaine spatial ou en clair. Cette solution ne nécessite pas la réorganisation des pixels de l'image et a été implémentée avec des algorithmes de chiffrement à flot. Dans [189], elle a été implémentée avec la modulation de tatouage par substitution de bits de poids faible, tant pour l'insertion de la pré-marque

dans le domaine spatial que pour l'insertion du message dans le domaine chiffré, couplée avec l'algorithme du chiffrement à flot RC4. Dans ce travail nous proposons d'étendre cette approche dans le cadre de données chiffrées à l'aide du chiffrement homomorphe et de la modulation par quantification d'indice, la QIM, qui est une généralisation de la méthode de substitution des LSB.

VI.3 La QIM

Dans ce travail, nous avons fait le choix d'utiliser une modulation de tatouage substitutive qui est : la modulation par quantification d'index (QIM pour "Quantization Index Modulation") introduite en 2000 par CHEN et WORNELL. Son principe est le suivant :

Soit $I = \{P_i\}_{1 \leqslant i \leqslant N}$ une image en niveaux de gris de N pixels p_i et $M = (b'_i)_{1 \leqslant i \leqslant s}$ un message binaire de longueur s. Ici, la dynamique J des pixels correspond à l'intervalle $[0, d]$ où d est le niveau de gris max que peut prendre un pixel. Pour des pixels codés sur x bits, $d = 2^x - 1$.

Comme décrit dans la Figure 5.5, lorsqu'elle est appliquée aux pixels de l'image, le principe de la QIM consiste : i) subdiviser la dynamique J des pixels en des sous-intervalles J_i disjoints et de même largeur Δ (Δ est le paramètre "pas de quantification" de la QIM) :

$$\begin{cases} J = \cup_{0 \leqslant i \leqslant \frac{d}{\Delta} - 1} J_i \\ J_i \cap J_j \neq \emptyset \forall i \neq j \end{cases} \tag{5.35}$$

et d'attribuer les valeurs binaires 0 aux intervalles J_{2i} et la valeur 1 aux intervalles J_{2i+1} pour tout $i = 0, ..., \frac{d}{2\Delta} - 1$. Cela revient en fait à construire deux dictionnaires D_0 et D_1 qui permettront de tatouer les pixels et qui définis tels que :

$$D_0 = \{c_0 + 2k\Delta / k = 0..., \frac{d}{2\Delta}\} \tag{5.36}$$

$$D_1 = \{c_1 + 2k\Delta / k = 0..., \frac{d}{2\Delta}\} \tag{5.37}$$

avec c_i est le centre de J_i.

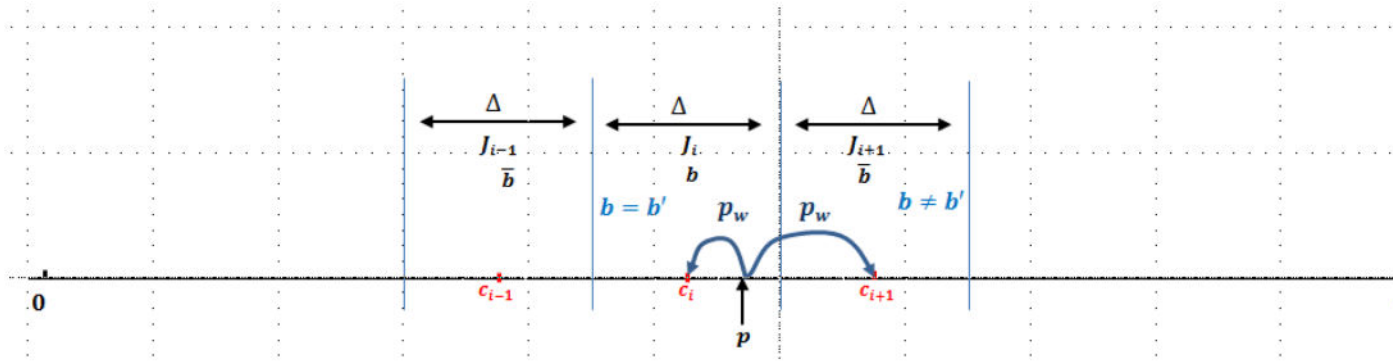


FIGURE 5.5 – Exemple d'insertion par la QIM. p_w représente p tatoué

Illustration L'insertion d'un bit b' dans un pixel p se fait en remplaçant p par p_w qui est la valeur la plus proche de p dans le dictionnaire codant la valeur de b', i.e. le dictionnaire $D_{b'}$.

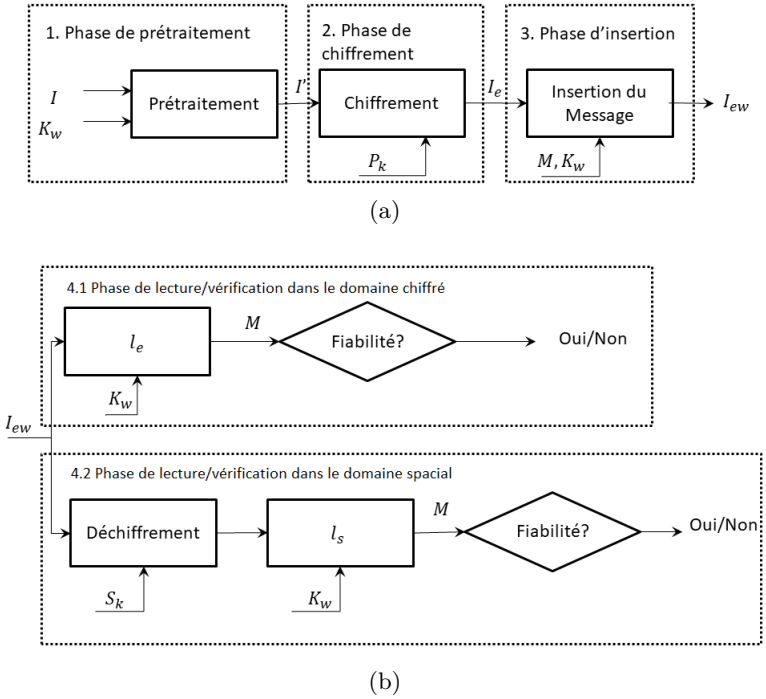


FIGURE 5.6 – Architecture du système, (a) Protection de l'image et (b) Vérification de l'image.

Dans le cas scalaire, puisque l'on travaille pixel par pixel, il n'est pas nécessaire de construire les dictionnaires. La valeur du pixel est facilement calculable : $P_w = [\frac{p}{\Delta}]\Delta + b'\Delta$.

Pour retrouver le message, en générale la valeur du bit tatoué est trouvée en déterminant à quel dictionnaire appartient le pixel tatoué p_w. À nouveau dans le cas scalaire, travaillant pixel par pixel, on trouve b' grâce à la formule : $b' = [\frac{p}{\Delta} \mod 2]$ où [] représente l'opérateur d'arrondi.

Il est important de signaler que le compromis entre la robustesse de la marque à des modifications de l'image (i.e. l'aptitude de retrouver le message après des modifications) et la distorsion de l'image est déterminée par la valeur du pas de quantification de la QIM Δ. Il est évident que des petites valeurs de Δ préserveront mieux la qualité de l'image tatouée. Cependant, la robustesse ne sera pas importante, car il suffit d'une modification faible pour qu'un pixel tatoué change d'intervalle et engendre une erreur de lecture. Vouloir augmenter la robustesse revient donc à prendre des valeurs Δ plus grandes au détriment de la qualité de l'image. La capacité de cette approche, c'est-à-dire le nombre de bits de message insérés par pixel de l'image (exprimée en bpp), est dans le cas scalaire de 1 bpp.

Comme on peut le voir, la QIM peut être exprimée sous d'addition ou de soustraction. Il est donc possible de la mettre en oeuvre dans le domaine chiffré sur la base d'un simple chiffrement homomorphe additif comme le cryptosystème de Paillier.

VII Un système de tatouage de données chiffrées

Dans un premier temps, nous décrivons l'architecture de notre système en donnant les principales phases qui le constituent, puis nous détaillons leur mise en oeuvre dans le domaine chiffré.

VII.0.1 Principe du système

Comme illustré en Figure 5.6, notre système fonctionne en quatre phases :

1. Phase de pré-traitement.
2. Phase de chiffrement.
3. Phase d'insertion.
4. Phase de lecture/vérification du message.

Dans la première phase, on va insérer une pré-marque prédéfinie W dans l'image à protéger I. W correspond à une séquence binaire générée aléatoirement en fonction de la clé de tatouage K_w et l'image résultante est I', elle est ensuite chiffrée à l'aide du cryptosysteme de Paillier sous la clé publique K_p pour obtenir l'image I_e. La phase de chiffrement et de pré-traitement sont exprimées par les équations ci-dessous :

$$I' = f_s(I, K_w) \tag{5.38}$$
$$I_e = E[I'] \tag{5.39}$$

où f_s est la fonction de pré-marquage et $E[]$ est la fonction de chiffrement de Paillier.

La phase de marquage dans le domaine chiffré correspond à l'insertion du message M dans l'image chiffrée. L'insertion dans le domaine chiffré est donnée par l'équation :

$$I_{ew} = f_e(I_e, M) \tag{5.40}$$

où f_e est la fonction d'insertion dans le domaine chiffré.

La dernière phase est la phase de l'extraction ou la lecture du message M. Ce dernier est accessible dans le domaine chiffré et spatial à la fois, c'est-à-dire, on peut extraire le message M à partir de I_{ew} ou de sa version déchiffrée :

$$M = l_e(I_{ew}, K_w) = l_s(D[I_{ew}], K_w) \tag{5.41}$$

où l_e (resp. l_s) est la fonction d'extraction dans le domaine chiffré (resp. en clair) et $D[]$ est la fonction de déchiffrement de Paillier.

Nous détaillons ces différentes phases ci-après en décrivant leur mise en oeuvre avec la QIM

VII.0.2 Tatouage de données chiffrées homomorphiquement

Phase de pré-traitement Cette phase permet d'insérer la pré-marque W une suite binaire générée à partir d'un générateur pseudo-aléatoire PRNG et d'une clé secrète K_w. W est insérée dans l'image I en utilisant la QIM de la manière suivante :

L'image I est secrètement partitionnée en sous-blocs $I_j = \{P_j^1, P_j^2, ..., P_j^s\}$ de s pixels. La pré-marque $W = \{w_i\}_{1 \leq i \leq s}$ est une séquence binaire de meme taille s et uniformément distribuée est insérée dans chaque sous bloc. Plus clairement, w_i est inséré dans P_j^i comme suit :

1. On calcule $[\frac{P_j^i}{\Delta}]$ et $[\frac{P_j^i}{\Delta}] \mod 2$ pour respectivement déterminer à quel sous-intervalle J_k appartient P_j^i et la valeur binaire b codée par J_k(i.e $b = [\frac{P_j^i}{\Delta}] \mod 2$)
2. Pour insérer w_i :
   — Si $w_i = b$: P_j^i est dans le bon intervalle, la valeur du pixel tatoué prendra la valeur $P_{jw}^i = [\frac{P_j^i}{\Delta}].\Delta + c_0$

— Si $w_i \neq b$: il change de sous-intervalle, la valeur du pixel tatoué est telle que : $P^i_{jw} = [\frac{P^i_j}{\Delta}].\Delta \pm c_1$.

Phase de chiffrement L'image pré-tatouée $I' = \{P_i\}_{1 \leq i \leq taille}$ est ensuite chiffrée par le cryptosystème de Paillier pixel par pixel, par le biais de l'algorithme suivant :

Algorithm 4 Algorithme de chiffrement

Entrée : une image en clair $I = (P_i)_{1 \leq i \leq taille}$, une clé publique K_p
Sortie : une image chiffrée $I_e = (P_{ei})_{1 \leq i \leq taille}$

1. Choisir un nombre $r \in \mathbb{Z}^*_{K_p}$ aléatoirement.
2. **For** i=1 to *taille* **do**
3. $P_{ei} = g^{P_i} r^{K_p} \mod K_p^2$
4. $r = rand(\mathbb{Z}^*_{K_p})$
5. **End for**

Phase de marquage Soit $M = \{M_j\}_{1 \leq j \leq N}$ un message binaire de longueur N et I_e l'image chiffrée. En fonction de la sensibilité du contenu du message M, il peut être chiffré avant d'être enfoui (e.g. identifiant du patient).

Soit également : $I_{ej} = \{C^1_j, C^2_j, ..., C^s_j\}$ un sous bloc de l'image I_e de taille s, avec C^i_j est le chiffré de P^i_j. Ces blocs correspondent au même partitionnement considéré lors du pré-marquage. L'insertion d'un bit M_j dans le bloc I_{ej} est réalisée comme suit :

— Si $M_j = 0$, $\forall j = 1, ..., N$
  — Si $LSB(C^i_j) = 0$, alors C^i_j est tatoué en $C^i_{wj} = C^i_j$.
  — Si $LSB(C^i_j) = 0$, alors C^i_j doit être modifié de manière à ce que $LSB(C^i_j) = 1$. Pour ce faire, nous proposons une procédure itérative où C^i_j est multiplié par le chiffré de '0' avec un aléa r variable (i.e. $E[0, r]$) ; procédure qui s'arrête dès que la condition LSB est réalisée.

  On pourra remarquer qu'en multipliant les pixels chiffrés avec le chiffré de '0', la pré-marque W n'est pas modifiée dans le domaine spatial du fait des propriétés du cryptosystème de Paillier (i.e. $C^i_j E[0, r] = E[P^i_j, r'].E[0, r] = E[P^i_j, rr']$). Sur cette base, la pré-marque étant inchangée, on identifiera l'insertion d'un bit '0' dans le bloc de pixels dans le domaine spatial.

— Si $M_j = 1$, Dans ce cas, il faut à la fois tatouer ce bit dans le bloc chiffré et perturber la pré-marque également insérée. Nous allons suivre la procédure que précédemment pour changer les LSB dans le domaine chiffré, mais en multipliant C^i_j par $E[\Delta, r]$ ce qui aura pour effet de changer automatiquement l'intervalle d'appartenance des niveaux de gris des pixels dans le domaine spatial. L'opération est dans ce cas la suivante :
  — On calcule $b = C^i_j.E[\Delta, r]$
  — Si $LSB(b) = 1$, alors on tatoue C^i_j en b.
  — Si $LSB(b) = 0$, on répète la même procédure itérative que précédemment mais en multipliant b par $E[0, r]$ jusqu'à ce que $LSB(b) = 1$.

À l'issue de cette procédure de marquage, le bit M_j du message M est codé dans tous les LSB des chiffrés du sous-bloc. Cet algorithme permet d'insérer un bit du message dans un sous-bloc de taille s et donc d'obtenir une capacité d'insertion de $\frac{1}{s}bpp$ dans le chiffré.Cette capacité peut être améliorée en réduisant la taille des sous-blocs. Pour cet algorithme, il est toujours possible d'insérer des messages quelque soit la valeur de s contrairement à la méthode proposée par Bouslimi *et al* [189] qui impose que la valeur de s soit supérieur à 10 pour réussir l'insertion.

VII.0.3 Extraction du message dans les deux domaines

Comme nous l'avons dit, le message M doit être accessible que l'image soit chiffrée ou en clair, sur la base de la connaissance de la clé de tatouage K_w qui vient paramétrer les deux fonctions de lecture de message I_e et I_s qui fonctionnent respectivement dans le domaine chiffré et le domaine en clair, c-à-d :

$$M = l_e(I_{ew}, K_w) = l_s(D[I_{ew}], K_w) \tag{5.42}$$

Les définitions de ces deux fonctions dépendent des modulations de tatouage utilisées dans les deux domaines et donnent accès au message indépendamment des clés de chiffrement.

Dans le domaine chiffré Toute personne qui veut extraire les données contenues dans cette image doit disposer de la clé de tatouage secrète K_w qui dans notre cas correspond à la taille $s$des sous-blocs et au partitionnement secret de l'image en sous-blocs I_{ej} de taille s, la fonction de lecture dans le domaine chiffré l_e est appliquée à chaque sous-bloc I_{ej} pour en extraire la valeur du $j^{ème}$ du message, i.e M_j. Soit le sous-bloc $I_{ej} = \{C^1_{wj}, ..., C^s_{wj}\}$, sur la base de la procédure d'insertion dans le chiffré, où on force tous les LSB du sous bloc chiffré à la valeur de M_j, la fonction de lecture l_e est telle que :

$$\hat{M}_j = l_e(I_{ej}, K_w) = \begin{cases} 1 & si \quad A_j > B_j \\ 0 & sinon \end{cases} \tag{5.43}$$

où, $\hat{M}_j$ est la valeur du $j^{ème}$ bit du message extrait du $j^{ème}$ sous-bloc chiffré, et A_j et B_j sont tels que

$$A_j = card\{C^i_{wj}, LSB(C^i_{wj}) = 1, \forall i = 1, ..., s\} \tag{5.44}$$

$$B_j = card\{C^i_{wj}, LSB(C^i_{wj}) = 0, \forall i = 1, ..., s\} \tag{5.45}$$

Dans le domaine en clair Une fois l'image chiffrée-tatouée I_{ew} déchiffrée à l'aide du crypto-système de Paillier sous la clé privée K_s, la lecture du message dans le domaine spatiale s'appuie sur la procédure suivante :

1. A l'aide de la clé secrète de tatouage K_s, l'utilisateur génère la pré-marque W.
2. L'image tatouée I_w est secrètement partitionnée sur la base de la clé K_w.
3. L'extraction du bit M_j du $j^{ème}$ sous-bloc de pixels s'appuie sur la fonction l_s qui consiste simplement à comparer la marque tatouée par QIM dans I_{wj}, i.e. W', à la pré-marque W. Ainsi, la valeur de M_j est donnée par l_s comme suit :

$$M_j = l_s(I_j, K_w) = \begin{cases} 1 & si \quad W = W' \\ 0 & sinon \end{cases} \tag{5.46}$$

VIII Résultats expérimentaux

Le schéma précédent a été testé sur une base de test constituée de 100 images d'échographie de 8 bits de profondeur et de 576×688 pixels.

VIII.1 Critère de performance

Les paramètres de performances considérés pour évaluer notre système sont : la capacité d'insertion, la robustesse de la marque à des modifications de l'image, le niveau de qualité de l'image tatouée et la complexité de calcul.

- **Distorsion** - Comme notre algorithme de tatouage introduit en moyenne la même distorsion dans chaque bloc de pixels, nous avons décidé d'utiliser le rapport signal sur bruit pic (PSNR - Peak Signal to Noise Ratio) pour évaluer le niveau d'imperceptibilité de la marque ou encore mesurer la distorsion entre l'image I et sa version tatouée et déchiffrée I_w :

$$PSNR = 10log_{10}(\frac{d^2}{EQM}) \tag{5.47}$$

  où, d est le niveau de gris max de l'image ($d = 2^x - 1$ dans le cas de pixels codés sur x bits) et EQM est l'erreur quadratique moyenne entre l'image originale et l'image tatouée. Pour des images de $(n \times m)$, l'EQM est telle que :

$$EQM(I_1, I_2) = \frac{1}{mn} \sum_{1 \leqslant i \leqslant n} \sum_{1 \leqslant j \leqslant m} (I_1(i,j) - I_2(i,j))^2 \tag{5.48}$$

  Cette mesure est pertinente dans notre cas, car elle suppose une distorsion de l'image constante sur toute l'image. C'est le cas de notre schéma. À noter que la valeur de PSNR est une valeur qui décroit avec la distorsion de l'image. Elle sera d'autant plus grande que l'image tatouée diffère peu de l'image originale.

- **La capacité d'insertion :** Elle correspond au taux d'insertion exprimé en nombre de bits de message enfoui par pixel de l'image (*bpp* pour "bit per pixel"). C'est un indicateur sur la taille du message qu'on peut insérer dans une image.
- **La robustesse :** Un tatouage est dit robuste si après toute modification de l'image tatouée, la marque ou le message dissimulé peut être extrait. A l'opposé de la robustesse on trouve la fragilité, une propriété utile pour détecter une modification des données, la marque disparaitra. Dans ces travaux nous avons considéré la robustesse du message tatoué aux attaques par compression et déformation de l'image, qui sont des traitements d'images assez classiques. Il faut donc pouvoir lire le message après leur application
- **Complexité de calcul :** Dans ces tests, elle est évaluée par le temps d'exécution de l'algorithme d'insertion dans le domaine chiffré qui est le plus coûteux par rapports aux traitements réalisés dans le clair.

VIII.2 Résultats obtenus

— Capacité d'insertion : Comme un bit de message est inséré dans un sous-ensemble de pixels, la capacité que l'on peut insérer dans une image dépend de la dimension des sous-ensembles de pixels et de l'image. En effet, la capacité atteinte est de $1/s$ *bpp*. En choisissant $s = 1$, cela conduit à une capacité d'insertion de 1 *bpp* ou de manière équivalente à un message d'environ 396 Kbits pour nos images test. Cette capacité est suffisamment grande pour

l'insertion de différents attributs de sécurité permettant de garantir la fiabilité de l'image. M peut contenir un code d'authenticité (environ 1000 bits en combinant l'identificateur national français avec l'identificateur unique de DICOM, la norme pour les images médicales [190]) et une preuve d'intégrité qui peut être une séquence binaire pseudo aléatoire. L'intégrité comme une sequence binaire pseudo aléatoire [189] dont la présence ou l'absence est la preuve de l'intégrité ou non de l'image chiffrée-tatouée.

— Robustesse : La robustesse de notre schéma de marquage dépend du pas de quantification (Δ) de la QIM. Plus Δ est grand plus la QIM est robuste. Cependant, du fait que nous travaillons pixel par pixel, le niveau de robustesse reste faible comparer à des techniques qui travaillent dans des domaines transformés (i.e. application de la QIM à des coefficients d'ondelette, de la transformée en cosinus discrète) et dont la mise en oeuvre à pour conséquence de coder ou « d'étaler » 1 bit du message sur plusieurs pixels. À noter également, qu'il est peut être possible de gagner en robustesse en utilisant un codage par répétition, en travaillant avec $s > 2$, permettant de répéter le message au moins 3 fois.

— Distorsion : La borne inférieure du PSNR de notre schéma peut être déterminée théoriquement en fonction du pas de quantification de la QIM Δ. Supposons que les pixels de l'image sont uniformément répartis sur les centres de dictionnaire de la QIM (voir Figure 5.5). Cela signifie que la probabilité qu'un pixel d'un sous-ensemble P_j^i appartienne à une intervalle qui code '0' (resp. '1') est 0.5. Puisque W est une suite binaire uniformément distribuée, la probabilité que le pixel P_j^i appartienne à l'intervalle qui code w_i est 0.5. Comme on peut le voir en Figure 5.7, les distorsions maximales que l'on peut introduire pour insérer w_i dans P_j^i en le déplaçant au centre le plus proche qui code w_j sont $\Delta/4$ et $\Delta/2$ dans les cas où P_j^i encode w_j ou non, respectivement. Sachant cela, on peut considérer que la distorsion maximale induite par le processus d'insertion de w_j dans P_j^i est donc $d_{ms} = \frac{1}{2}(\frac{\Delta}{2} + \frac{\Delta}{4}) = \frac{3}{8}\Delta$.
Après l'insertion de M dans l'image chiffrée, seuls les sous-ensembles codant $b_i =' 1'$ sont modifiés. Du fait que la distorsion induite par l'insertion de $b_i =' 1''$ dans P_j^i est $\frac{\Delta}{2}$ et que la probabilité que $b_i =' 1'$ est 0.5, la distorsion induite par l'insertion de b_i dans P_i^{je} est alors $d_{me} = \frac{\Delta}{4}$. Par conséquent, la distorsion maximale induite par pixel à notre schéma de tatouage est $d_m = d_{ms} + d_{me} = \frac{5}{8}\Delta$. Dès lors, la limite inférieur du PSNR peut être précisé.

$$PSNR_{Pail}(I, I_{wd}) \geq 20 \log_{10}(\frac{408}{\Delta}) \tag{5.49}$$

Nous donnons en Figure 5.7 la variation de cette limite pour différentes valeurs de Δ. En pratique, les valeurs PSNR obtenues sont beaucoup plus grandes. Elles sont respectivement environ $51,15$ dB, $44,2$ dB et $37,8$ dB pour $\Delta = [2, 4, 8]$. Ceci peut s'expliquer par le fait qu'en pratique, les pixels de l'image originale ne sont pas uniformément répartis sur les centres du dictionnaire de la QIM. Avec notre système, une perte d'information se produit mais elle reste faible lorsque $\Delta < 5$. Pour les images d'échographie, Chen *et al.* ont rapporté dans [190] qu'une perte d'information peut être tolérée dans la mesure où le PSNR reste dans l'intervalle de laveurs 40 et 50 dB.

— Complexité de calcul : Comme nous l'avons vu en section VII, l'insertion d'un bit de message dans un pixel chiffré s'appuie sur une procédure itérative. Elle multiplie le pixel chiffré par le chiffré de '0', avec un aléa r que l'on fait varier jusqu'à ce que le LSB de la multiplication des chiffrés corresponde au bit du message. Les opérations effectuées à chaque itération sont donc un chiffrement de la valeur suivi d'une multiplication entre chiffrés. Considérant, la complexité de la multiplication négligeable devant celle du chiffrement

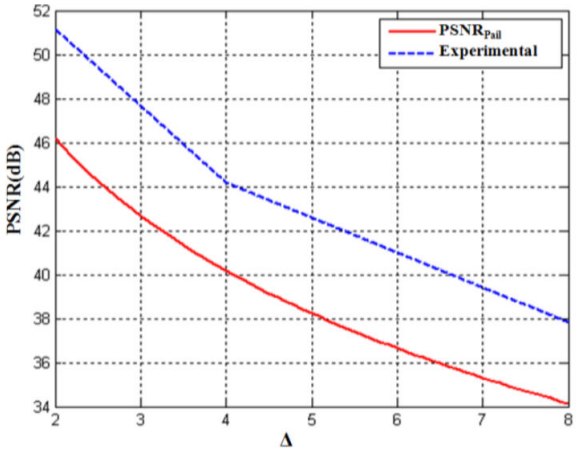


FIGURE 5.7 – Taux du PSNR inférieur théorique ($PSNR_{Pail}$) et les valeurs de PSNR expérimentales obtenues pour différentes valeurs de Δ.

d'une donnée par le cryptosystème de Paillier, et que le nombre d'itération moyen est de 1, la complexité de calcul de l'étape d'insertion dans le chiffré est O(1) chiffrement. Plus simplement, le temps de tatouage d'un pixel est équivalent à son temps de chiffrement.

VIII.3 Conclusion

Nous avons proposé une méthode de tatouage qui permet l'insertion et l'extraction d'une marque ou d'un message dans le domaine en clair et chiffré qui peut permettre de vérifier l'intégrité de données externalisées par des utilisateurs autorisés et non-autorisés à accéder au contenu de l'image.

Si la solution proposée profite du cryptosystème homomorphe additif de Paillier et de ses propriétés d'homomorphie pour implémenter la modulation de tatouage substitutive QIM dans le domaine chiffré, elle peut être mise en oeuvre avec tous cryptosystèmes possèdant les propriétés suivante : 1) Au moins homomorphiquement additif afin de perturber la prémarque W, 2) Probabiliste ou sémantiquement sûr pour donner la possibilité de moidifier le LSB du message chiffré.

La solution proposée offre une capacité d'insertion plus élevée que celle fournie par la méthode de Bouslimi *et al.* [189] qui combine le RC4 et la modulation de tatouage par substitution de bits de poids faible. Avec notre solution, il n'y a pas d'incertitude quant au résultat de l'opération de marquage d'un bit dans un sous-bloc de taille s . En effet, avec la solution de [189], cette probabilité est non nulle et n'est négligeable que pour des valeurs de $s \geq 10$. Avec notre, nous pouvons travailler avec de sous-blocs de taille unitaire.

IX Conclusion

Dans ce chapitre, nous avons abordé le problème du partage de données de plusieurs utilisateurs qui est nécessaire au déploiement de nos approches de CBIR et de machine learning sécurisées et également celui de la protection et de la fiabilité des données externalisées qu'elles soient chiffrées ou non.

Pour permettre le partage et le traitement de données externalisées sous forme chiffrées sous des clés différentes, nous avons proposé le premier schéma PRE homomorphe (HPRE). Il s'agit

d'un nouveau concept de PRE qui diffère du schéma PRE classique qui tire son originalité de deux fonctions D_{iff} et *GenRand* qui repose respectivement, sur la solution proposée dans le chapitre 2 section III et qui permet de calculer la différence entre des données chiffrées et, d'autre part, sur un générateur congruentiel linéaire combiné sécurisé (SCLCG) que nous avons implémenté dans le domaine chiffré. Ces deux fonctions permettent de réduire fortement la complexité par rapport aux solutions actuelles fondées sur le « pairing » (le couplage a une complexité de calcul très élevée par rapport aux opérations modulaires [173]). Nous avons proposé deux solutions. Si la première a besoin d'une phase de rafraichissement du bruit qui augmente la complexité de calcul du délégué. Dans la seconde approche, la totalité des opérations de partage sont réalisées par le cloud. Comme les données sont chiffrées de façon homomorphe, il est possible de traiter ces dernières avant et après le partage en tout assurant leur confidentialité. Quelque soit la solution proposée, toutes assurent les propriétés importantes d'un schéma PRE. Elles sont ainsi unidirectionnelles et « collusion resistant ». Notre HPRE a été implémenté dans le cas du partage d'images non compressées stockées dans le cloud montrant de bonnes performances en termes de temps de calcul. Notre système n'est pas limité aux images et peut être utilisé avec n'importe quel type de données. Nous avons aussi montré que le principe de notre système HPRE peut être généralisé à d'autres schémas de chiffrement homomorphe (« somewhat » et « fully-homomorphe »).

En ce qui concerne le tatouage de données chiffrées homomorphiquement, nous avons proposé une solution qui permet d'insérer dans une image chiffrée un message accessible par la suite dans le domaine en clair et chiffré. L'intérêt est de pouvoir vérifier l'intégrité de données externalisées par des utilisateurs autorisés et non-autorisés sans que ces derniers n'accèdent au contenu de l'image. Si la solution proposée profite du cryptosystème homomorphe additif de Paillier et de ses propriétés d'homomorphie pour implémenter la modulation de tatouage substitutive QIM dans le domaine chiffré, elle peut être mise en oeuvre avec tous les cryptosystèmes possédant les propriétés suivantes : i) probabiliste ou sémantiquemnt sûr ; ii) au moins homomorphiquement additif. La solution proposée offre une capacité d'insertion plus élevée que celle fournie par les méthodes similaires actuelles.

Conclusion générale

L'imagerie médicale évolue aujourd'hui vers le « medical cloud imaging ». Au-delà de la réduction des coûts de maintenance et des services, l'intérêt est aussi de faciliter et d'améliorer la prise en charge des patients par le biais de l'externalisation des données. Sur la base de cette mutualisation des données, il est possible de développer de nouveaux services comme des outils d'aide au diagnostic fondés sur la réutilisation des données d'imagerie de patients à l'aide techniques de recherche d'images par le contenu (CBIR) ou d'apprentissage automatique (ML). Cependant, le déploiement de ces techniques dans un environnement ouvert (i.e. internet) et sous le contrôle de fournisseurs de cloud soulève de nombreuses questions en matière de sécurité ; des besoins imposés par un cadre législatif et déontologique très stricte (voir Chapitre 1). La confidentialité des données et le respect du droit à la vie privée sont essentiels ! L'intégrité et la traçabilité des données ne sont pas non plus à négliger. Dans l'externalisation, l'utilisateur perd le contrôle sur ses données, et sa confiance dans un fournisseur de service doit rester toute relative. Nous avons vu qu'un modèle de cloud « semi-honnête » ou « honnête mais curieux » était dans ce cas approprié. Dans ce contexte, les données ne peuvent externalisées que sous forme chiffrée. Pour pouvoir réutiliser les données, le traitement et le partage de données chiffrées sont des éléments clés. Ce sont là les principaux de ces travaux de thèse.

En ce qui concerne les techniques de CBIR externalisée sécurisée, nous avons vu que les solutions existantes nécessitent beaucoup de communications avec entre plusieurs fournisseurs de cloud voire avec l'utilisateur à l'origine de la requête, et qu'il y a un intérêt pour des solutions sans communications. Notamment, ces solutions en fonctionnent plus en cas de perturbation des réseaux. Nos premiers travaux ont également mis en évidence la nécessité de sécuriser l'ensemble de la chaîne de CBIR. En effet, si les signatures extraites ne sont pas chiffrées, elles peuvent donner beaucoup d'indices à un cloud honnête mais curieux. À noter aussi qu'avec ces solutions, les données sont chiffrées avec la même clé. Il faut donc trouver des solutions originales adaptées à l'imagerie médicale tant pour l'extraction de signature que pour la réutiliser des données externalisées chiffrées par des utilisateurs différents. Cette mise en adéquation passe par la sécurisation de techniques de CBIR qui exploitent des signatures globales, qui sont adaptées à l'imagerie médicale, et qui n'ont pas été sécurisées.

Sur cette base, nous avons proposé trois approches (chapitre 2 et chapitre 3) capables d'exploiter des données chiffrées et qui permettent à un cloud de trouver des images semblables sans accéder à leur contenu réel. La première permet d'extraire une signature globale non-chiffrée à partir d'une image chiffrée sans communications supplémentaires entre l'utilisateur et le serveur d'un seul fournisseur de cloud. Pour ce faire, en profitant des propriétés du chiffrement homomorphe, nous avons proposé une nouvelle méthode de comparaison entre des données chiffrées. Cette solution est de complexité très faible par rapport aux solutions existantes (Cf. Chapitre 2). Un défaut de cette solution est qu'elle extrait d'une image chiffrée une signature en clair qui peut mettre en danger la confidentialité des données dans le cas où le cloud à une connaissance *a priori* sur les données, notamment sur les propriété statistiques des données. Pour résoudre ce problème, nous avons développé une deuxième approche qui extrait des signatures sécurisées,

i.e. chiffrées, des images à l'aide de fournisseurs de cloud indépendants, tout en évitant des communications avec l'utilisateur ou des interactions avec un tiers de confiance. La dernière solution de CBIR sécurisée proposée extrait des signatures avec un seul serveur. Son originalité s'appuie sur la combinaison d'une méthode de calcul de différences entre des données chiffrées avec un algorithme de sélection de données proche du protocole PIR (« Private Information Retrieval »). Si cette solution est « communication free », sa complexité de calcul est importante, et sa mise en œuvre en imagerie médicale limitée.

À noter que ces trois approches ont été expérimentées sur des bases de données médicales et biométriques montrant des performances identiques à la même approche CBIR dans le domaine en clair.

Par la suite, nous nous sommes intéressés à la sécurisation des techniques de machine learning avec un intérêt particulier pour les réseaux de neurones qui sont un des éléments clés des méthodes d'apprentissage profond, particulièrement d'actualité avec les données massives. L'enjeu étant de sécuriser la phase d'apprentissage et d'assurer la convergence de celle-ci lorsqu'elle est appliquée sur des données chiffrées. Avec notre approche, la confidentialité des paramètres du réseau comme les données de l'utilisateur et les réponses du réseau sont assurées. Cet objectif est atteint grâce au chiffrement homomorphe et deux fournisseurs de cloud indépendants et semi-honnêtes. Nous avons en particulier proposé une nouvelle version sécurisée de la fonction $Max(.,.)$ qui, contrairement aux solutions actuelles, retourne un résultat sous forme chiffrée. Le perceptron multi-couches que nous avons sécurisé converge notamment grâce à la gestion des « overflows » dans la fonction d'expansion. Si cette fonction permet de transformer des nombres réels en nombre entier, et de conduire les phases d'apprentissage et de classification de manière équivalente, elle est source d'augmentation de la taille des données. Pour résoudre ce problème et assurer la convergence du réseau, nous proposons d'ajouter après chaque multiplication une division par le facteur d'expansion Q. Il conviendra cependant de trouver un compromis entre facteur expansion et précision de classification en fonction du contexte applicatif, afin d'exploiter ce type d'approches (CBIR et d'apprentissage automatique) sur la base des bases de données externalisé mutualisées.

Pour pouvoir déployer ces différentes solutions de CBIR et de MLP sécurisés, il convient de pouvoir réutiliser des données issues de dossiers des patients externalisés par plusieurs médecins et aussi de tracer et de s'assurer de l'intégrité des données. Pour répondre au besoin du partage de données chiffrées sous des clés d'utilisateurs différents tout en permettant le post-traitement de données nous avons proposé un schéma de « Homomorphic Proxy Re-Encryption » (HPRE). À notre connaissance, ce schéma est le premier du genre et introduit un nouveau concept de PRE. Cet HPRE diffère du schéma PRE classique et tire son originalité de deux fonctions « D_{iff} » et « $GenRand$ » qui permettent respectivement de calculer la différence entre des données chiffrées et implémente un générateur congruentiel linéaire combiné sécurisé (SCLCG) dans le domaine chiffré. Ces deux fonctions permettent de réduire fortement la complexité par rapport aux solutions actuelles fondées sur le « pairing ». Nous avons proposé deux solutions. Si la première nécessite que le délégué effectue une opération de rafraichissement de bruit, la totalité des opérations de partage et de rafraichissement de bruit sont réalisées par le cloud dans la seconde approche. Ces solutions assurent les propriétés importantes d'un schéma PRE et sont unidirectionnelles et résistantes aux collusions. Nous avons généralisé ce HPRE à d'autres schémas de chiffrement homomorphe de type « somewhat » et « fully». Il fonctionne sur tout type de données et permet de partager une image 92×112 pixels dans presque une minute.

Pour pouvoir tracer les données ou vérifier leur intégrité lorsqu'elles sont externalisées, nous avons suggéré d'utiliser le tatouage de données qui est dans ce cas pertinent. Nous avons développé une solution qui combine le cryptosystème homomorphe additif de Paillier et la modulation

de tatouage substitutive QIM. Elle permet d'insérer dans une image chiffrée homomorphiquement un message accessible par la suite dans les domaines en clair et chiffré. Des utilisateurs non-autorisés peuvent ainsi vérifier l'intégrité des données et les tracer sans accéder à l'image en clair. Cette solution est généralisable à tout cryptosystème probabiliste et a minima homomorphiquement additif.

Néanmoins, au-delà de ces travaux un certain nombre de points restent à étudier. Nous pouvons citer :

- Le besoin de réduire la complexité de la sécurisation des méthodes de CBIR qui extraient des signatures sécurisées sans communications. Il convient en fait de réduire le nombre d'opérations et d'itérations sécurisées nécessaires pour réaliser d'une opération en clair. En effet, la pratique médicale est particulièrement exigeante quant à la rapidité d'accès aux données. C'est d'ailleurs une caractéristique des applications médicales à succès. Un praticien refusera d'utiliser une solution s'il doit attendre plusieurs minutes pour accéder à une image.
- Le développement de cryptosystèmes homormophes rapides et qui fonctionnent avec des nombre réels. Pour aller au-delà des premiers résultats de notre MLP sécurisé, son extension aux images ou à des gros volumes de données nécessite, quand bien même on arriverait à réduire la complexité du traitement sécurisé, des cryptosystèmes homomorphes rapides. Dans le domaine en clair, la phase d'apprentissage d'un MLP sur des données de grande taille peut d'ores et déjà prendre plusieurs jours. Une version sécurisée prendrait aujourd'hui des mois voire des années. Les cryptosystèmes actuellement disponibles, fonctionnent sur des nombres entiers ce qui impose des contraintes importantes, comme nous avons pu le voir dans la sécurisation d'un MLP. C'est un enjeu important qui permettrait de ne plus avoir de pertes d'information liées à la fonction d'expansion.
- Le développement de générateurs pseudo-aléatoires sécurisés qui génèrent des séquences aléatoires de nombre chiffrés. Si ce type de système est au cœur de notre HPRE, il peut servir à bon nombre d'applications. Il conviendra en particulier de s'intéresser à la rapidité d'exécution de ceux-ci.
- Concernant le tatouage de données chiffrées, nous nous sommes limités à une modulation relativement simple. Il existe des modulations plus performantes et notamment plus robustes qui travaillent dans des espaces transformées. Vis-à-vis de notre méthode de marquage, il y a besoin de trouver une solution non-itérative pour modifier le LSB d'un chiffré pour, d'une part, réduire sa complexité de calcul et, d'autre part, augmenter la capacité d'insertion dans un pixel.

Bibliographie

[1] Chao-Yung Hsu, Chun-Shien Lu, and Soo-Chang Pei. Image feature extraction in encrypted domain with privacy-preserving sift. *IEEE Transactions on Image Processing*, 21(11) :4593–4607, 2012.

[2] Reda Bellafqira, Gouenou Coatrieux, Dalel Bouslimi, and Gwénolé Quellec. Content-based image retrieval in homomorphic encryption domain. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 2944–2947. IEEE, 2015.

[3] Reda Bellafqira, Gouenou Coatrieux, Dalel Bouslimi, and Gwénolé Quellec. An end to end secure cbir over encrypted medical database. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 2537–2540. IEEE, 2016.

[4] Reda Bellafqira, Gouenou Coatrieux, Dalel Bouslimi, Gwénolé Quellec, and Michel Cozic. Secured outsourced content based image retrieval based on encrypted signatures extracted from homomorphically encrypted images. *arXiv preprint arXiv :1704.00457*, 2017.

[5] Reda Bellafqira, Gouenou Coatrieux, Dalel Bouslimi, Gwénolé Quellec, and Michel Cozic. Sharing data homomorphically encrypted with different encryption keys. *arXiv preprint arXiv :1706.01756*, 2017.

[6] Dalel Bouslimi, Reda Bellafqira, and Gouenou Coatrieux. Data hiding in homomorphically encrypted medical images for verifying their reliability in both encrypted and spatial domains. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 2496–2499. IEEE, 2016.

[7] Gwénolé Quellec, Mathieu Lamard, Guy Cazuguel, Béatrice Cochener, and Christian Roux. Wavelet optimization for content-based image retrieval in medical databases. *Medical image analysis*, 14(2) :227–241, 2010.

[8] Eliot Siegel and Bruce Reiner. Work flow redesign : the key to success when using pacs. *American Journal of Roentgenology*, 178(3) :563–566, 2002.

[9] Chia-Chi Teng, Jonathan Mitchell, Christopher Walker, Alex Swan, Cesar Davila, David Howard, and Travis Needham. A medical image archive solution in the cloud. In *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on*, pages 431–434. IEEE, 2010.

[10] William H Delone and Ephraim R McLean. The delone and mclean model of information systems success : a ten-year update. *Journal of management information systems*, 19(4) :9–30, 2003.

[11] David Aubry. *Développement et validation d'un modèle de succès du PACS dans les hôpitaux*. PhD thesis, 2005.

[12] Guy Paré, Luigi Lepanto, David Aubry, and Claude Sicotte. Toward a multidimensional assessment of picture archiving and communication system success. *International journal of technology assessment in health care*, 21(4) :471–479, 2005.

[13] George C Kagadis, Christos Kloukinas, Kevin Moore, Jim Philbin, Panagiotis Papadimitroulas, Christos Alexakos, Paul G Nagy, Dimitris Visvikis, and William R Hendee. Cloud computing in medical imaging. *Medical physics*, 40(7), 2013.

[14] James Philbin, Fred Prior, and Paul Nagy. Will the next generation of pacs be sitting on a cloud ? *Journal of digital imaging*, 24(2) :179–183, 2011.

[15] Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011.

[16] Chao-Tung Yang, Lung-Teng Chen, Wei-Li Chou, and Kuan-Chieh Wang. Implementation of a medical image file accessing system on cloud computing. In *Computational Science and Engineering (CSE), 2010 IEEE 13th International Conference on*, pages 321–326. IEEE, 2010.

[17] Chenguang He, Xiaomao Fan, and Ye Li. Toward ubiquitous healthcare services with a novel efficient cloud platform. *IEEE Transactions on Biomedical Engineering*, 60(1) :230–234, 2013.

[18] Tom White. *Hadoop : The definitive guide*. " O'Reilly Media, Inc.", 2012.

[19] Yin Zhang, Meikang Qiu, Chun-Wei Tsai, Mohammad Mehedi Hassan, and Atif Alamri. Health-cps : Healthcare cyber-physical system assisted by cloud and big data. *IEEE Systems Journal*, 11(1) :88–95, 2017.

[20] Pierre Audoin Mathieu Poujol. *Le Cloud Computing en France*. "PAC – EMC, Intel, VMware 2010.", 2010.

[21] Md Mahmudur Rahman, Prabir Bhattacharya, and Bipin C Desai. A framework for medical image retrieval using machine learning and statistical similarity matching techniques with relevance feedback. *IEEE transactions on Information Technology in Biomedicine*, 11(1) :58–69, 2007.

[22] William Horsthemke, Daniela Raicu, and Jacob Furst. Task-oriented medical image retrieval. In *MICCAI 2007 Workshop on Content-based Image Retrieval for Biomedical Image Archives : Achievements, Problems, and Prospects*, pages 31–44, 2007.

[23] Hayit Greenspan and Adi T Pinhas. Medical image categorization and retrieval for pacs using the gmm-kl framework. *IEEE Transactions on Information Technology in Biomedicine*, 11(2) :190–202, 2007.

[24] Mohammad-Reza Siadat, Hamid Soltanian-Zadeh, Farshad Fotouhi, and Kost Elisevich. Content-based image database system for epilepsy. *Computer Methods and Programs in Biomedicine*, 79(3) :209–226, 2005.

[25] Jinman Kim, Weidong Cai, Dagan Feng, and Hao Wu. A new way for multidimensional medical data management : volume of interest (voi)-based retrieval of medical images with visual and functional features. *IEEE Transactions on Information Technology in Biomedicine*, 10(3) :598–607, 2006.

[26] Issam El-Naqa, Yongyi Yang, Nikolas P Galatsanos, Robert M Nishikawa, and Miles N Wernick. A similarity learning approach to content-based image retrieval : application to digital mammography. *IEEE transactions on medical imaging*, 23(10) :1233–1244, 2004.

[27] Scott Doyle, Mark Hwang, Shivang Naik, Michael Feldman, John Tomaszeweski, and Anant Madabhushi. Using manifold learning for content-based image retrieval of prostate histopathology. In *MICCAI 2007 Workshop on Content-based Image Retrieval for Biomedical Image Archives : Achievements, Problems, and Prospects*, pages 53–62, 2007.

[28] E Balmashnova, B Platel, L Florack, and BM ter Haar Romeny. Content-based image retrieval by means of scale-space top-points and differential invariants. In *Proceedings of the MICCAI Workshop on Medical Content-Based Image Retrieval for Biomedical Image Archives : Achievements, Problems, and Prospects (Brisbane, Australia*, pages 83–92, 2007.

[29] Jennifer G. Dy, Carla E. Brodley, Avi Kak, Lynn S. Broderick, and Alex M. Aisen. Unsupervised feature selection applied to content-based retrieval of lung images. *IEEE transactions on pattern analysis and machine intelligence*, 25(3) :373–378, 2003.

[30] Hossein Pourghassem and Hassan Ghassemian. Content-based medical image classification using a new hierarchical merging scheme. *Computerized Medical Imaging and Graphics*, 32(8) :651–661, 2008.

[31] Sameer K Antani, L Rodney Long, and George R Thoma. A biomedical information system for combined content-based retrieval of spine x-ray images, associated text information. In *ICVGIP*, 2002.

[32] Wei-Ying Ma and BS Manjunath. A comparison of wavelet transform features for texture image annotation. In *Image Processing, 1995. Proceedings., International Conference on*, volume 2, pages 256–259. IEEE, 1995.

[33] Sabrina Tollari. *Indexation et recherche d'images par fusion d'informations textuelles et visuelles.* PhD thesis, Toulon, 2006.

[34] Robert M Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5) :786–804, 1979.

[35] Remco C Veltkamp and Michiel Hagedoorn. State of the art in shape matching. In *Principles of visual information retrieval*, pages 87–119. Springer, 2001.

[36] Isabelle Bloch, Y Gousseau, H Maître, D Matignon, B Pesquet-Popescu, F Schmitt, M Sigelle, and F Tupin. Le traitement des images. *Polycopié du cours ANIM, version*, 5 :12, 2004.

[37] Huazhong Shu, Limin Luo, and Jean-louis Caatrieux. Moment-based approaches in imaging. 1. basic features [a look at...]. *IEEE Engineering in Medicine and Biology Magazine*, 26(5) :70–74, 2007.

[38] Beijing Chen, Huazhong Shu, Gouenou Coatrieux, Gang Chen, Xingming Sun, and Jean Louis Coatrieux. Color image analysis by quaternion-type moments. *Journal of mathematical imaging and vision*, 51(1) :124–144, 2015.

[39] Tony Lindeberg. Scale invariant feature transform. *Scholarpedia*, 7(5) :10491, 2012.

[40] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[41] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf : Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006.

[42] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk : Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[43] K Velmurugan and Lt Dr S Santhosh Baboo. Content-based image retrieval using surf and colour moments. *Global Journal of Computer Science and Technology*, 2011.

[44] Thanh-Toan Do, Ewa Kijak, Teddy Furon, and Laurent Amsaleg. Deluding image recognition in sift-based cbir systems. In *Proceedings of the 2nd ACM workshop on Multimedia in forensics, security and intelligence*, pages 7–12. ACM, 2010.

[45] Anna Saro Vijendran and S Vinod Kumar. A new content based image retrieval system by hog of wavelet sub bands. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(4) :297–306, 2015.

[46] Gregory T Flitton, Toby P Breckon, and Najla Megherbi Bouallagu. Object recognition using 3d sift in complex ct volumes. In *BMVC*, pages 1–12, 2010.

[47] Hiteshree Lad and Mayuri A Mehta. Feature based object mining and tagging algorithm for digital images. In *Proceedings of International Conference on Communication and Networks*, pages 345–352. Springer, 2017.

[48] Yan Ke and Rahul Sukthankar. Pca-sift : A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[49] Luo Juan and Oubong Gwun. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4) :143–152, 2009.

[50] Maria Rifqi. Mesures de similarité, raisonnement et modélisation de l'utilisateur. *Habilitation à*, 2010.

[51] Philippe-Henri Gosselin, Micheline Najjar, Matthieu Cord, Christophe Ambroise, and Sylvie Philipp-Foliguet. Méthodes d'apprentissage pour la recherche d'images par le contenu. 2004.

[52] Francesca Cesarini, Marco Lastri, Simone Marinai, and Giovanni Soda. Encoding of modified xy trees for document classification. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 1131–1136. IEEE, 2001.

[53] Gwénolé Quellec. *Indexation et fusion multimodale pour la recherche d'information par le contenu. Application aux bases de données d'images médicales.* PhD thesis, TELECOM Bretagne, 2008.

[54] Wei Qian, Maria Kallergi, Laurence P Clarke, Huai-Dong Li, Priya Venugopal, Dansheng Song, and Robert A Clark. Tree structured wavelet transform segmentation of microcalcifications in digital mammography. *Medical physics*, 22(8) :1247–1254, 1995.

[55] Chad Carson, Megan Thomas, Serge Belongie, Joseph M Hellerstein, and Jitendra Malik. Blobworld : A system for region-based image indexing and retrieval. In *International Conference on Advances in Visual Information Systems*, pages 509–517. Springer, 1999.

[56] L Lucchese and SK Mitra. Unsupervised segmentation of color images based on k-means clustering in the chromaticity plane. In *Content-Based Access of Image and Video Libraries, 1999.(CBAIVL'99) Proceedings. IEEE Workshop on*, pages 74–78. IEEE, 1999.

[57] Chi-Ren Shyu, CE Brodley, AC Kak, Akio Kosaka, A Aisen, and L Broderick. Local versus global features for content-based image retrieval. In *Content-Based Access of Image and Video Libraries, 1998. Proceedings. IEEE Workshop on*, pages 30–34. IEEE, 1998.

[58] Mia K Markey, Joseph Y Lo, Georgia D Tourassi, and Carey E Floyd. Self-organizing map for cluster analysis of a breast cancer database. *Artificial Intelligence in Medicine*, 27(2) :113–127, 2003.

[59] Jean-Michel Cauvin. *Raisonnement médical et aide à la décision en endoscopie digestive.* PhD thesis, Rennes 1, 2001.

[60] Euripides G. M. Petrakis and A Faloutsos. Similarity searching in medical image databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(3) :435–447, 1997.

[61] Hemant Tagare, Frans M Vos, Conrade C. Jaffe, and James S. Duncan. Arrangement : A spatial relation between parts for evaluating similarity of tomographic section. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9) :880–893, 1995.

[62] Yanxi Liu and Frank Dellaert. Classification-driven medical image retrieval. In *Proc. of the Image Understanding Workshop*, 1998.

[63] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12) :1349–1380, 2000.

[64] Henning Müller, Nicolas Michoux, David Bandon, and Antoine Geissbuhler. A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International journal of medical informatics*, 73(1) :1–23, 2004.

[65] Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, 1992.

[66] Chi-Ren Shyu, Christina Pavlopoulou, Avinash C Kak, Carla E Brodley, and Lynn S Broderick. Using human perceptual categories for content-based retrieval from a medical image database. *Computer Vision and Image Understanding*, 88(3) :119–151, 2002.

[67] Michael D Abràmoff, Bram Van Ginneken, and Meindert Niemeijer. Automatic detection of red lesions in digital color fundus photographs, January 6 2009. US Patent 7,474,775.

[68] Sung-Nien Yu, Chih-Tsung Chiang, and Chin-Chiang Hsieh. A three-object model for the similarity searches of chest ct images. *Computerized Medical Imaging and Graphics*, 29(8) :617–630, 2005.

[69] Recommendation No. Rec (97) 5 of the committee of ministers to member states on the protection of medical data (adopted by the committee of ministers on 13 february 1997 at the 584th meeting of the ministers' deputies). *Council of Europe*, 1997.

[70] Catherine Quantin, Pierre Métral, and Liliane Dusserre. Intégration des contraintes du pmsi dans la mise en place d'un sih.

[71] Francis H Roger. Le resume du dossier medical : indicateur informatise de performance et de qualite des soins. 1982.

[72] Aris Gkoulalas-Divanis and Grigorios Loukides. *Medical Data Privacy Handbook*. Springer, 2015.

[73] Jean-Yves Marion. Sécurité des systèmes d'information.

[74] Secrétariat Général de la Défense Nationale. Ebios-expression des besoins et identification des objectifs de sécurité, méthode de gestion des risques, 2010.

[75] W. Pan D. Bouslimi, G. Coatrieux. Analyse de risque d'une plateforme de télémédecine,. 2010.

[76] Dalel Bouslimi. *Protection de données d'imagerie par tatouage et chiffrement-Application à la télémédecine*. PhD thesis, Télécom Bretagne, 2013.

[77] Auguste Kerckhoffs. La cryptographic militaire. *Journal des sciences militaires*, pages 5–38, 1883.

[78] Henri Gilbert and Helena Handschuh. Security analysis of sha-256 and sisters. In *International workshop on selected areas in cryptography*, pages 175–193. Springer, 2003.

[79] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11) :169–180, 1978.

[80] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2) :120–126, 1978.

[81] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 365–377. ACM, 1982.

[82] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4) :469–472, 1985.

[83] Josh Benaloh. Dense probabilistic encryption. In *Proceedings of the workshop on selected areas of cryptography*, pages 120–128, 1994.

[84] David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In *Proceedings of the 5th ACM conference on Computer and communications security*, pages 59–66. ACM, 1998.

[85] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. *Advances in Cryptology—EUROCRYPT'98*, pages 308–318, 1998.

[86] Pascal Paillier et al. Public-key cryptosystems based on composite degree residuosity classes. In *Eurocrypt*, volume 99, pages 223–238. Springer, 1999.

[87] Ivan Damgard and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Public Key Cryptography*, volume 1992, pages 119–136. Springer, 2001.

[88] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Multi-bit cryptosystems based on lattice problems. In *International Workshop on Public Key Cryptography*, pages 315–329. Springer, 2007.

[89] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC*, volume 3378, pages 325–341. Springer, 2005.

[90] Craig Gentry. *A fully homomorphic encryption scheme*. Stanford University, 2009.

[91] Josh Daniel Cohen Benaloh. Verifiable secret-ballot elections. 1987.

[92] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6) :644–654, 1976.

[93] Kristian Gjøsteen. Subgroup membership problems and public key cryptosystems. 2004.

[94] Steven D Galbraith. Elliptic curve paillier schemes. *Journal of Cryptology*, 15(2) :129–138, 2002.

[95] Tomas Sander, Adam Young, and Moti Yung. Non-interactive cryptocomputing for nc1. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, pages 554–, Washington, DC, USA, 1999. IEEE Computer Society.

[96] Yuval Ishai and Anat Paskin. Evaluating branching programs on encrypted data. In *TCC*, volume 4392, pages 575–594. Springer, 2007.

[97] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. A simple bgn-type cryptosystem from lwe. *Advances in Cryptology–EUROCRYPT 2010*, pages 506–522, 2010.

[98] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *EUROCRYPT*, volume 6632, pages 129–148. Springer, 2011.

[99] Craig Gentry, Shai Halevi, and Nigel P Smart. Better bootstrapping in fully homomorphic encryption. In *International Workshop on Public Key Cryptography*, pages 1–16. Springer, 2012.

[100] Nigel P Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography*, volume 6056, pages 420–443. Springer, 2010.

[101] Nigel P Smart and Frederik Vercauteren. Fully homomorphic simd operations. *Designs, codes and cryptography*, pages 1–25, 2014.

[102] Damien Stehlé and Ron Steinfeld. Faster fully homomorphic encryption. *Advances in Cryptology-ASIACRYPT 2010*, pages 377–394, 2010.

[103] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–43. Springer, 2010.

[104] Nick Howgrave-Graham. Approximate integer common divisors. In *CaLC*, volume 1, pages 51–66. Springer, 2001.

[105] Yuanmi Chen and Phong Q Nguyen. Faster algorithms for approximate common divisors : Breaking fully-homomorphic-encryption challenges over the integers. In *EUROCRYPT*, volume 7237, pages 502–519. Springer, 2012.

[106] Jung Hee Cheon and Damien Stehlé. Fully homomophic encryption over the integers revisited. *EUROCRYPT (1)*, 9056 :513–536, 2015.

[107] Koji Nuida and Kaoru Kurosawa. (batch) fully homomorphic encryption over integers for non-binary message spaces. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 537–555. Springer, 2015.

[108] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *Annual cryptology conference*, pages 505–524. Springer, 2011.

[109] Zvika Brakerski, Craig Gentry, and Shai Halevi. Packed ciphertexts in lwe-based homomorphic encryption. In *Public Key Cryptography*, volume 7778, pages 1–13. Springer, 2013.

[110] Jacob Alperin-Sheriff and Chris Peikert. Practical bootstrapping in quasilinear time. In *Advances in Cryptology–CRYPTO 2013*, pages 1–20. Springer, 2013.

[111] Craig Gentry, Shai Halevi, Chris Peikert, and Nigel P Smart. Field switching in bgv-style homomorphic encryption. *Journal of Computer Security*, 21(5) :663–684, 2013.

[112] Léo Ducas and Daniele Micciancio. Fhew : Bootstrapping homomorphic encryption in less than a second. *EUROCRYPT (1)*, 9056 :617–640, 2015.

[113] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors : Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013*, pages 75–92. Springer, 2013.

[114] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 35–54. Springer, 2013.

[115] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3) :13, 2014.

[116] Shai Halevi and Victor Shoup. Algorithms in helib. In *International Cryptology Conference*, pages 554–571. Springer, 2014.

[117] Shai Halevi and Victor Shoup. Bootstrapping for helib. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 641–670. Springer, 2015.

[118] Michael O. Rabin. How to exchange secrets with oblivious transfer, 2005. Harvard University Technical Report 81 talr@watson.ibm.com 12955 received 21 Jun 2005.

[119] Andrew C Yao. Protocols for secure computations. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, pages 160–164. IEEE, 1982.

[120] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit : Free xor gates and applications. *Automata, Languages and Programming*, pages 486–498, 2008.

[121] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11) :612–613, 1979.

[122] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1) :1–30, 2006.

[123] Wenjun Lu, Ashwin Swaminathan, Avinash L Varna, Min Wu, et al. Enabling search over encrypted multimedia databases. In *Media Forensics and Security*, volume 7254, 2009.

[124] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 235–253. Springer, 2009.

[125] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. Efficient privacy-preserving face recognition. In *ICISC*, volume 9, pages 229–244. Springer, 2009.

[126] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster secure two-party computation using garbled circuits. In *USENIX Security Symposium*, volume 201, 2011.

[127] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 162–167. IEEE, 1986.

[128] David Evans, Yan Huang, Jonathan Katz, and Lior Malka. Efficient privacy-preserving biometric identification. In *Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*, 2011.

[129] Marina Blanton and Paolo Gasti. Secure and efficient protocols for iris and fingerprint identification. In *European Symposium on Research in Computer Security*, pages 190–209. Springer, 2011.

[130] Margarita Osadchy, Benny Pinkas, Ayman Jarrous, and Boaz Moskovich. Scifi-a system for secure face identification. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 239–254. IEEE, 2010.

[131] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 245–254. ACM, 1999.

[132] Marina Blanton and Mehrdad Aliasgari. Secure outsourced computation of iris matching. *Journal of Computer Security*, 20(2-3) :259–305, 2012.

[133] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. *Advances in Cryptology–EUROCRYPT 2008*, pages 146–162, 2008.

[134] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In *TCC*, volume 5444, pages 457–473. Springer, 2009.

[135] Matthias Schneider and Thomas Schneider. Notes on non-interactive secure comparison in image feature extraction in the encrypted domain with privacy-preserving sift. In *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, pages 135–140. ACM, 2014.

[136] Zhan Qin, Jingbo Yan, Kui Ren, Chang Wen Chen, and Cong Wang. Towards efficient privacy-preserving image feature extraction in cloud computing. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 497–506. ACM, 2014.

[137] Alexandra Boldyreva, Nathan Chenette, Younho Lee, Adam O'neill, et al. Order-preserving symmetric encryption. In *Eurocrypt*, volume 5479, pages 224–241. Springer, 2009.

[138] Zhan Qin, Jingbo Yan, Kui Ren, Chang Wen Chen, and Cong Wang. Secsift : Secure image sift feature extraction in cloud computing. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 12(4s) :65, 2016.

[139] Qian Wang, Shengshan Hu, Kui Ren, Jingjun Wang, Zhibo Wang, and Minxin Du. Catch me in the dark : Effective privacy-preserving outsourcing of feature extractions over image data. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[140] Yu Bai, Li Zhuo, Bo Cheng, and Yuan Fan Peng. Surf feature extraction in encrypted domain. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE, 2014.

[141] Qian Wang, Shengshan Hu, Jingjun Wang, and Kui Ren. Secure surfing : Privacy-preserving speeded-up robust feature extractor. In *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on*, pages 700–710. IEEE, 2016.

[142] Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon, Ryusuke Masuoka, and Jesus Molina. Controlling data in the cloud : outsourcing computation without outsourcing control. *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 85–90, 2009.

[143] National Vulnerability Databasecve statistics, 2014.

[144] J Alex Halderman, Seth D Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A Calandrino, Ariel J Feldman, Jacob Appelbaum, and Edward W Felten. Lest we remember : cold-boot attacks on encryption keys. *Communications of the ACM*, 52(5) :91–98, 2009.

[145] Dave Lewis. icloud data breach : Hacking and celebrity photos. *Forbes Online*, 2014.

[146] Stephane G Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7) :674–693, 1989.

[147] Peijia Zheng and Jiwu Huang. Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain. *IEEE Transactions on Image Processing*, 22(6) :2455–2468, 2013.

[148] Thijs Veugen. Improving the dgk comparison protocol. In *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*, pages 49–54. IEEE, 2012.

[149] William Gasarch. A survey on private information retrieval. *The Bulletin of the EATCS*, 82(72-107) :1, 2004.

[150] Nico Schlitter. A protocol for privacy preserving neural network learning on horizontal partitioned data. *PSD*, 2008.

[151] Tingting Chen and Sheng Zhong. Privacy-preserving backpropagation neural network learning. *IEEE Transactions on Neural Networks*, 20(10) :1554–1564, 2009.

[152] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015.

[153] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. Machine learning classification over encrypted data. In *NDSS*, 2015.

[154] Hervé Chabanne, Amaury de Wargny, Jonathan Milgram, Constance Morel, and Emmanuel Prouff. Privacy-preserving classification on deep neural network. *IACR Cryptology ePrint Archive*, 2017 :35, 2017.

[155] Pengtao Xie, Misha Bilenko, Tom Finley, Ran Gilad-Bachrach, Kristin Lauter, and Michael Naehrig. Crypto-nets : Neural networks over encrypted data. *arXiv preprint arXiv :1412.6181*, 2014.

[156] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[157] Thijs Veugen. Encrypted integer division. In *Information Forensics and Security (WIFS), 2010 IEEE International Workshop on*, pages 1–6. IEEE, 2010.

[158] Wenxiu Ding, Zheng Yan, and Robert H Deng. Encrypted data processing with homomorphic re-encryption. *Information Sciences*, 409 :35–55, 2017.

[159] Fen Wu, Hong Zhong, Runhua Shi, and Hongsheng Huang. Secure two-party computation of the quadratic function's extreme minimal value. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 2975–2978. IEEE, 2012.

[160] Oded Goldreich. *Foundations of cryptography : volume 2, basic applications*. Cambridge university press, 2009.

[161] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. *Advances in Cryptology—EUROCRYPT'98*, pages 127–144, 1998.

[162] Giuseppe Ateniese and Susan Hohenberger. Proxy re-signatures : new definitions, algorithms, and applications. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 310–319. ACM, 2005.

[163] Zhiguang Qin, Hu Xiong, Shikun Wu, and Jennifer Batamuliza. A survey of proxy re-encryption for secure data sharing in cloud computing. *IEEE Transactions on Services Computing*, 2016.

[164] Markus Jakobsson. On quorum controlled asymmetric proxy re-encryption. In *Public key cryptography*, pages 632–632. Springer, 1999.

[165] Anca-Andreea Ivan and Yevgeniy Dodis. Proxy cryptography revisited. In *NDSS*, 2003.

[166] Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In *Applied Cryptography and Network Security*, pages 288–306. Springer, 2007.

[167] Jinguang Han, Willy Susilo, and Yi Mu. Identity-based data storage in cloud computing. *Future Generation Computer Systems*, 29(3) :673–681, 2013.

[168] Cheng-Kang Chu, Wen-Guey Tzeng, et al. Identity-based proxy re-encryption without random oracles. In *ISC*, volume 7, pages 189–202. Springer, 2007.

[169] Toshihiko Matsuo. Proxy re-encryption systems for identity-based encryption. *Pairing-Based Cryptography–Pairing 2007*, pages 247–267, 2007.

[170] Xiaohui Liang, Zhenfu Cao, Huang Lin, and Jun Shao. Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 276–286. ACM, 2009.

[171] Peng Xu, Tengfei Jiao, Qianhong Wu, Wei Wang, and Hai Jin. Conditional identity-based broadcast proxy re-encryption and its application to cloud email. *IEEE Transactions on Computers*, 65(1) :66–79, 2016.

[172] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Certificateless public key encryption without pairing. In *International Conference on Information Security*, pages 134–148. Springer, 2005.

[173] Robert H Deng, Jian Weng, Shengli Liu, and Kefei Chen. Chosen-ciphertext secure proxy re-encryption without pairings. In *International Conference on Cryptology and Network Security*, pages 1–17. Springer, 2008.

[174] Brian A Wichmann and I David Hill. Algorithm as 183 : An efficient and portable pseudo-random number generator. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 31(2) :188–190, 1982.

[175] Pierre L'ecuyer. Tables of linear congruential generators of different sizes and good lattice structure. *Mathematics of Computation of the American Mathematical Society*, 68(225) :249–260, 1999.

[176] Pierre L'ecuyer, Richard Simard, E Jack Chen, and W David Kelton. An object-oriented random-number package with many long streams and substreams. *Operations research*, 50(6) :1073–1075, 2002.

[177] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010.

[178] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical ? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124. ACM, 2011.

[179] Nasir Memon and Ping Wah Wong. A buyer-seller watermarking protocol. *IEEE Transactions on image processing*, 10(4) :643–649, 2001.

[180] Xinpeng Zhang. Reversible data hiding in encrypted image. *IEEE signal processing letters*, 18(4) :255–258, 2011.

[181] William Puech, Marc Chaumont, and Olivier Strauss. A reversible data hiding method for encrypted images. In *Electronic Imaging*, number 6819, page 68191E. SPIE/IS&T, 2008.

[182] Xinpeng Zhang. Separable reversible data hiding in encrypted image. *IEEE transactions on information forensics and security*, 7(2) :826–832, 2012.

[183] Di Xiao and Shoukuo Chen. Separable data hiding in encrypted image based on compressive sensing. *Electronics Letters*, 50(8) :598–600, 2014.

[184] Roland Schmitz, Shujun Li, Christos Grecos, and Xinpeng Zhang. Towards more robust commutative watermarking-encryption of images. In *Multimedia (ISM), 2013 IEEE International Symposium on*, pages 283–286. IEEE, 2013.

[185] Dalel Bouslimi, Gouenou Coatrieux, Michel Cozic, and Christian Roux. A joint encryption/watermarking system for verifying the reliability of medical images. *IEEE Transactions on Information Technology in Biomedicine*, 16(5) :891–899, 2012.

[186] Roland Schmitz, Shujun Li, Christos Grecos, and Xinpeng Zhang. A new approach to commutative watermarking-encryption. In *Communications and Multimedia Security*, pages 117–130. Springer, 2012.

[187] Kede Ma, Weiming Zhang, Xianfeng Zhao, Nenghai Yu, and Fenghua Li. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transactions on Information Forensics and Security*, 8(3) :553–562, 2013.

[188] Weiming Zhang, Kede Ma, and Nenghai Yu. Reversibility improved data hiding in encrypted images. *Signal Processing*, 94 :118–127, 2014.

[189] Dalel Bouslimi, Gouenou Coatrieux, Michel Cozic, and Christian Roux. Tatouage d'images chiffrées pour une protection a priori et a posteriori : application aux images d'échographie. In *TAIMA 2013 : atelier Traitement et Analyse de l'Information : Méthodes et Applications*, volume 1, pages 63–71, 2013.

[190] Keshi Chen and Tenkasi V Ramabadran. Near-lossless compression of medical images through entropy-coded dpcm. *IEEE Transactions on Medical Imaging*, 13(3) :538–548, 1994.

Résumé

La mutualisation et l'externalisation de données concernent de nombreux domaines y compris celui de la santé. Au-delà de la réduction des coûts de maintenance, l'intérêt est d'améliorer la prise en charge des patients par le déploiement d'outils d'aide au diagnostic fondés sur la réutilisation des données. Dans un tel environnement, la sécurité des données (confidentialité, intégrité et traçabilité) est un enjeu majeur. C'est dans ce contexte que s'inscrivent ces travaux de thèse. Ils concernent en particulier la sécurisation des techniques de recherche d'images par le contenu (CBIR) et de « machine learning » qui sont au cœur des systèmes d'aide au diagnostic. Ces techniques permettent de trouver des images semblables à une image requête non encore interprétée. L'objectif est de définir des approches capables d'exploiter des données externalisées et sécurisées, et de permettre à un « cloud » de fournir une aide au diagnostic. Plusieurs mécanismes permettent le traitement de données chiffrées, mais la plupart sont dépendants d'interactions entre différentes entités (l'utilisateur, le cloud voire un tiers de confiance) et doivent être combinés judicieusement de manière à ne pas laisser fuir d'information lors d'un traitement.

Au cours de ces trois années de thèse, nous nous sommes dans un premier temps intéressés à la sécurisation à l'aide du chiffrement homomorphe, d'un système de CBIR externalisé sous la contrainte d'aucune interaction entre le fournisseur de service et l'utilisateur. Dans un second temps, nous avons développé une approche de « Machine Learning » sécurisée fondée sur le perceptron multicouches, dont la phase d'apprentissage peut être externalisée de manière sûre, l'enjeu étant d'assurer la convergence de cette dernière. L'ensemble des données et des paramètres du modèle sont chiffrés. Du fait que ces systèmes d'aides doivent exploiter des informations issues de plusieurs sources, chacune externalisant ses données chiffrées sous sa propre clef, nous nous sommes intéressés au problème du partage de données chiffrées. Un problème traité par les schémas de « Proxy Re-Encryption » (PRE). Dans ce contexte, nous avons proposé le premier schéma PRE qui permet à la fois le partage et le traitement des données chiffrées. Nous avons également travaillé sur un schéma de tatouage de données chiffrées pour tracer et vérifier l'intégrité des données dans cet environnement partagé. Le message tatoué dans le chiffré est accessible que l'image soit ou non chiffrée et offre plusieurs services de sécurité fondés sur le tatouage.

Mots clef : Cloud-computing, Chiffrement homomorphe, Calcul multipartite sécurisé, Recherche par le contenu, Proxy Re-Encryption, Tatouage des images, Apprentissage automatique sécurisé

Abstract

Cloud computing has emerged as a successful paradigm allowing individuals and companies to store and process large amounts of data without a need to purchase and maintain their own networks and computer systems. In healthcare for example, different initiatives aim at sharing medical images and Personal Health Records (PHR) in between health professionals or hospitals with the help of the cloud. In such an environment, data security (confidentiality, integrity and traceability) is a major issue. In this context that these thesis works, it concerns in particular the securing of Content Based Image Retrieval (CBIR) techniques and machine learning (ML) which are at the heart of diagnostic decision support systems. These techniques make it possible to find similar images to an image not yet interpreted. The goal is to define approaches that can exploit secure externalized data and enable a cloud to provide a diagnostic support. Several mechanisms allow the processing of encrypted data, but most are dependent on interactions between different entities (the user, the cloud or a trusted third party) and must be combined judiciously so as to not leak information.

During these three years of thesis, we initially focused on securing an outsourced CBIR system under the constraint of no interaction between the users and the service provider (cloud). In a second step, we have developed a secure machine learning approach based on multilayer perceptron (MLP), whose learning phase can be outsourced in a secure way, the challenge being to ensure the convergence of the MLP. All the data and parameters of the model are encrypted using homomorphic encryption. Because these systems need to use information from multiple sources, each of which outsources its encrypted data under its own key, we are interested in the problem of sharing encrypted data. A problem known by the "Proxy Re-Encryption" (PRE) schemes. In this context, we have proposed the first PRE scheme that allows both the sharing and the processing of encrypted data. We also worked on watermarking scheme over encrypted data in order to trace and verify the integrity of data in this shared environment. The embedded message is accessible whether or not the image is encrypted and provides several services.

Keywords: Homomorphic encryption, Digital watermarking, Cloud-computing, Secure machine learning, Secure multiparty computation, Content based image retrieval

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

N° d'ordre : 2017IMTA0063

IMT Atlantique Bretagne-Pays de la Loire - www.imt-atlantique.fr

Campus de Brest
Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 03
T +33 (0)2 29 00 11 11
F +33 (0)2 29 00 10 00

Campus de Nantes
4, rue Alfred Kastler - La Chantrerie
CS 20722
44307 Nantes Cedex 3
T +33 (0)2 51 85 81 00
F +33 (0)2 51 85 81 99

Campus de Rennes
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
T +33 (0)2 99 12 70 00
F +33 (0)2 99 12 70 08