Méthodologie d'évaluation par simulation de la sécurité des circuits face aux attaques par faute

Olivier FAURAX

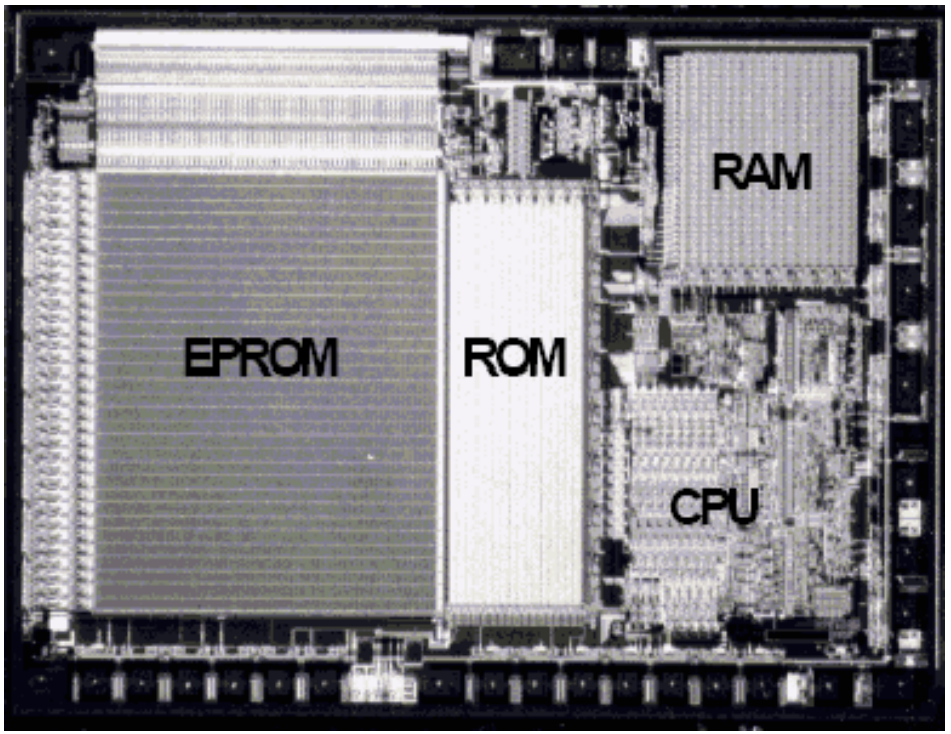
Université de la Mediterranée
Centre Microélectronique de Provence
ST Microelectronics

3 juillet 2008

Plan

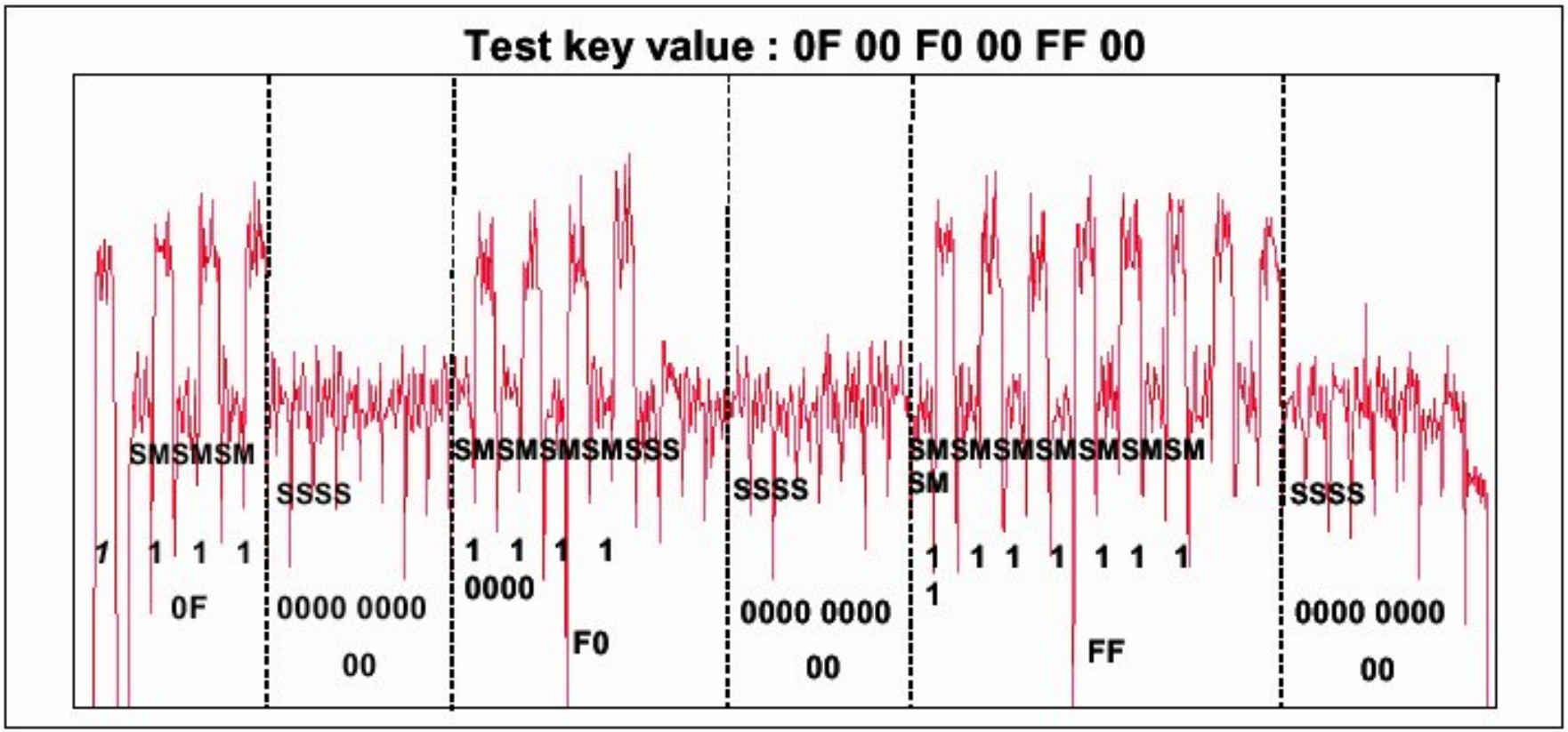
Introduction

- Enjeu économique : brique de sécurité des télécoms, de la finance, etc.
- 25 mm^2 (miniaturisation)
- 3875 millions d'unités prévues pour 2008
- D'une simple mémoire jusqu'à la cryptographie
- Nouvelles menaces : attaque par faute



Attaques connues sur les circuits

- Analyse du temps d'exécution
- Analyse de consommation électrique (SPA/DPA)
- Rayonnement électromagnétique (SEMA/DEMA)



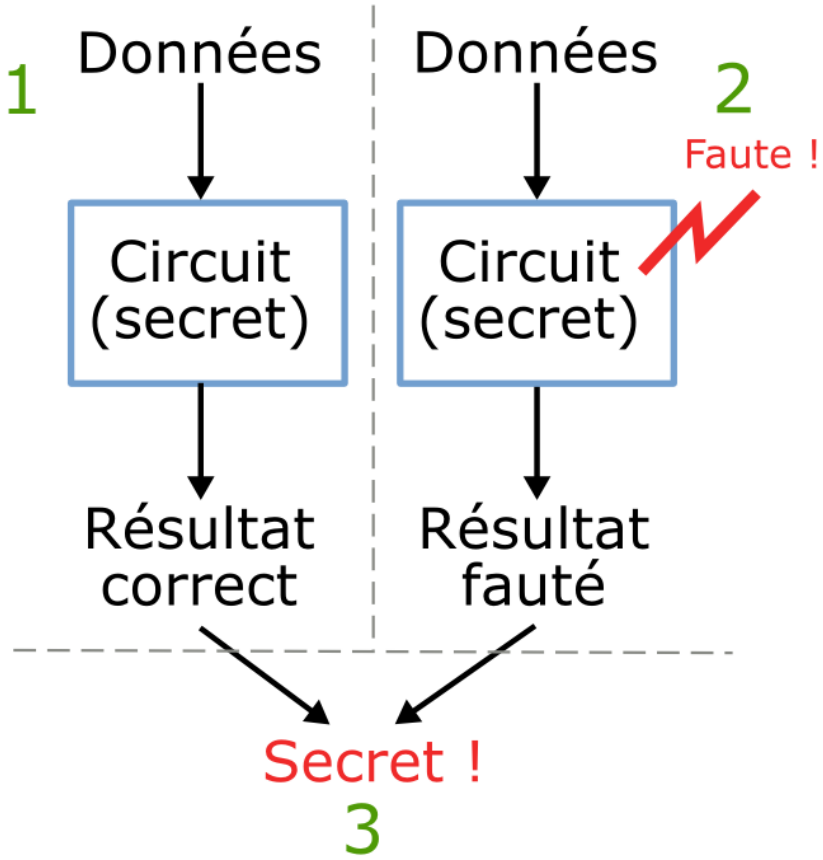
Attaque par faute

Principe

1. Exécution normale
2. Exécution en présence de faute
3. Analyse conjointe

But de la thèse

- Injection de fautes avant construction du circuit
- Outil de simulation des attaques par faute



Injection physique & injection logicielle

Injection physique

- Perturbations physiques sur le circuit final
- Très représentatif des fautes « naturelles »
- Nécessite du matériel spécialisé
- Peu de contrôlabilité

Injection logicielle

- Mécanismes internes : logiciel intégré, scan-chain, JTAG
- À la compilation ou dynamiquement (condition de déclenchement)
- Contrôlabilité améliorée
- Nécessite le circuit

Injection en simulation

Principe

- Simulation du circuit
- Simulation de faute sur la représentation du circuit
- Contrôlabilité limitée seulement par le modèle de circuit
- Plus lent à effectuer

Outils

- FOCUS : modélisation physique (VLSI)
- MEFISTO : manipulation des signaux (mutants, saboteurs)
- DEPEND : comportement des composants modélisés en C++
- VERIFY : manipulation des signaux VHDL
- SINJECT : du niveau switch au niveau structurel

Approche

État de l'art

- La sécurité est une contrainte industrielle
- Les attaques sur les circuits sont difficiles à prendre en compte
- Beaucoup de travaux sur les fautes sont orientés vers la fiabilité

Approche

- Nécessité d'une nouvelle méthodologie pour estimer l'impact des fautes sur la sécurité d'un circuit
- Outil d'injection de fautes en simulation pour tester un circuit dès sa conception

Plan

Objectifs

Buts

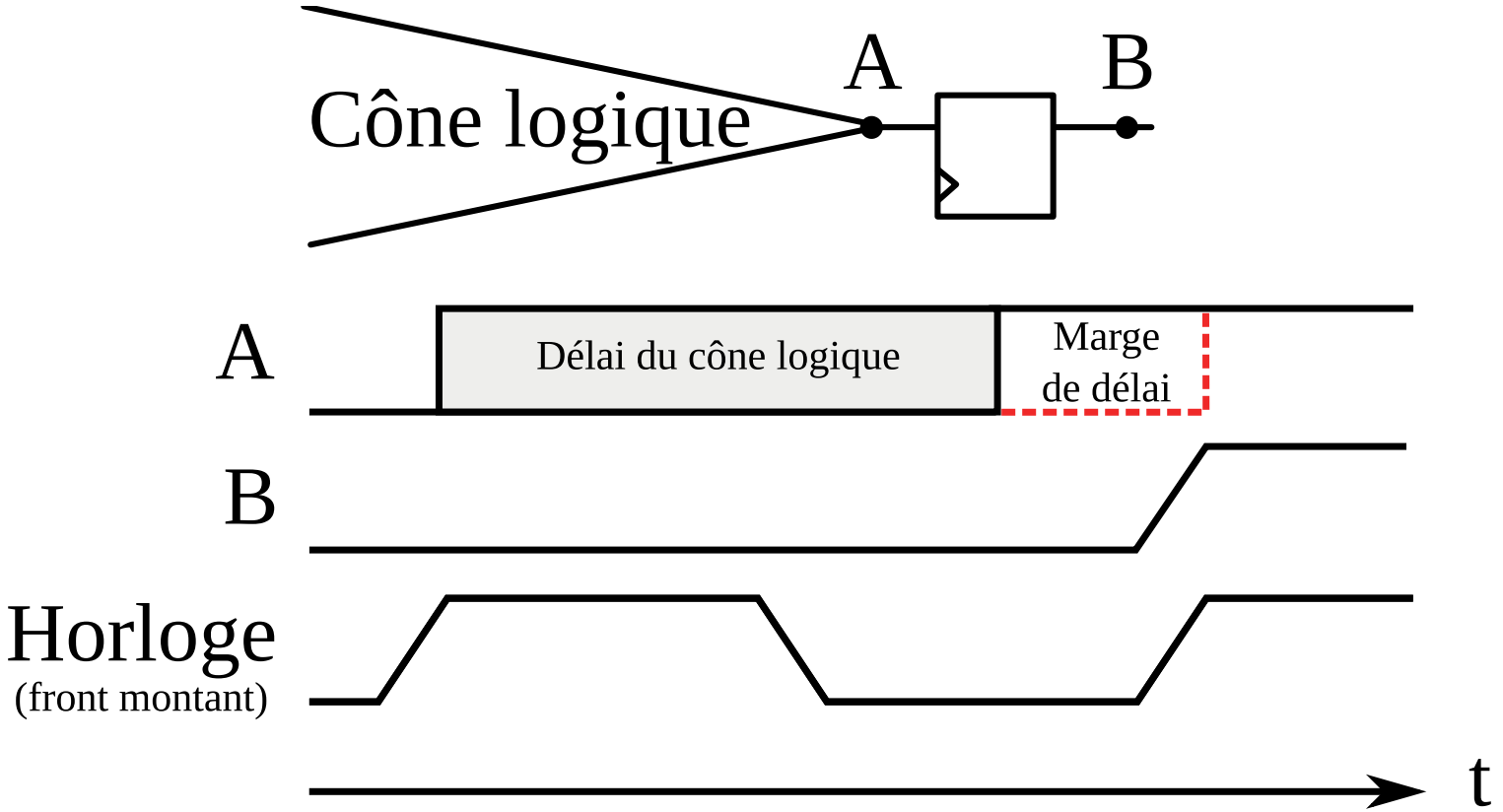
- Modèle de faute réaliste
- Estimation de la robustesse face aux attaques par faute
- Analyse et évaluation automatisées du circuit

Problématique

- Modéliser les fautes
- Choisir les fautes à simuler
- Classifier les résultats

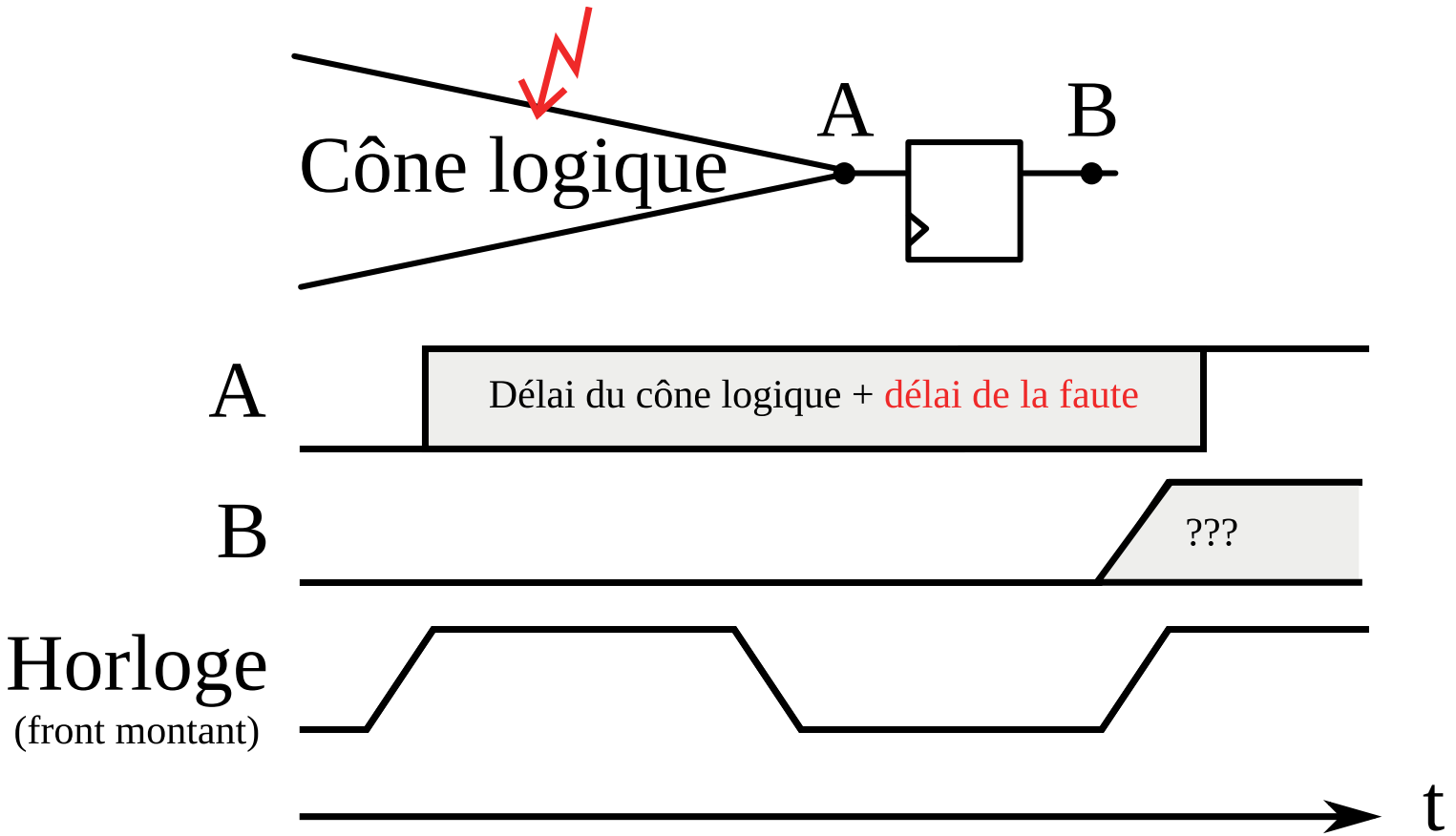
Fautes transitoires de délai (1/2)

Cône logique : ensemble de portes logiques interconnectées



Fautes transitoires de délai (2/2)

Modèle : Ajout de délai sur tout le circuit (ex : coupure d'alimentation)



Choix des fautes à simuler

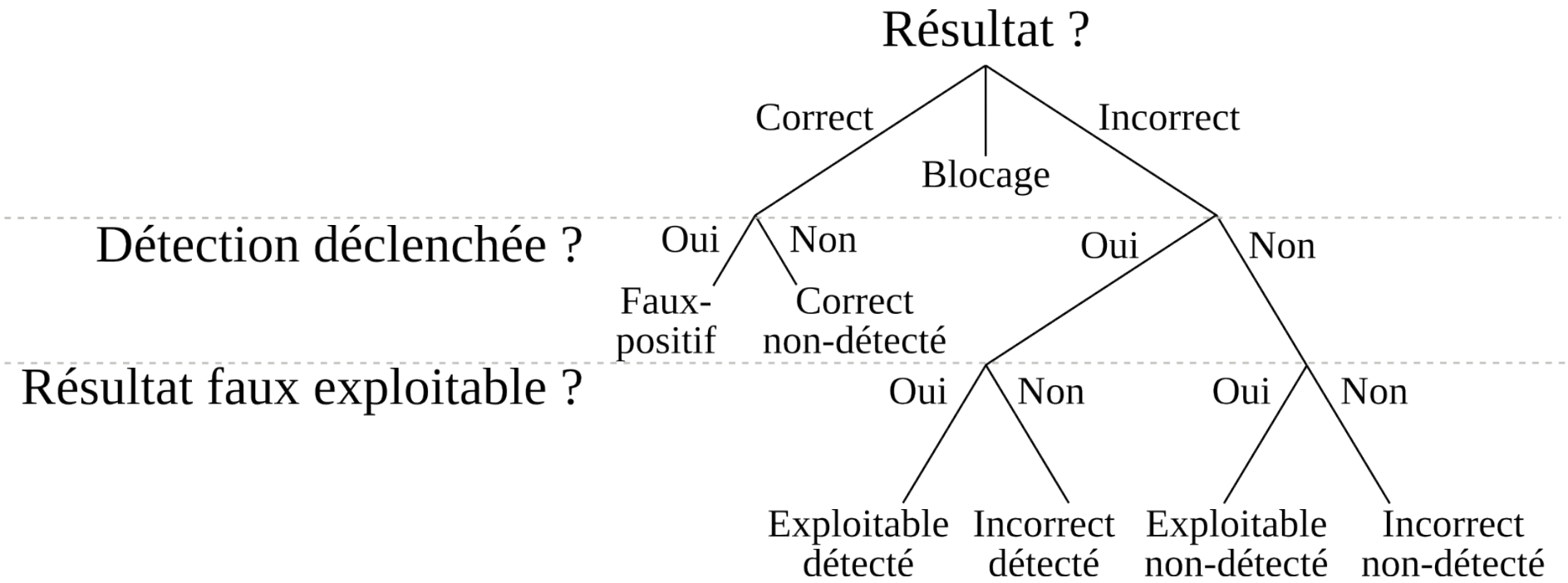
Sélection de fautes

- Injection simple exhautive encore possible pour de petits circuits
- Injection multiple : nombre d'injection exponentiel !
- Sélection des moments d'injection
- Sélection des endroits d'injection

Notion de poids

- Représentation de l'importance d'une faute
- Attaché à un endroit et un moment d'injection
- Le calcul du poids dépends du modèle de faute et de l'utilisateur
- Les fautes sont alors ordonnées

Classification des fautes



Bilan

Méthodologie

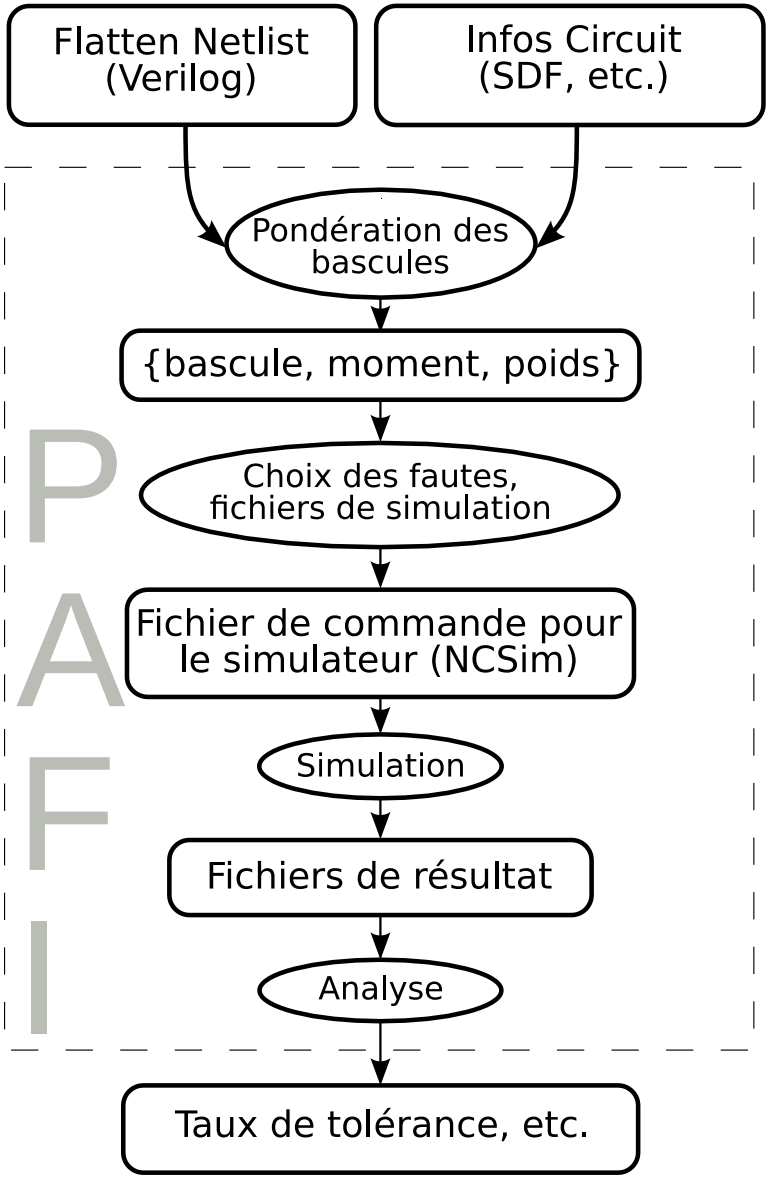
- Sélection de modèles de faute réalistes
- Choix des fautes à simuler
- Classification des fautes pour l'évaluation

Implémentation

- Analyse du circuit et choix des fautes automatisables

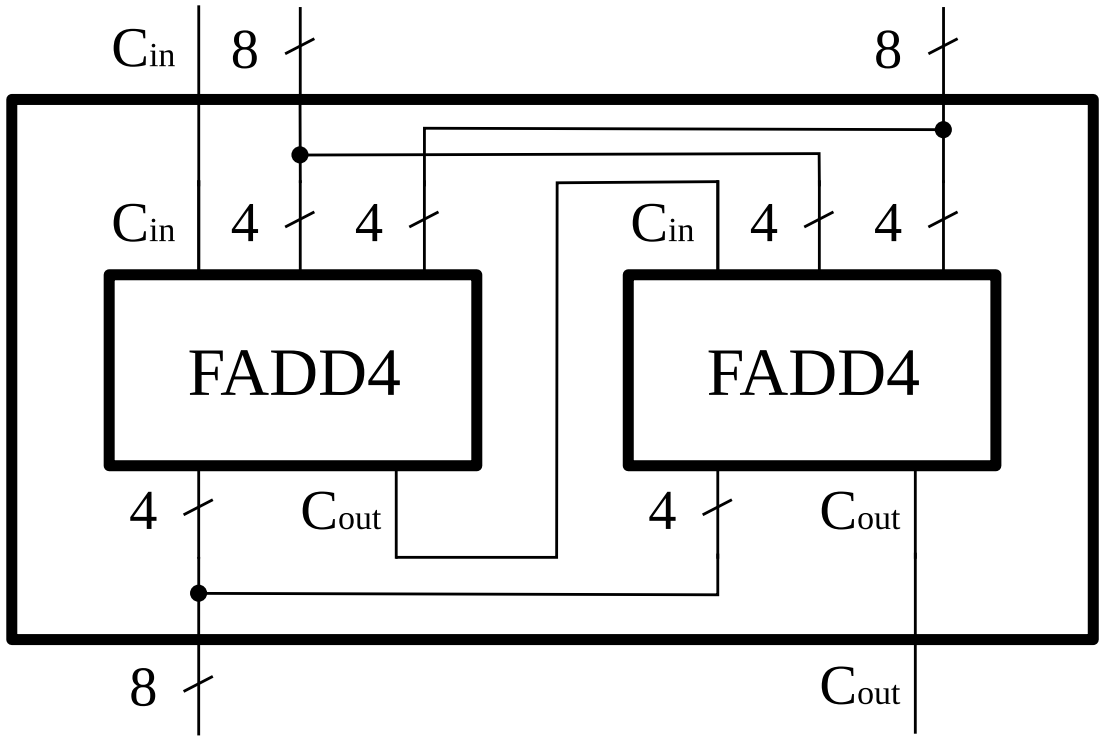
Plan

Flot de conception de PAFI



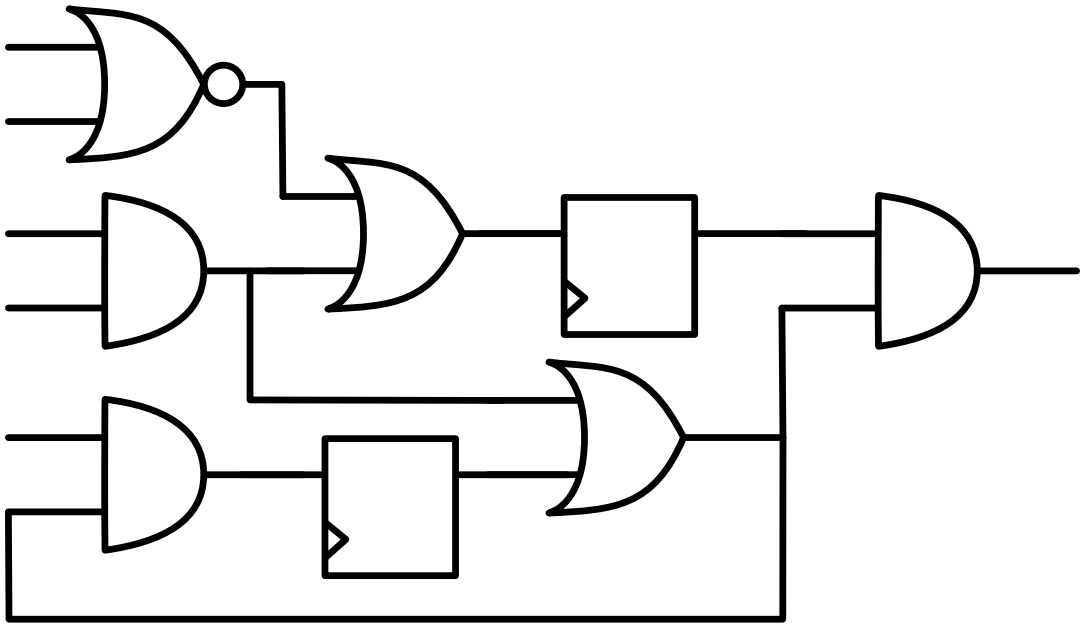
Modèles de circuit

- Modélisation analogique : trop détaillée
- Modélisation hiérarchique : peu représentative



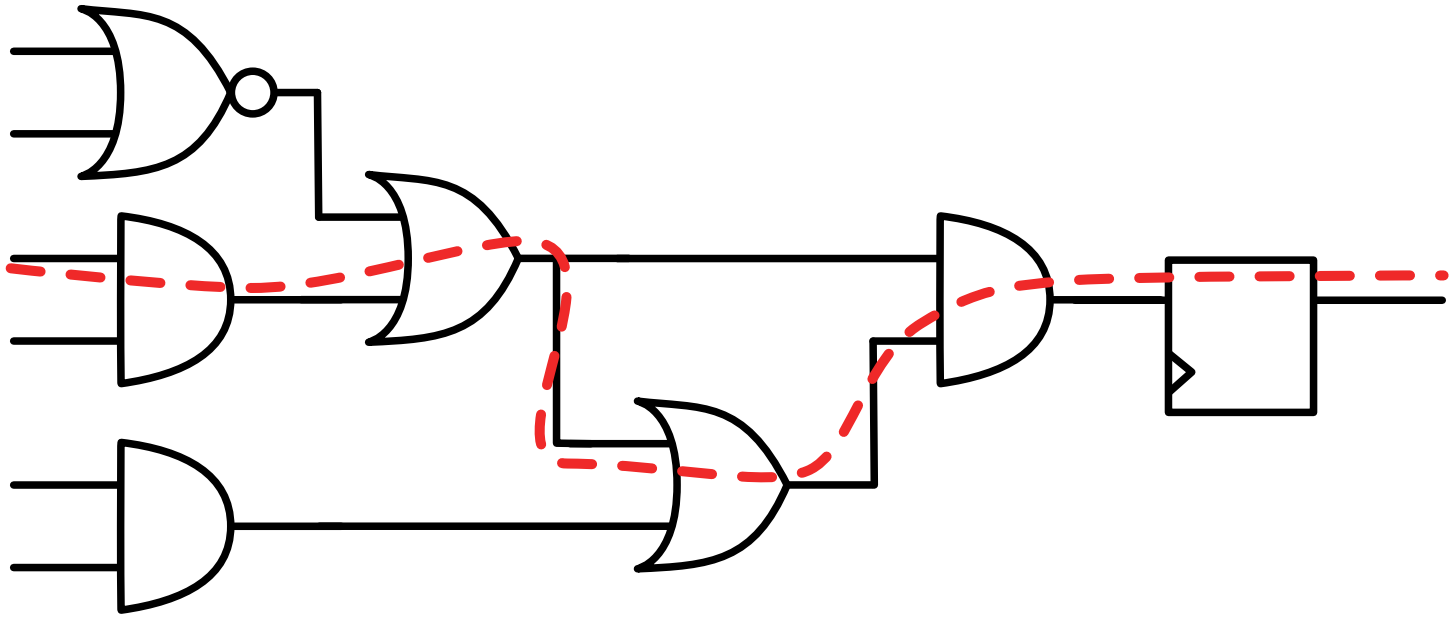
Modèles de circuit

- Modélisation analogique : trop détaillée
- Modélisation hiérarchique : peu représentative
- Modélisation logique aplatie : meilleur compromis



Pondération par le délai

- Petite marge = grande sensibilité à un ajout de délai
- Poids d'une bascule : maximum des délais depuis les entrées



Méthodes d'injection

En pratique

- Analyse lexicale du fichier en langage Verilog
- Constitution de la liste des points et moments d'injection
- Calcul du poids
- Forçage de la valeur d'entrée d'une bascule au moment voulu

Limitations d'injection

- Une seule entrée (ou équivalent)
- Circuits synchrones

Analyse des résultats

Métriques

- Taux de résultats faux
- Taux de blocages du circuit
- Taux de faux positifs (détection erronée)
- Mesure pertinente suivant le contexte

Exemple

- Dans le cas d'un circuit qui recalcule en cas de détection d'erreur, on mesurera aussi les fautes qui augmentent le temps de calcul

Avantages et limites de PAFI

Avantages

- Adéquation à la méthodologie
- Entièrement automatique (scripts)
- Flexible et adaptable à n'importe quel circuit

Limites

- Pas d'injections multiples actuellement
- Adaptation parfois laborieuse

Plan

Cryptographie symétrique

DES (Data Encryption Standard)

- Algorithme standardisé en 1977
- Clé de 56 bits (faible !)
- A évolué en triple DES (112 bits)
- Standard supplanté par l'AES

AES (Advanced Encryption Standard)

- Algorithme standardisé en 2001
- Clé de 128, 192 ou 256 bits

Cryptographie asymétrique

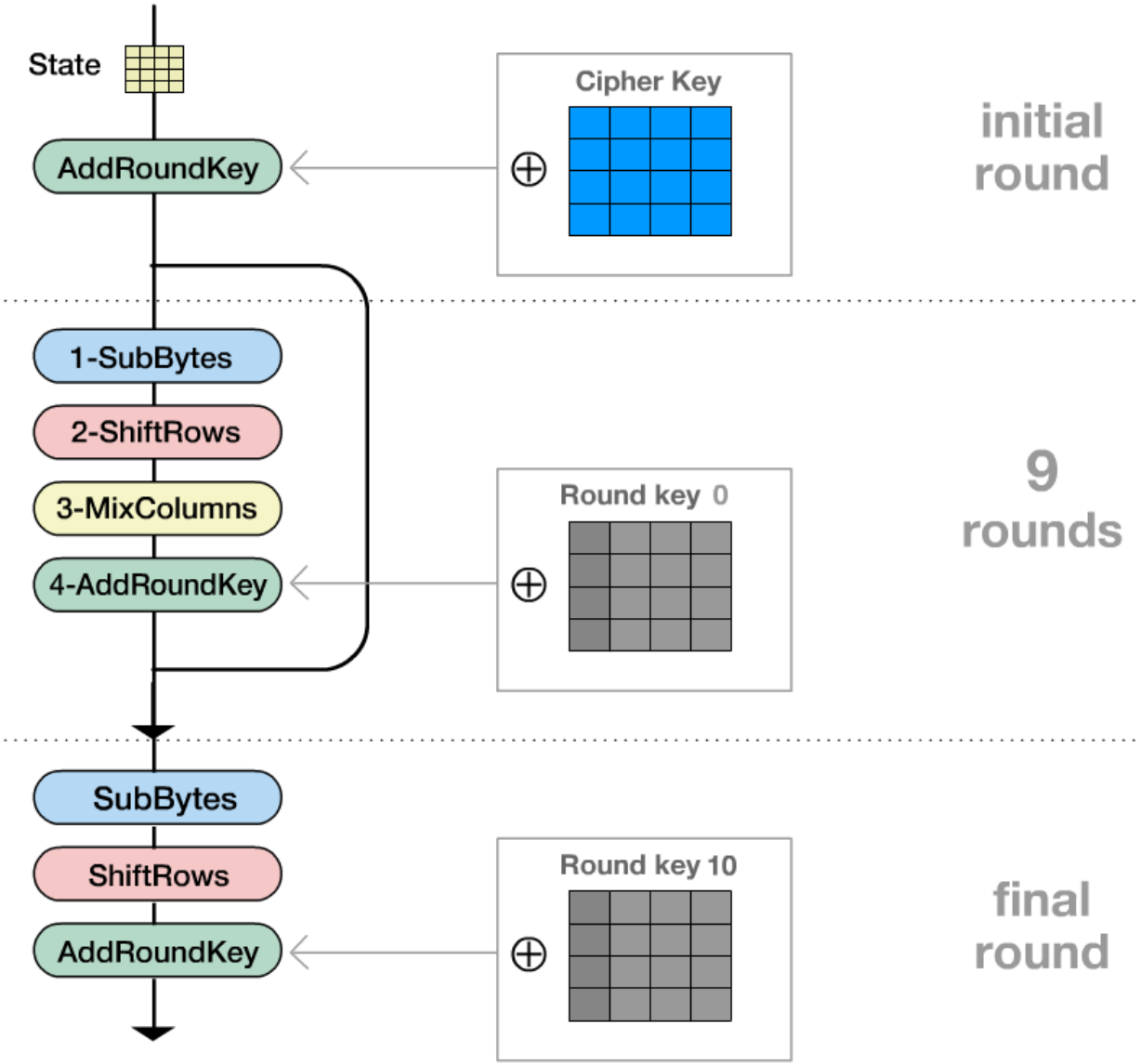
RSA

- Algorithme asymétrique publié en 1977
- Clé de 1024, 2048, 4096 bits, voire plus

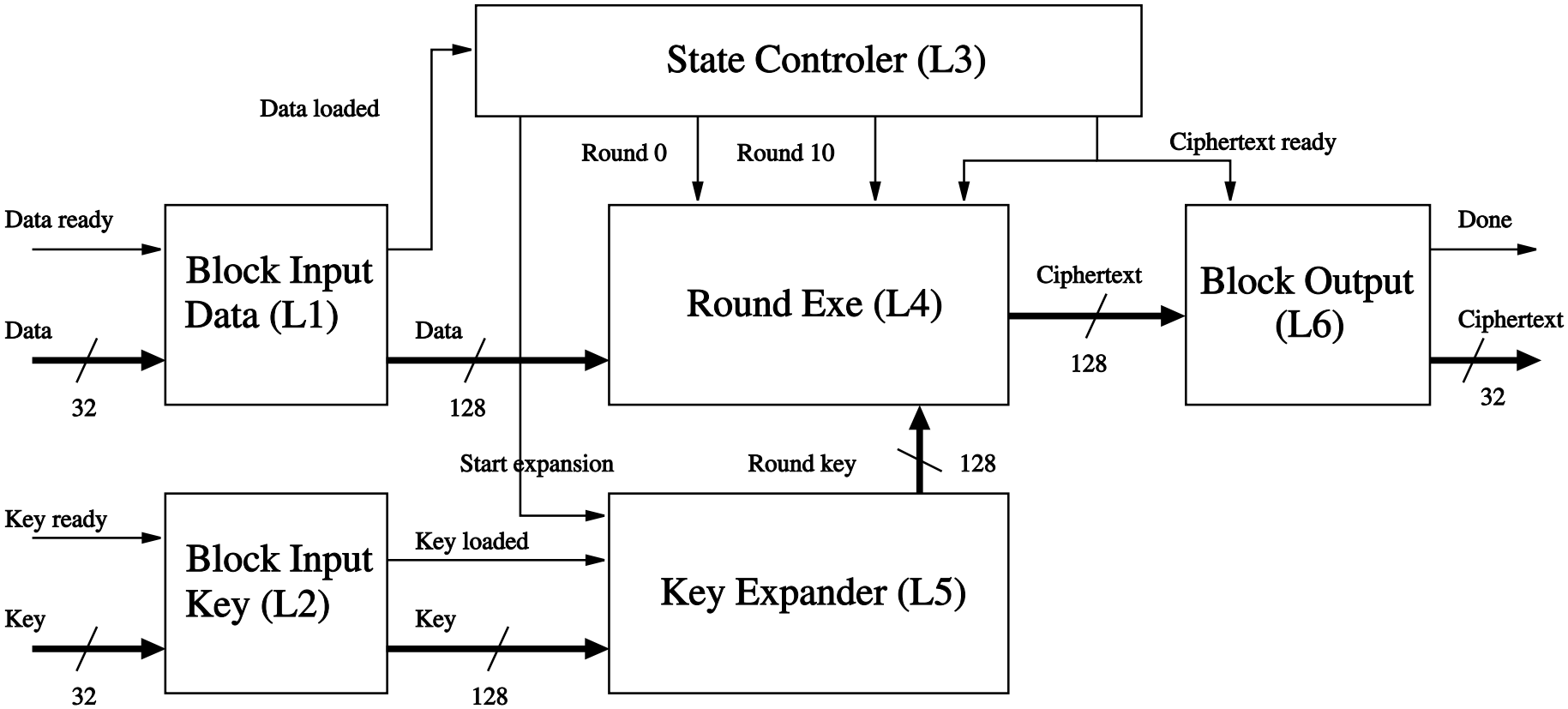
Courbes elliptiques

- Algorithme asymétrique publié en 1985
- Clés beaucoup plus courtes (384 bits)

AES : Algorithme

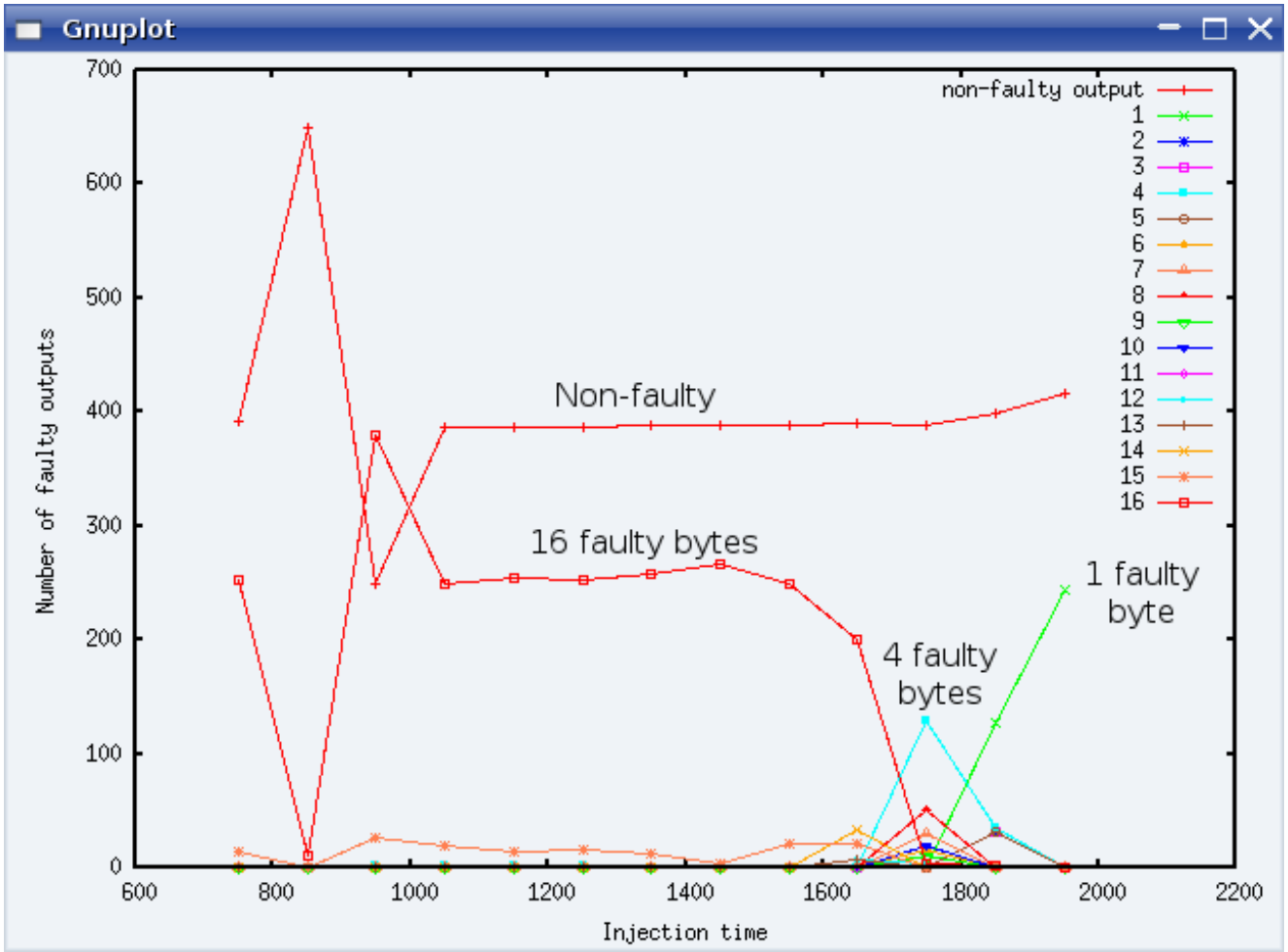


AES : Implémentation



Injection exhaustive sur l'AES

8645 injections, 39,4% de résultats faux en moyenne (1,5% exploitable)



Délais des cônes logiques de l'AES (en ns)

Type	min.	moy.	max.
Données	10.24	11.11	11.71
Buffer de sortie	10.51	11.60	12.28
Calcul de clé	7.35	9.20	10.88
Autres	0	0.15	3.80

Différentes implémentations de l'AES

Différents SubBytes

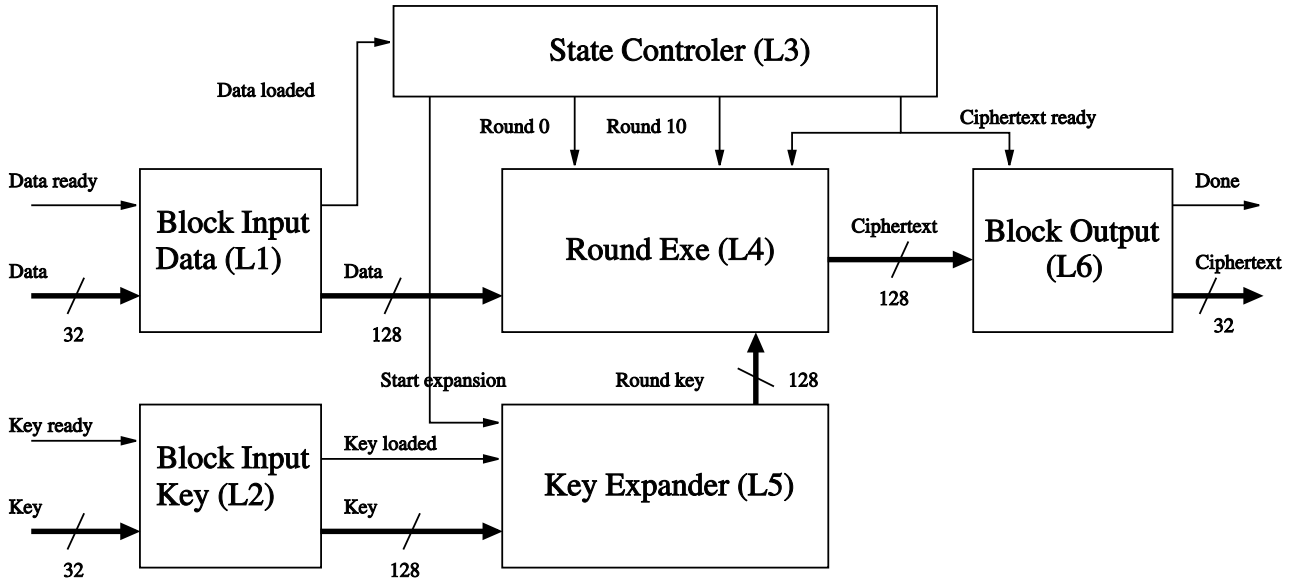
- Table de correspondance : Look-up table
- Arbre et-ou
- Opérations dans $GF(2^4)$

Différents MixColumns

- Addition conditionnelle
- Addition non-conditionnelle
- Réduction classique
- Opérations dans $GF(2^4)$

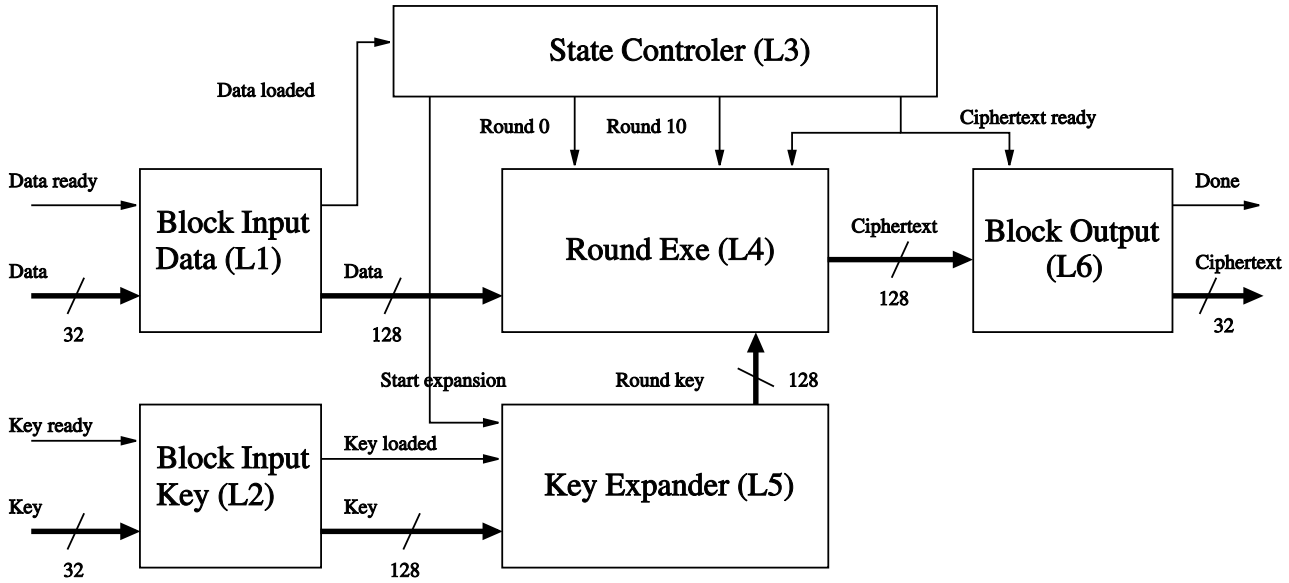
Différences de délai entre les données et la clé

SubBytes	MixColumns	Diff. délai (ns)
Look-up table (LUT)	Addition cond.	1.76
Arbre et-ou		1.91
Opérations dans $GF(2^4)$		**2.39**
Look-up table (LUT)	Addition non-cond.	**1.39**
Arbre et-ou		1.59
Opérations dans $GF(2^4)$		1.65



Différences de délai entre les données et la clé

SubBytes	MixColumns	Diff. délai (ns)
Look-up table (LUT)	Réduction classique	**0.88**
Arbre et-ou		1.08
Opérations dans $GF(2^4)$		2.15
Look-up table (LUT)	$GF(2^4)$	7.39
Arbre et-ou		7.00
Opérations dans $GF(2^4)$		**7.43**



Attaques par faute sur l'AES

Attaque	**Type de faute requis**
Giraud 1	1 bit de donnée
Choukri et Tunstall	1 bit de donnée
Blömer and Krummel	1 bit de donnée
Giraud 2	1 octet de donnée ou de clé
Chen et Yen	1 octet de clé
Piret et Quisquater	**1 octet d'une colonne de donnée**
Dusart, Letourneux et Vivolo	1 octet de donnée
Blömer et Seifert	1 octet de donnée
Moradi, Shalmani et Salmasizadeh	**max. 3 octets d'une colonne de donnée**

Précision temporelle nécessaire

MixColumns : $GF(2^4)$

SubBytes : Arbre et-ou			
18.23	17.36	17.36	16.98
17.62	18.02	18.02	17.17
17.59	18.11	18.11	17.37
18.71	17.96	17.93	16.27
Max1-Max2, Max-min			
0.47	0.09	0.09	0.21
1.11	**0.75**	**0.75**	**1.10**

SubBytes : $GF(2^4)$			
26.50	26.10	26.10	26.98
27.37	27.69	27.69	27.27
28.27	28.15	28.15	26.77
28.16	27.50	27.49	27.90
Max1-Max2, Max-min			
0.12	**0.46**	**0.46**	**0.63**
1.77	**2.05**	**2.05**	**1.13**

- Max1-Max2 : Attaque Piret-Quisquater
- Max-min : Attaque Moradi, Shalmani et Salmasizadeh

Résultats : vulnérabilité des implémentations

MixColumns

- Le MixColumns en $GF(2^4)$ est plus sensible que les autres implémentations
- Le MixColumns en addition non-conditionnelle est légèrement moins sensible

SubBytes

- Pas de tendance générale
- Dépend du MixColumns

Conclusion

Bilan

- Méthodologie d'évaluation de la sécurité des circuits face aux attaques par faute
- Automatisation de l'analyse du circuit, de l'injection de faute et du calcul des résultats
- Application à l'estimation de la sécurité de plusieurs implémentations d'AES face aux fautes de délai
- Publications (NordSec, JCSC, ...), brevet en cours de dépôt

Perspectives

- Analyse du modèle de faute sur circuit réel
- Prise en compte des fautes multiples
- Application à d'autres circuits de sécurité (ex : cryptographie asymétrique)