Détection d'Erreur dans les Architectures de Chiffrement

Paolo Maistri

Pourquoi ?

- Les attaques par fautes exploitent les résultats erronés
  - Il faut détecter la faute avant la sortie du résultat
- Le domaine de la détection d'erreurs a une histoire longue avec beaucoup de solutions…
  - … mais les algorithmes de chiffrement sont fortement non linéaires
  - Les fautes peuvent être d'origine intentionnelle : un modèle de fautes complexe
- Il est donc nécessaire d'adapter les techniques connues au nouveau domaine
  - Premières solutions sans modèle de fautes clair !

L'Injection de fautes

- Sur l'alimentation
  - Survoltage (corruption des données) ou sous-alimentation (fautes de délai)
- Sur l'horloge
  - Mauvais échantillonnage des valeurs
- Tirs laser
  - Corruption des données dans les registres (ASIC) / Altération de la configuration (FPGA)
- EM
- …

⇒ Erreurs multiples !

⇒ Changement de la fonction calculée !

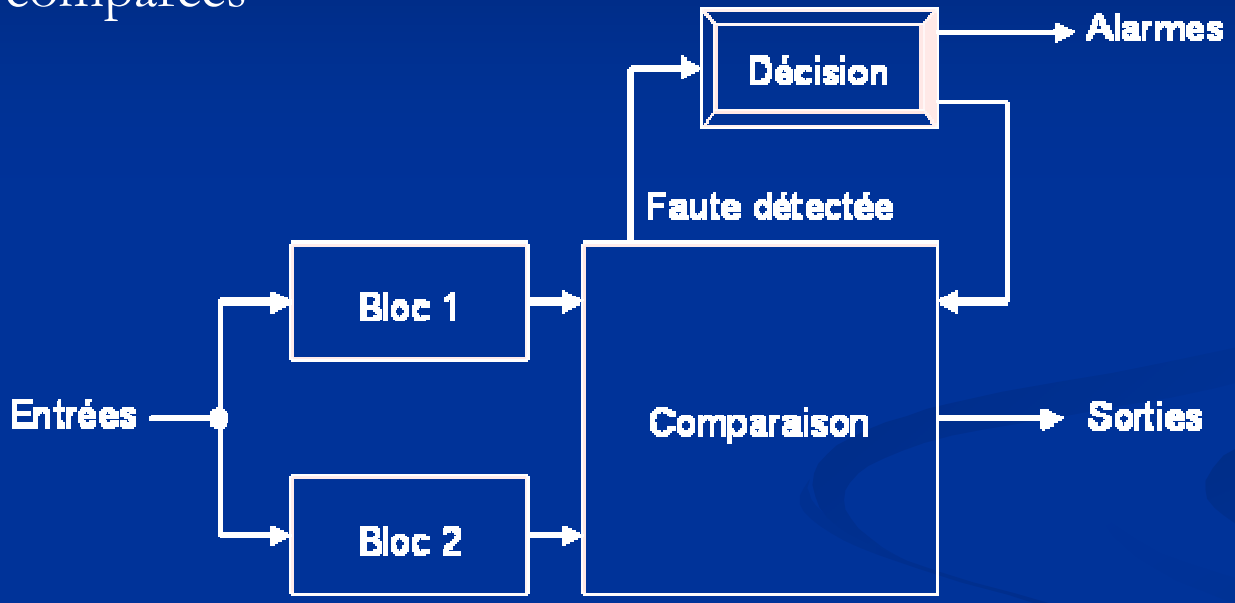
Table des Matières

- Redondance
  - matérielle
  - d'information
  - temporelle
- Autres protections
- Discussion et conclusion



Redondance Matérielle

- Plusieurs (différents) blocs fonctionnels sont implantés et les sorties sont comparées



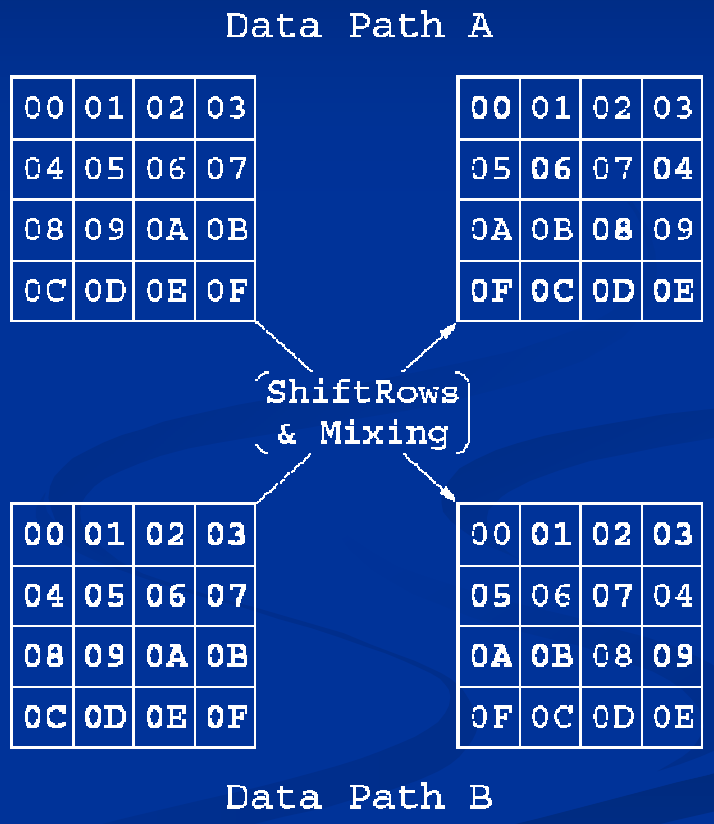
☺ Efficace contre fautes transitoires ou destructives

☹ Surcoût plus que double en surface et consommation

☹ Encore plus de corrélation entre données et puissance consommée

Redondance et Diffusion

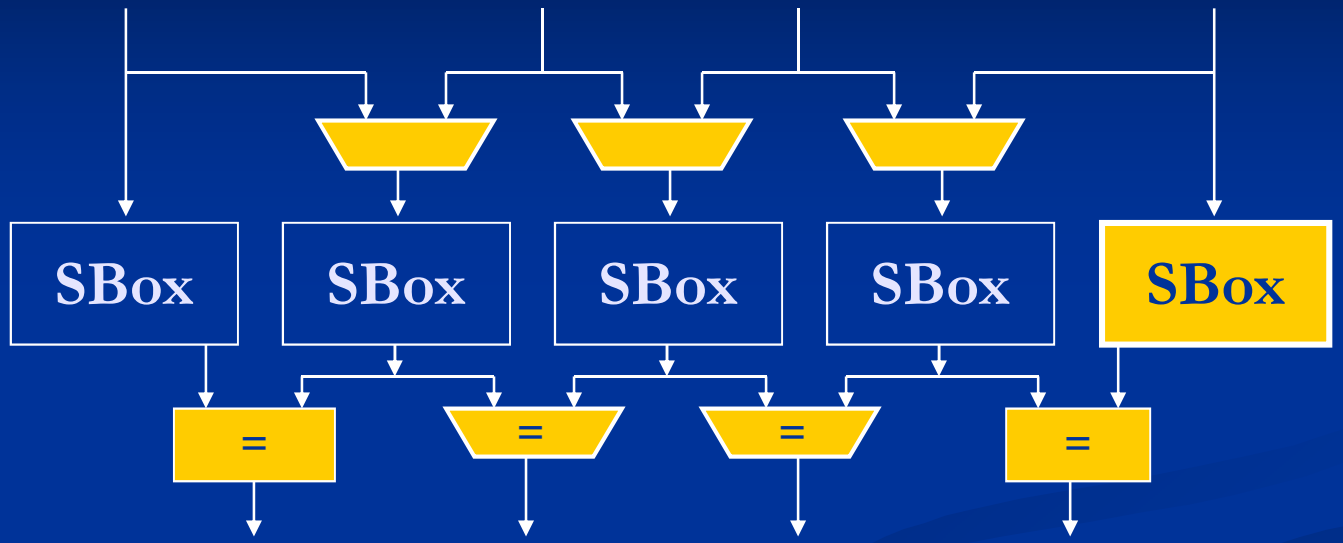
- Duplication du matériel pour la détection de fautes
- Contamination des registres pour éviter l'exploitation
- Mêmes problèmes et avantages que la duplication simple
  - Sécurité majeure si la validation finale est compromise



[Joye, Manet, Rigaud. IET Inf Sec 2007]

TIMAlaboratory

Redondance Partielle (1/2)

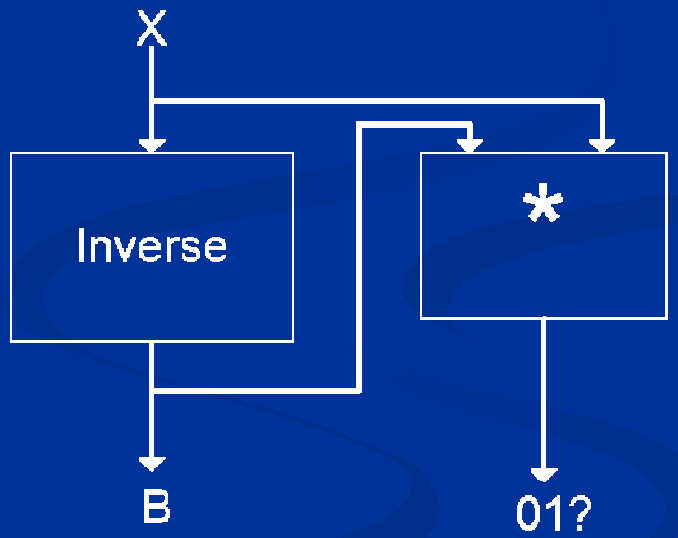


- Comparaison dynamique avec composant additionnel
- ☺ Surcoût limité et bonne détection de fautes permanentes
- ☹ Détection limitée de fautes transitoires

[Di Natale, Flottes, Rouzeyre. WDSN 2007]

Redondance Partielle (2/2)

- La S-Box est
  - le composant le plus complexe et coûteux
  - non linéaire
  - basé sur le calcul de l'inversion
- Détection basée sur calcul de la multiplication inverse
  - Ajout d'un multiplieur *GF*
  - Nombre de bits configurable



☹ Couverture limitée au bloc S-Box
☹ Couverture dépendant du surcoût
☹ Code linéaire requis pour le reste du circuit

[Kulikowski, Karpovsky, Taubin. CARDIS 2004]

Redondance Matérielle - Résumé

- On utilise deux (ou plus) blocs fonctionnels indépendants et on compare les résultats
  - On peut limiter la protection à une sous partie
  - On peut croiser les chemins
- Efficace contre fautes transitoires et permanentes
- Très chère en surface, mais pas de réduction de débit
- Particulièrement vulnérable aux attaques par canaux cachés

Redondance d'Information

- AES
  - Parité au niveau octet
  - Parité au niveau bloc
  - Autres parités
  - Codes non linéaires
- Chiffreurs symétriques
  - Opérations
  - Fréquence
  - Granularité
- Chiffrement à clé publique

Error Detecting Codes (EDCs)

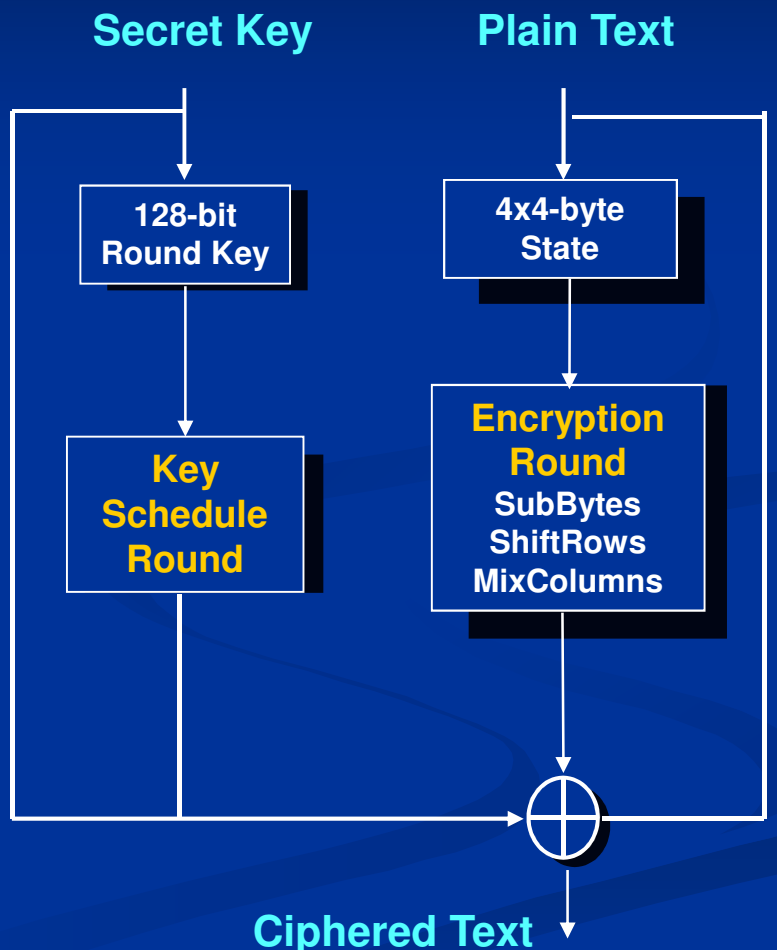
- Première approche contre les attaques par fautes
- Très bonne couverture des fautes simples
  - Souvent 100 %
- Couverture des fautes complexes dépendante du niveau de redondance

Règles de Prédiction

- Choix du code
  - Complexité de l'algorithme
  - Prédiction pour toutes les opérations
- Surcoût matériel devant être inférieur à celui de la duplication
  - Génération simple du code
  - Règles de prédiction simples et complètes
- Efficacité vs. coût
  - Par rapport à une menace caractérisée

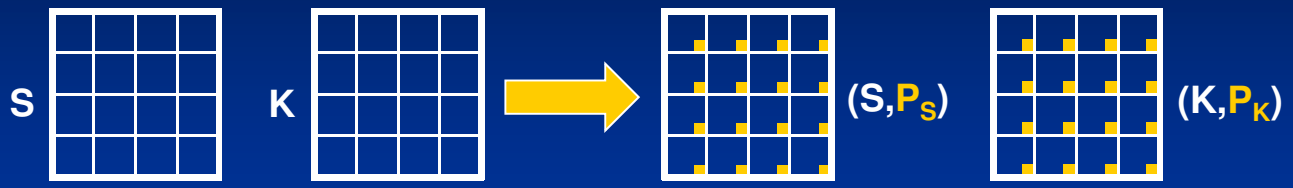
Advanced Encryption Standard

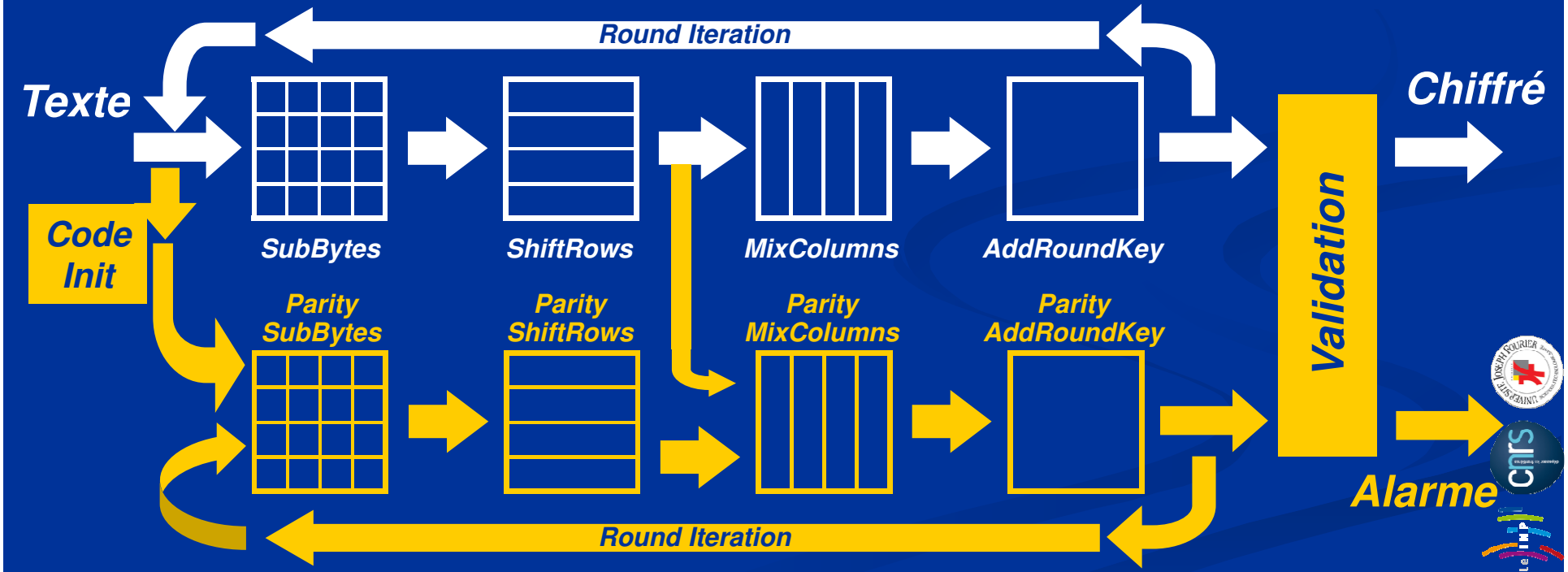
- Entrée 128 bits, clé privée 128 à 256 bits
- Algorithme itératif :
  - **SubBytes** : substitution d'octet non linéaire
  - **ShiftRows** : rotation d'un mot
  - **MixColumns** : transformation linéaire d'un mot
  - **AddRoundKey** : addition modulo-2 avec la clé de ronde
- Chiffrement et Key Schedule utilisent les même opérations de base
  - Déchiffrement utilise les opérations inverses…



TIMAlaboratory

AES – Parité Multiple





[Bertoni et al. TC 2003]

TIMA laboratory

AES – Parité Simple

- L'étage de permutation dans l'AES ne change pas la parité globale de la matrice d'état
  - L'opération *MixColumns* préserve en fait la parité
- La propagation d'un seul bit de parité n'a donc pas besoin de logique additionnelle
- L'étage de substitution pose des problèmes :
  - Impossible de propager la parité à travers les S-Box !
  - On vérifie le code avant la S-Box, …
  - … et on régénère le code après la S-Box

☹ Limité aux fautes simples (transitoires et permanentes)

[Karri et al. CHES 2003]

Chiffreurs à Blocs (1/2)

Chiffreur	$\oplus$	$\wedge,\vee$	$+,-$	$\times$	Sbox	Rot	Sh	Perm	$\times$ mod G(x)
Camellia	✓	✓			✓	✓		✓	
DES	✓				✓			✓	
IDEA	✓		✓	✓				✓	
MARS	✓		✓	✓	✓	✓		✓	
RC5	✓		✓			✓		✓	
RC6	✓		✓	✓		✓		✓	
Rijndael	✓				✓			✓	✓
Serpent	✓				✓	✓	✓		
Twofish	✓		✓		✓	✓		✓	✓

Chiffreur	Code suggéré	Chiffreur	Code suggéré
AES	Parité, par octet	RC5	Parité ou résidu
DES	Parité	RC6	Résidu
IDEA	Résidu, mais cher	Serpent	Parité, par octet
MARS	Résidu, mais cher	Twofish	Parité, par octet

Chiffreurs à Blocs (2/2)

- Les codes de détection doivent être peu chers
  - Génération / Propagation / Validation
- …et efficaces contre les fautes intentionnelles !
- Plusieurs paramètres à considérer
  - Le modèle d'erreurs …
  - L'algorithme de chiffrement …
  - Le type de code …
  - Le niveau de redondance …
  - La fréquence des points de contrôle …
  - La distribution du code par rapport aux données …
- Il y a un compromis entre le coût et la sécurité
  - Pour l'AES, la parité ne protège pas vraiment bien …

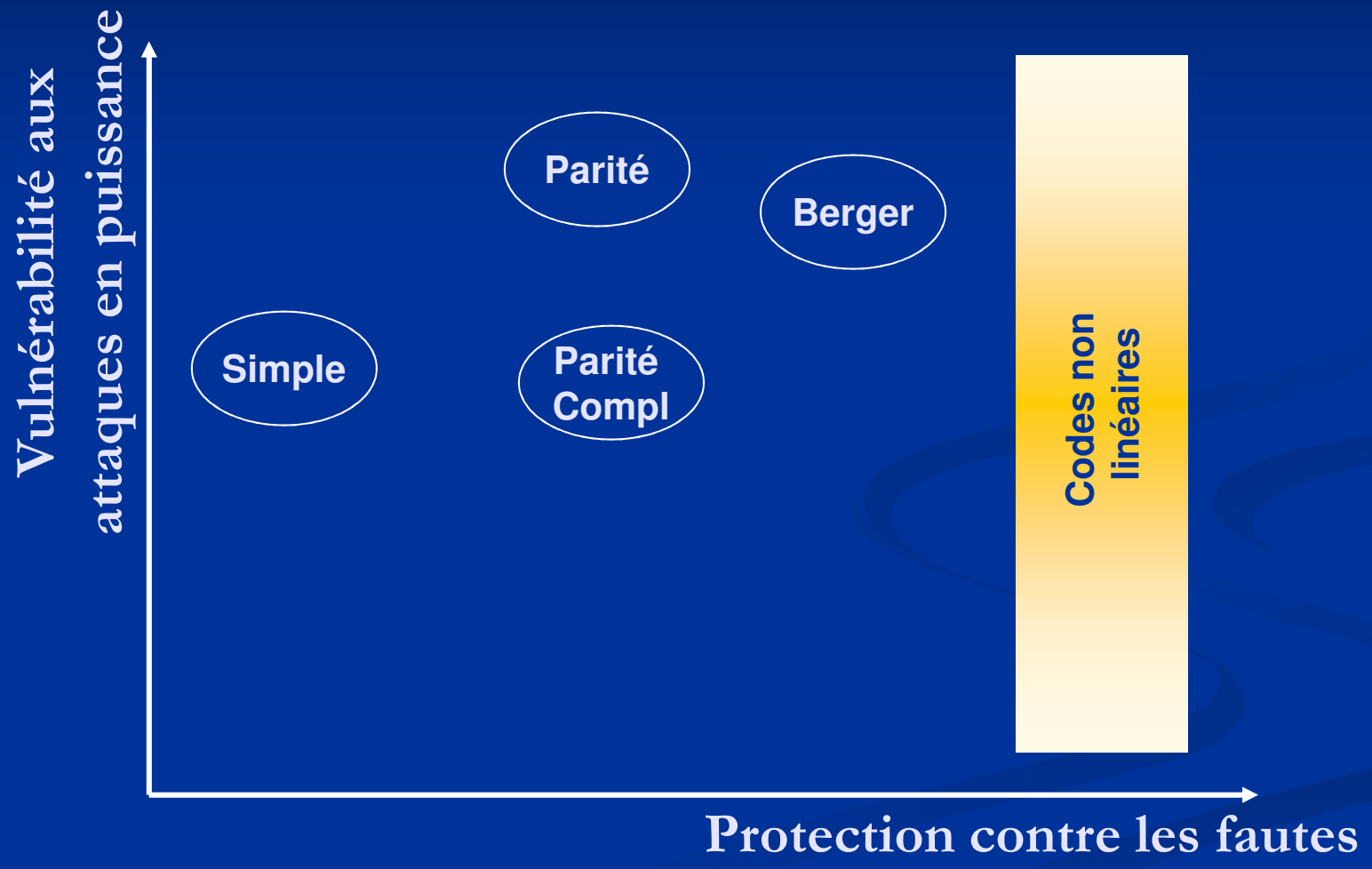


Encore sur la parité…

- La redondance matérielle ou d'information rend les attaques en puissance plus faciles
  - +35% bits trouvés grâce à la parité (1 bit par octet) [Regazzoni et al., DFT07]
- En codant la parité en dual rail (parité complémentaire), on peut réduire la corrélation entre le code et la puissance consommée [Maingot et Leveugle, ICECS 2006]

Code	Corrélation		
	Encodeur	Registre	Décodeur
Parité simple	0,995	0,927	0,977
Parité double	0,998	0,799	0,952
Parité complémentaire	0,012	0,334	0,056
Code Berger	0,989	0,999	0,994

Détection vs Sécurité

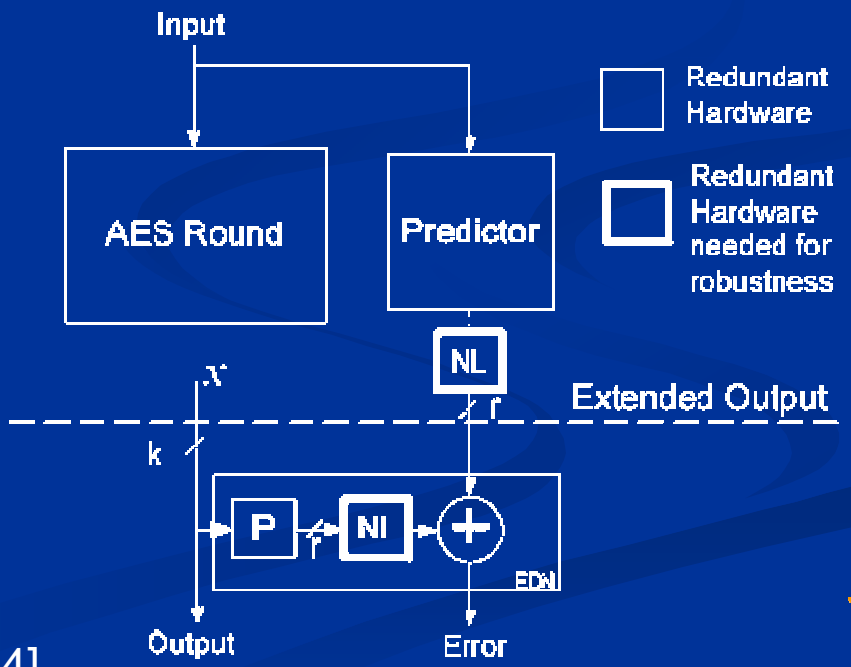


TIMA laboratory

Codes Non Linéaires

- La détection par un code linéaire ne dépend que de la valeur de l'erreur
  - Vulnérabilité potentielle
- La détection par un code **non** linéaire dépend aussi des données
  - Taux de détection plus uniforme
  - L'attaquant ne peut pas injecter une erreur ciblée

☺ Détection très élevée
☺ Injection de fautes ciblées très difficile
☹ Surcoût non négligeable (+70%)



[Kulikowski, Karpovsky, Taubin. DSN 2004]

Variations sur le Thème …

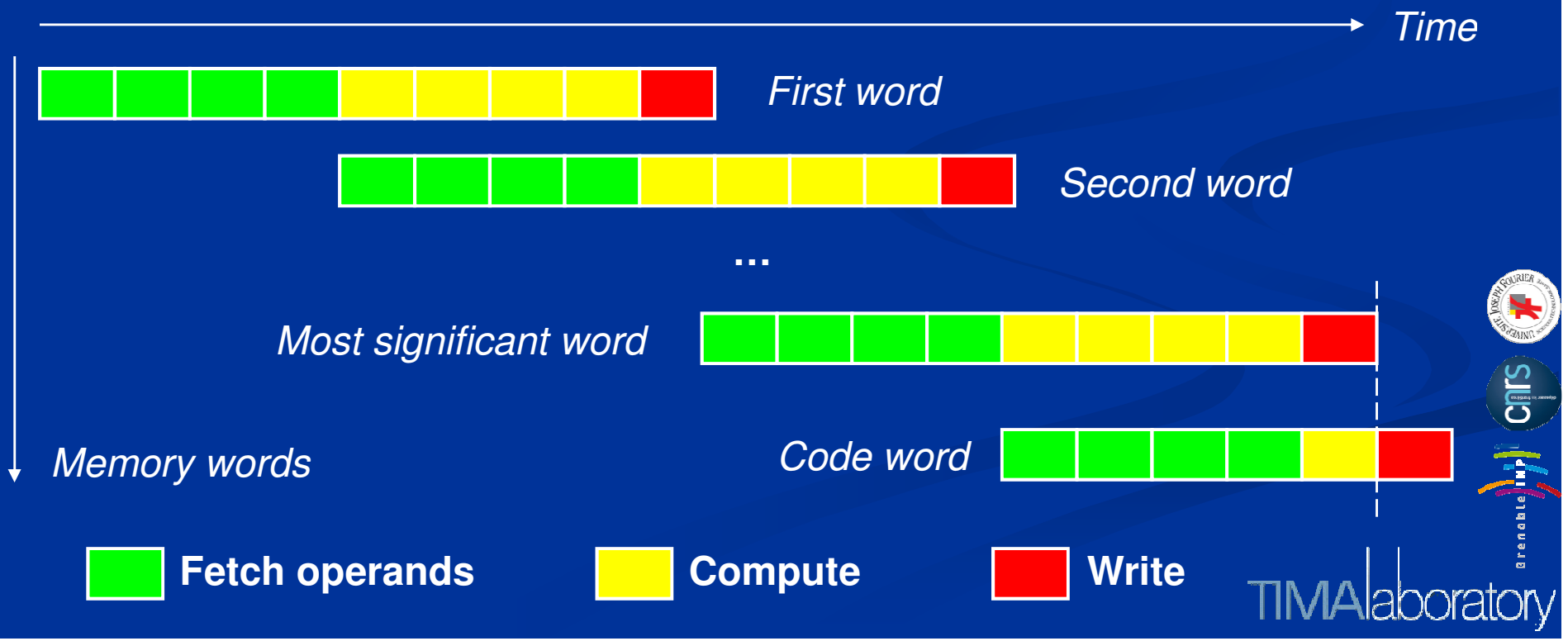
Auteurs	Publication	Notes
Yen, Wu	TC 2006	Protection du chemin AES par code de redondance CRC
Kermani, Masoleh	DFT 2006	Prédiction de la parité avec décomposition de la Sbox
Cota et al.	ISCAS 2008	Codes de Hamming et Reed Solomon
Di Natale et al.	IOLTS 2007	La prédiction de la parité prend en compte l'entrée et la sortie de la Sbox
Kermani, Masoleh	CHES 2008	Décomposition de la Sbox avec base normale
…		

Systèmes à Clé Publique

- Les attaques par fautes s'appliquent aussi aux systèmes de chiffrement asymétriques
  - CRT-RSA est vulnérable avec une seule faute
- La vérification du message chiffré demande une nouvelle opération de chiffrement (exponentiation)
  - Potentiellement très lent
  - A gérer dans la puce, car il ne faut pas sortir un résultat fauté !
- Les systèmes asymétriques sont fortement basés sur l'arithmétique modulaire
  - Les erreurs de calcul peuvent être identifiées efficacement avec les codes de détection
  - Des codes non linéaires peuvent aussi être utilisés [Gaubatz et al., FDTC 2006]

RSA – Codes Modulaires

- ***Detection and correction of transient errors in a hardware implementation of the RSA cryptosystem [...] can be done efficiently and reliably with acceptable time and area costs equivalent to an increase in the size of the modulus by one digit or less*** [C. Walter, CHES 2000]



Redondance d'Information - Résumé

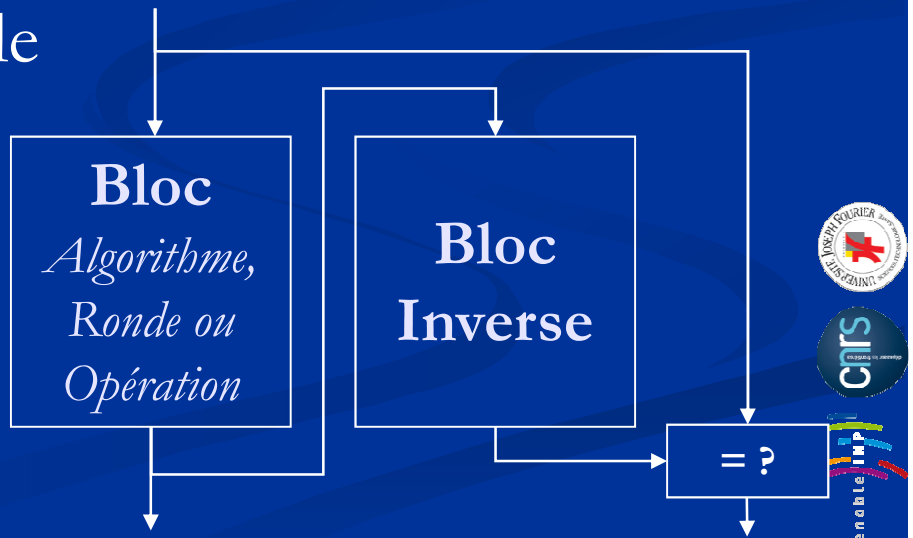
- On utilise des codes de détection d'erreur
- Niveau de détection de moyen à très bon
  - Plutôt conçus pour les fautes naturelles, ils deviennent assez chers contre les attaques intentionnelles
- Niveau de redondance souvent configurable
  - Compromis entre détection et coûts
- Latence de détection minimale
- Vulnérabilité aux attaques par canaux cachés

Redondance Temporelle

- Répétition du calcul
- Calcul de l'opération inverse
  - Chiffreurs involutifs
- Exploitation du pipeline
- Calcul inverse par pipeline
- Calcul par Double Data Rate

Calcul inverse

- On exploite les différents chemins de chiffrement et déchiffrement
- 3 niveaux de granularité : algorithme, ronde, opération
  - Surcoût, réduction du débit et latence de détection
- La fonction inverse est calculée et le résultat est comparé avec la valeur initiale



😐 Déchiffrement requis
☺ Fautes transitoires et permanentes
☹ Latence et surcoûts

[Karri et al. DAC 2001]

TIMAlaboratory

Chiffreur Involutif

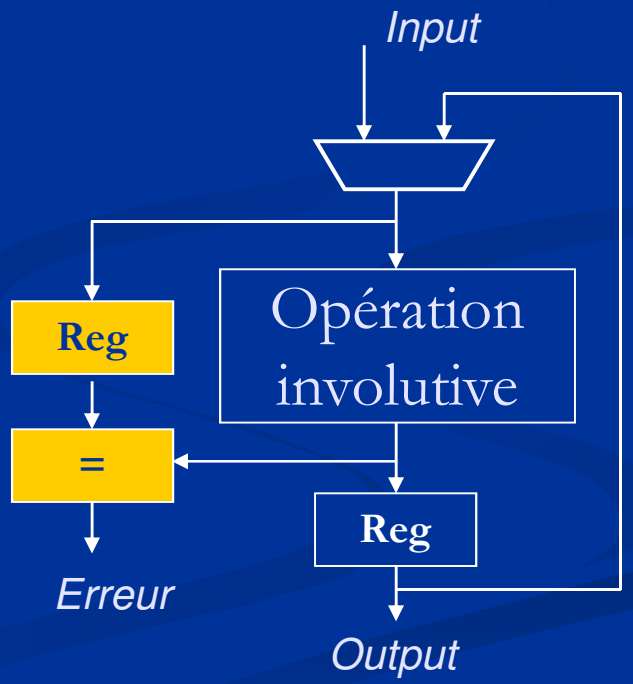
- Chiffreur involutif : la fonction de chiffrement et de déchiffrement coïncident
- Même principe que le calcul inverse…

☺ La logique de déchiffrement est déjà présente

☺ Fautes transitoires et permanentes

☹ 100 % temps de calcul

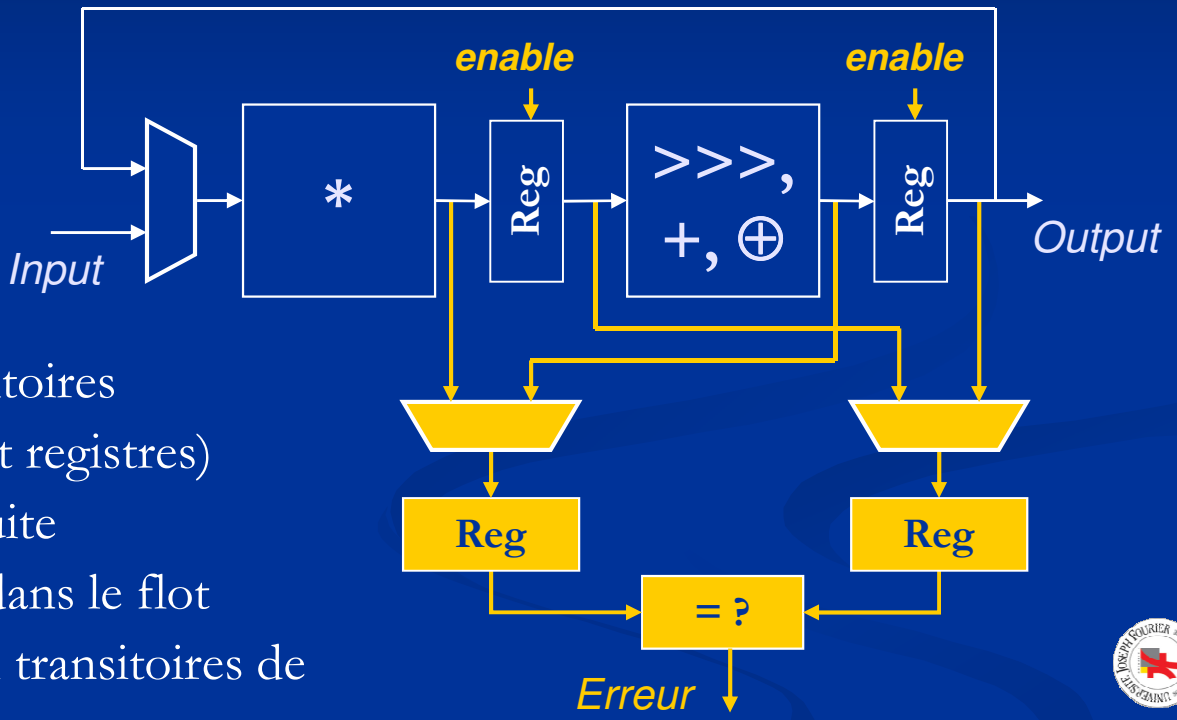
Améliorable avec une approche par pipeline…



[Joshi et al., CHES 2004]

Redondance par Pipeline

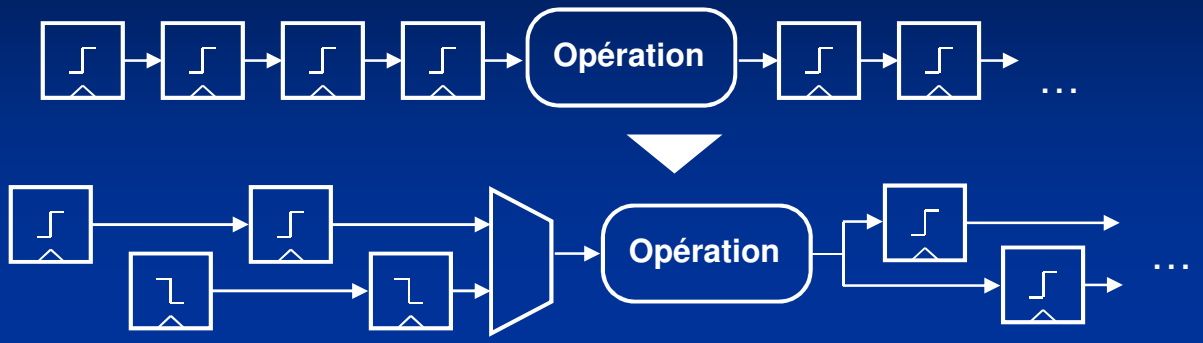
- **On utilise les étage inutilisés d'un pipeline pour refaire le calcul une deuxième fois**



- ☺ Détection de fautes transitoires
- ☺ Surcoût limité (contrôle et registres)
- ☺ Latence de détection réduite
- ☺ Pipeline facile à intégrer dans le flot
- ☹ Ni fautes permanentes, ni transitoires de longue durée (>1 cycle)
- ☹ Nécessité d'un pipeline partiellement inutilisé

[Wu, Karri. DFT 2001]

Double Data Rate (DDR)



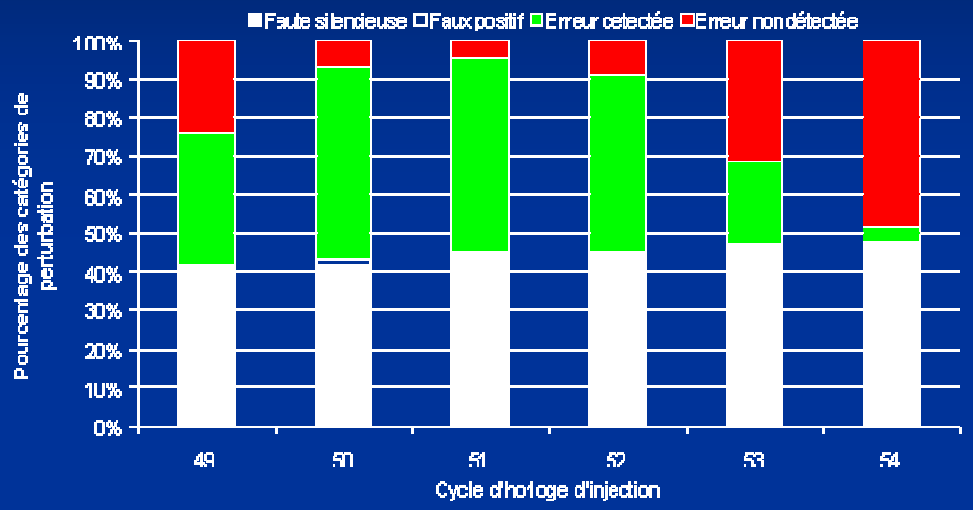
- ☺ Surcoût acceptable
- ☺ Fautes transitoires longues
- ☹ Conception délicate
- ☹ Fautes permanentes

Cycle	1	2	3	4	5
Pipeline classique	D1	D2 D1	D3 D2	D4 D3	D4
Redondance par pipeline	D1	D1 D1	D2 D1	D2 D2	D2
Redondance DDR	D1 D2	D3 D4 D1 D2	D1 D2 D3 D4	D3 D4 D1 D2	D3 D4

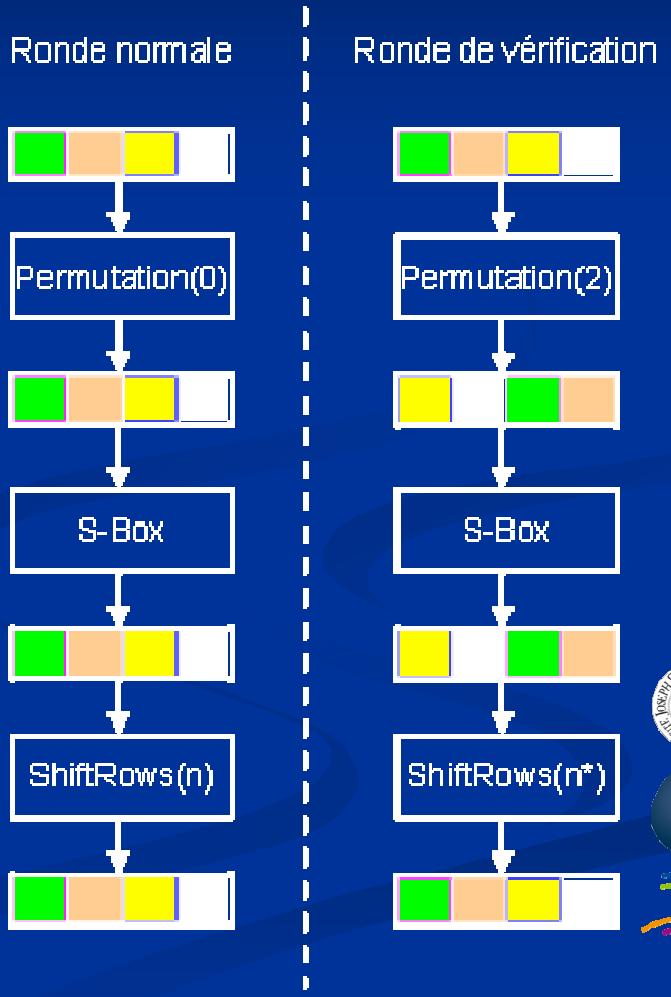
[Maistri, Leveugle. TC 2008]

TIMA laboratory

Redondance Temporelle & Fautes Permanentes…



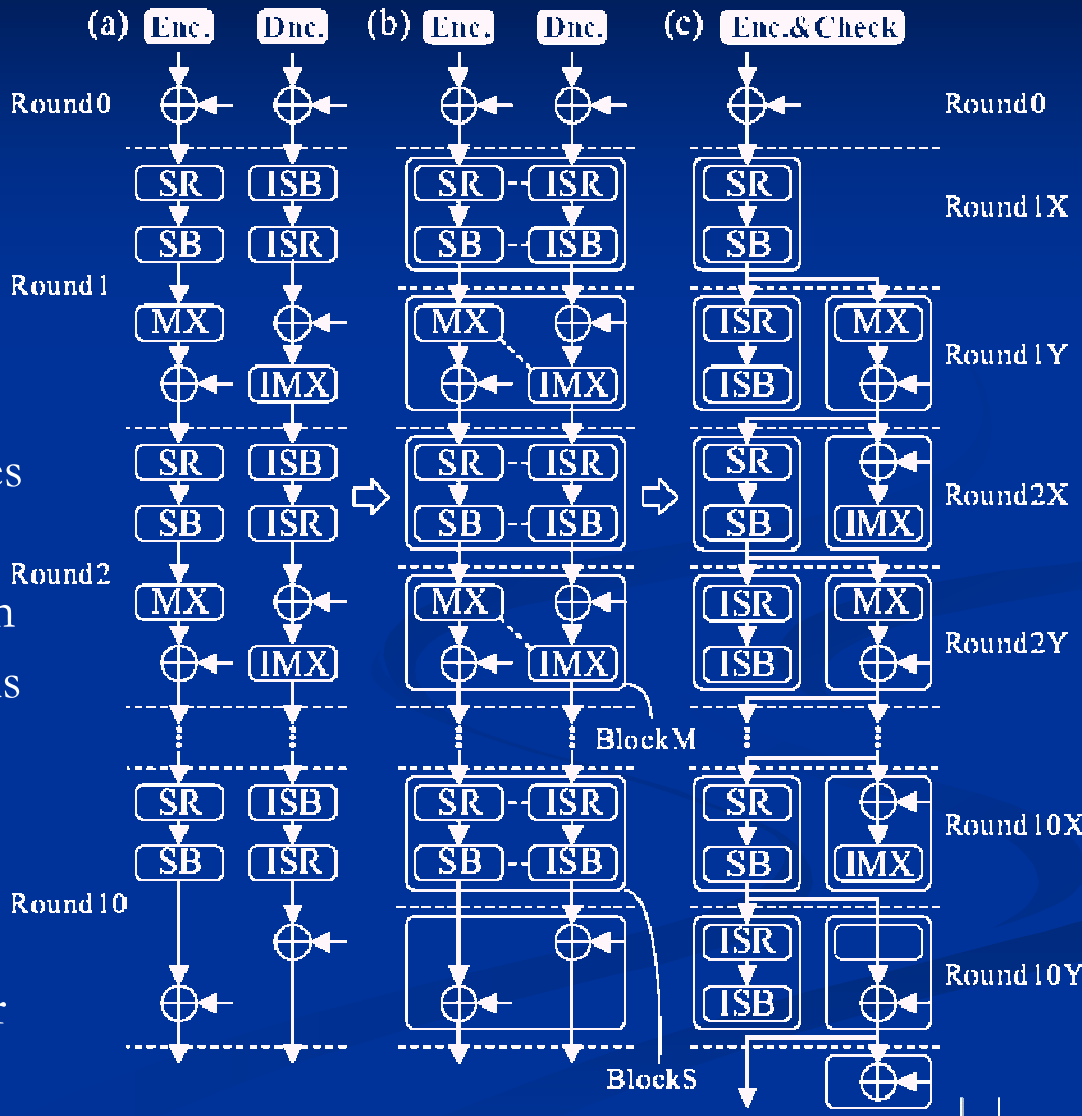
Résultat de l'expérience	Pourcentage
Tirs Sans Effets	54,6 %
Configuration modifiée	45,4 %
Faute Silencieuse	39,6 %
Faux Positif	2,3 %
Erreur Détectée	57,9 %
Erreur Non Détectée	0,3 %



TIMA laboratory

Calcul Inverse par Pipeline

- **On combine le calcul inverse avec la répétition par pipeline**



- ☺ Détection de fautes transitoires et permanentes
- ☺ Surcoût limité
- ☺ Baisse latence de détection
- 😐 Assez facile à intégrer dans le flot de conception
- 😐 Nécessité de la logique de déchiffrement
- 😐 Pipeline requis
- 😐 Autres chiffreurs à vérifier

[Satoh et al. CHES 2008]

TIMA laboratory

Redondance Temporelle - Résumé

- Répétition du calcul ou de son inverse
- Détection des fautes transitoires
  - Fautes permanentes détectées par calcul inverse
- Coût en surface et réduction de débit limités avec pipeline
- Pas de vulnérabilité majeure contre les attaques par canaux cachés
  - si l'architecture de base est elle-même protégée

Comparaison (1/2)

Redondance	Faute simple	Fautes multiples	Attaques par fautes	Surcoût surface	Réduction débit
Matérielle	100%	☺	☺	> 100 %	-
Matérielle partielle	☺	☺	☹	25 – 35 %	-
Parité par octet	100 %	> 99 %	50 – 99 %	20 – 30 %	3 %
Parité par bloc	98 %	☹	50%	18 – 24 %	
Code non linéaire	$1 - 2^{-56}$	$1 - 2^{-56}$	☺	77 %	13 %

Permanentes

Transitoires

TIMAlaboratory

Comparaison (2/2)

Redondance	Faute simple	Fautes multiples	Attaques par faute	Surcoût surface	Réduction débit
Opération Inverse	100 %	☺	☺	19 – 38 %	23 – 61 %
Pipeline	0 %	☹	☺	50 %	18 %
Double Data Rate (DDR)	0 %	☹	☺	36 %	15 – 55 %
DDR + Réallocation	> 90 %	☺	☺	~ 36 %	~ 15 – 55 %
Inverse avec pipeline	100 %	☺	☺	**-21** – 24 %	~ 25 %

Permanentes

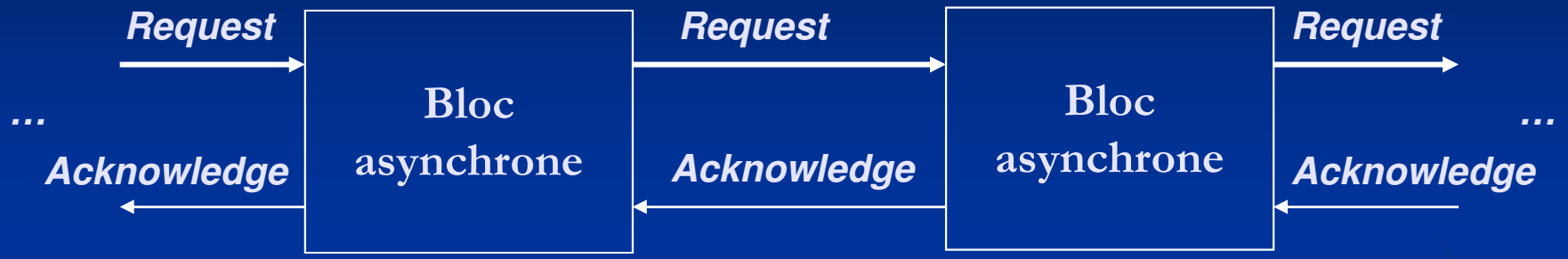
Transitoires

TIMA laboratory

Prévenir est Mieux que Guérir

- Au lieu de détecter l'erreur, on intervient avant :
  - Filtres sur l'alimentation
  - Détecteurs de surtension ou de glitch
  - Capteur de température
  - Capteur de lumière
  - Protections métalliques
  - Chiffrement de la mémoire
  - Brouillage des adresses
  - …

Logique Asynchrone



- Communication basée sur un mécanisme d'échange avec acquittement
  - Plus de signal d'horloge global !
- Nettement plus robuste par rapport aux
  - Perturbations (fautes de délai, …)
  - Attaques par observation (consommation équilibrée)

Quelques Perspectives

- Meilleure modélisation d'erreurs réalistes
- Analyse des contre-mesures par rapport aux autres attaques
  - Corrélation avec la puissance consommée ?
  - Est-ce que la détection est exploitable elle-même ?
- Intégration dans le flot de conception
- …