ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2012-ENST-083

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Électronique et Communication »

présentée et soutenue publiquement par

Houssem MAGHREBI

le 21 décembre 2012

Les contre-mesures par masquage contre les attaques

HO-DPA : évaluation et amélioration de la sécurité en utilisant des encodages spécifiques

Directeur de thèse : **Jean-Luc DANGER**
Co-encadrement de la thèse : **Sylvain GUILLEY**

Jury

M. Louis GOUBIN, Professeur, Université de Versailles	Président
M. Claude CARLET, Professeur, Université PARIS 8	Rapporteur
M. David NACCACHE, Professeur, École Normale Supérieure	Rapporteur
M. François-Xavier STANDAERT, Professeur, Université Catholique de Louvain	Examinateur
M. Emmanuel PROUFF, Expert en sécurité, ANSSI	Examinateur

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech

THÈSE

à mes très chers parents ...

Abstract

Side Channel attacks are nowadays well known and most designers of secure embedded systems are aware of them. Since the first public reporting of these threats in 1996, a lot of effort has been devoted towards the research about side channel attacks and the development of corresponding countermeasures. Side channel attacks take advantage of the fact that the power consumption of a cryptographic device depends on the internally used secret key. Since this property can be exploited with relatively cheap equipment, these attacks pose a serious practical threat to cryptographic embedded systems. A very common countermeasure against side channel attacks is masking. It consists in splitting the sensitive variable of cryptographic algorithms into random shares (the masked data and the random mask) so that the knowledge on a subpart of the shares does not give information on the sensitive data itself. However, other attacks, such as higher-order side channel attacks, can defeat masking schemes. These attacks consist in combining the shares in order to cancel (at least partially) the effects of the mask.

The overall goal of this thesis is to give a deep analysis of higher-order attacks and to improve the robustness of masking schemes. The first part of this thesis focuses on higher-order attacks. We propose three novel distinguishers. Theoretical and experimental results show the advantages of these attacks when applied to a masking countermeasure. The second part of this thesis is devoted to a formal security evaluation of hardware masking schemes. We propose a new side channel metric to jointly cover the attacks efficiency and the leakage estimation. In the last part, we propose three novel masking schemes remaining more efficient than the state-of-the-art masking. They remove (or at least reduce) the dependency between the leakage and the sensitive variable when the leakage function is known (*e.g.* the Hamming weight or the Hamming distance leakage model). The new solutions have been evaluated within a security framework proving their excellent resistance against higher-order attacks.

Remerciements

En Octobre 2009, a débarqué à Télécom ParisTech un jeune étudiant timide plein d'ambitions. Il en repartira, en Janvier 2012, un jeune chercheur débutant beaucoup plus confiant en lui-même. Cette évolution est le fruit de nombreuses rencontres des personnes que je souhaite remercier ici pour toute l'aide, le soutien, les conseils qu'elles m'ont apportés ou tout simplement pour leur bonne humeur et leur joie de vivre qui ont fait de ces trois années de thèse, une aventure exceptionnelle.

J'adresse tout d'abord mes vifs remerciements à **Jean-Luc Danger** et **Sylvain Guilley** mes directeurs et encadrants de thèse. Merci à Jean-Luc pour m'avoir offert l'opportunité de faire cette thèse dans son groupe de recherche Systèmes Électroniques Numériques. Je le remercie d'avoir dirigé cette thèse et de m'avoir soutenu durant ces trois années. Je le remercie enfin autant pour ses qualités pédagogiques et son implication dans ce travail que pour ses qualités humaines, qui ont rendu chaque seconde au laboratoire toujours plus sympathique. Jean-Luc, t'es un grand frère. Je tiens à exprimer plus particulièrement ma profonde gratitude à Sylvain, pour la qualité de son encadrement et sa bienveillance tout au long de ce travail de recherche. J'avoue que j'ai eu la chance d'être encadré par un chercheur exceptionnel qui m'a transmis la passion de ce métier. En effet, Sylvain tu as le don de rendre abordable la compréhension des phénomènes aussi complexes soient-ils. Merci pour les longues heures passées ensemble devant le PC jusqu'à ce qu'on soit viré des locaux de l'école par le gardien. Sylvain, t'es un gladiateur. J'espère sincèrement que la collaboration que nous avons eue tous les trois se poursuivra dans le futur.

Je suis reconnaissant à **Louis Goubin** pour l'honneur qu'il m'a fait de présider le jury de ce travail. Il me tient à coeur d'exprimer aussi toute ma gratitude à **David Naccache** pour le soin apporté à la lecture de mon manuscrit.

Ces trois années ont été ponctuées de nombreux échanges qui m'ont beaucoup apporté tant sur le plan professionnel que sur le plan personnel. Je remercie de tout coeur **Claude Carlet** et **Emmanuel Prouff**, membres de mon jury, amis et merveilleux co-auteurs. Au delà de m'avoir fait partager leur expertise scientifique, ils m'ont considérablement apporté en terme de méthodologie et de professionnalisme, ces qualités m'accompagneront tout au long de ma carrière, je leur en suis reconnaissant. Qu'ils trouvent ici l'expression de mes remerciements les plus sincères pour leur aide et disponibilité.

Je remercie très chaleureusement **François-Xavier Standeart** qui m'a honoré en acceptant d'être membre du jury et qui a su trouver le temps de me lire, de m'encourager et de m'éclairer de ses remarques judicieuses.

Mes remerciements vont aussi naturellement au reste de l'équipe Systèmes Électroniques Numériques à Télécom ParisTech. Que toutes ces personnes attachantes soient remerciées pour les mémorables moments partagés et chiffrés : **Shivam** (Fhal, VR SF), **Aziz** (paradise, terrasse), **Nidhal** (Ni), **Youssef** (Joe Bauer), **Taoufik** (Battle-field), **Sébastien** (Toxico man), **Salah** (petit Haj), **Zouha** (ma-

man à aziz) **Pierre** (rugby man), **Jerémie** (forestière), **Nicolas** (Nara), **Maxime** (papa), **Olivier** (zitouni), **Annelie** (salvia), **Thibault** (camrounais), **Robert** (détourneur d'avions), **Pablo** (PSG), **Tarek** (water-marking), **Sébastien** (l'english man), **Laurent** (cartho), **Guillaume** (chouquettes), **Philippe** (Enigma), **Molka** (non), **Florent** (Flou). Je suis aussi reconnaissant au staff administratif du département COMELEC : **Zouina**, **Florence**, **Chantal**, **Yvone**, **Bruno**.

Enfin, je n'oublierai pas mes amis les plus proches toujours aussi fidèles : **Tarek**, **Amine**, **Haythem**, **Salim**, **Nizar**, **Samir** ... Je garderai notamment toujours la saveur de ces fabuleux repas qu'on préparait, les voyages, les soirées, les matchs de foot, les parties de RAMI ... Merci mes frères.

Un grand merci plein de tendresse à **Ibtissem** ma bien aimée pour son soutien et son amour. Je t'adore.

Cette thèse, aboutissement de longues années d'études, je la dois beaucoup à mes parents exceptionnels **Nejib** et **Radhia**, mon frère **Mohamed** et ma soeur **Essia**. Il m'est impossible de trouver des mots pour dire à quel point je suis fier d'eux, et à quel point je les aime. Cette thèse vous est dédiée.

Résumé de la thèse en français

Les circuits électroniques réalisés avec les méthodes de conception assistée par ordinateur usuelles présentent une piètre résistance par rapport aux attaques physiques. Parmi les attaques physiques les plus redoutables figurent les attaques sur les canaux cachés, comme la "timing attack" ou la DPA, qui consistent à enregistrer une quantité physique (temps, consommation) fuie par le circuit pendant qu'il calcule. Cette information peut être exploitée pour remonter aux secrets utilisés dans des calculs de chiffrement ou de signature. Plusieurs méthodes de durcissement des circuits contre les attaques sur les canaux cachés ont été proposées. On peut en distinguer deux catégories :

1. Les contre-mesures par dissimulation (ou par logique différentielle), visant à rendre la fuite constante, donc statiquement indépendante des secrets.

2. Les contre-mesures par masquage, visant à rendre la fuite aléatoire, donc statistiquement indépendante des secrets.

La contre-mesure par masquage est la moins complexe et la plus simple à mettre en oeuvre, car elle peut s'appliquer au niveau algorithmique comme au niveau logique. Idéalement, le concepteur s'affranchit donc d'un placement-routage manuel, comme cela est le cas des contre-mesures statiques. En revanche, elle est la cible d'attaques du second ordre, voire d'ordre plus élevé, permettant d'exhiber le secret en attaquant plusieurs variables simultanément.

Cette thèse se fixe comme objectifs l'analyse en robustesse et complexité des implémentations de contre-mesures par masquage et la proposition des nouvelles structures de masquage qui permettent de faire face aux attaques d'ordre élevé.

Ce manuscrit de thèse s'articule en cinq parties que nous détaillerons dans ce qui suit.

Première Partie : Préliminaires

Dans cette première partie, nous exposons les bases de la cryptographie moderne et développons un peu plus la question de leur implémentation sur les circuits cryptographiques (carte à puce, FPGA) et de la fuite d'information qui accompagne sa mise en oeuvre. Ensuite, nous introduisons la notion des attaques par canaux cachés et nous présentons quelques modèles classiques utilisés pour leur analyse. Enfin, nous décrivons les principales attaques et contre-mesures existantes dans la littérature et nous mettons l'accent sur le contexte d'attaques d'ordre supérieur.

Chapitre 1 : Contexte Général

La cryptographie est définie comme la science de la protection des informations sensibles. Elle se repose sur l'étude des méthodes qui permettent d'envoyer des messages à

l'aide d'une information secrète *key* (*chiffrement / cryptage*) de telle sorte que seul le destinataire puisse lire le message (*déchiffrement/décryptage*). Le message d'origine est appelé *plaintext* et sous une forme cryptée est appelé *ciphertext*. La *Cryptanalyse* confronte la cryptographie et vise à briser des moyens cryptographiques et lire les informations sensibles. Plus précisément, elle consiste à analyser les textes chiffrés afin de trouver les textes en clair sans connaissance a priori du processus de déchiffrement. Ces deux branches : cryptographie et cryptanalyse constituent la *cryptologie*: la science du secret.

Il existe deux types d'algorithmes cryptographiques. Les plus anciens sont les algorithmes cryptographiques dits symétriques : l'émetteur et le récepteur partage la même clef. On parlera de chiffrement symétrique ou à clef privée. La cryptographie moderne a vue naître au XX^e siècle le chiffrement à clef privée ou asymétrique : émetteur et récepteur disposent chacun d'une clef privée et utilise une clef publique. La clef publique sera utilisée pour chiffrer un message, alors que la clef privée qui doit rester confidentielle sera utilisée pour le déchiffrer.

Dans la suite nous utilisons principalement les algorithmes symétriques de chiffrements par blocs : l'AES Advanced Encryption Standard et le DES Data Encryption Standard. Ces algorithmes cryptographiques réputés sûrs d'un point de vue mathématiques deviennent vulnérables du fait de leur implantation sur des composants électroniques comme par exemple les FPGA, les cartes à Puce: ces objets qui envahissent peu à peu notre vie quotidienne.

En effet, ces dispositifs cryptographiques, lors de l'exécution d'un calcul, fuient de l'information sur la variable sensible. Cette fuite peut être le temps d'exécution, la consommation ou le rayonnement électromagnétique produit au cours du calcul. Les attaques qui ciblent la mise en oeuvre physique et non pas la faiblesse mathématique de l'algorithme afin d'en extraire la clef secrète sont appelés attaques physiques.

Ces attaques physiques sont essentiellement de deux types : les *attaques par injection de fautes* (FA) et les *attaques par canaux cachés* (SCA). L"attaque FA tente d'injecter des modifications physiques (des fautes) dans l'environnement de la carte (lumineuses, impulsions électriques, magnétiques, etc...) pour introduire des modifications dans le contenu des mémoires de la carte afin de provoquer des résultats erronées exploitables. Contrairement à l'attaque FA, une attaque par canaux cachés est une attaque passive qui vise à analyser et exploiter les fuites physiques du dispositif cryptographique pendant le chiffrement.

Chapitre 2 : Introduction Générale sur les SCA

Les attaques par canaux cachées peut être divisée en fonction de la nature de l'information physique exploitée:

- le temps de calcul,
- la consommation de courant,
- le rayonnement électromagnétique,

Ces informations nous renseignent sur le fonctionnement du composant, et donne des indices à un attaquant pour retrouver les éléments secrets.

Dans la littérature, la mise en oeuvre d'une attaque par canaux cachés se décompose de deux étapes comme le montre la figure. 1. Une *étape expérimentale*

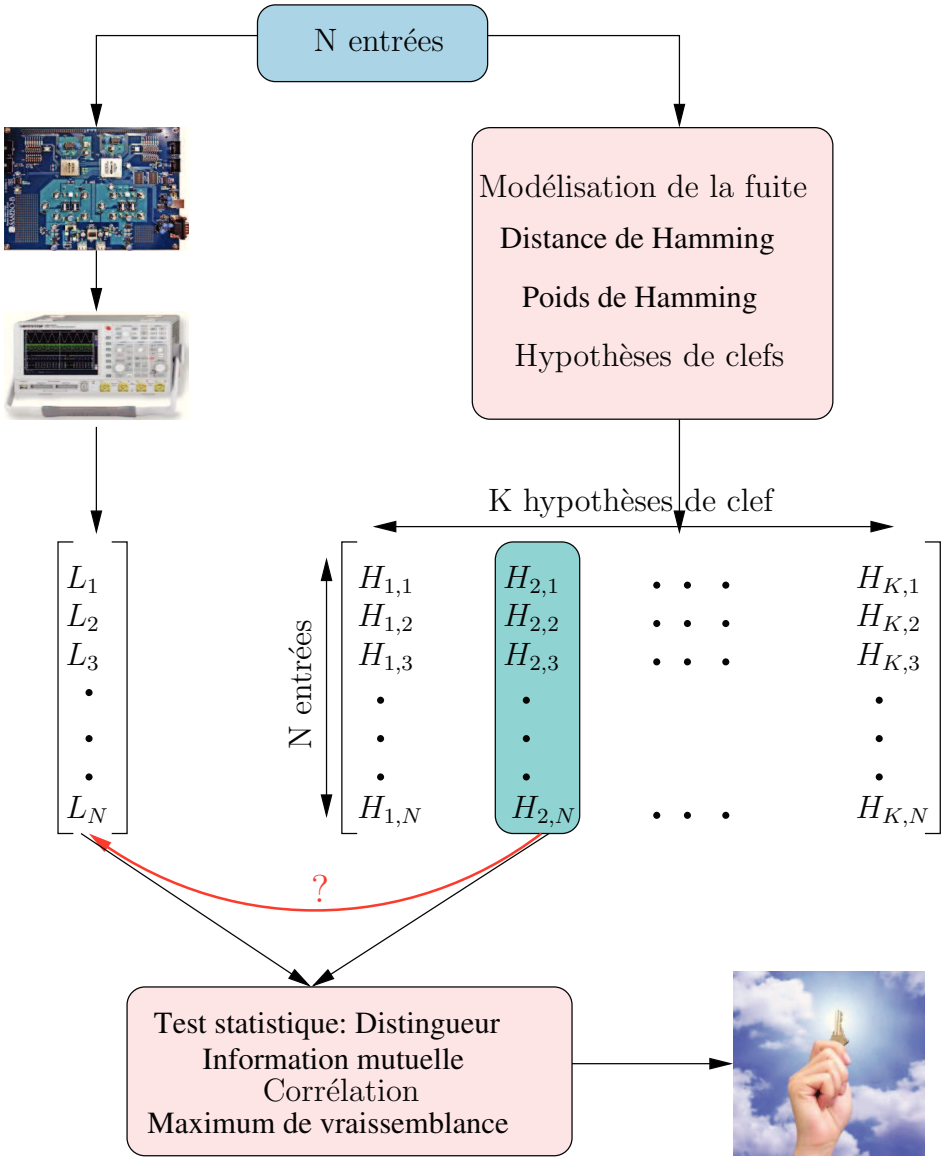


Figure 1: Illustration d'une attaque par canaux cachés.

qui consiste à acquérir les traces de consommations (leakage/fuite) lors d'un chiffrement; et une *étape de prédiction* qui consiste à étudier l'algorithme cryptographique et construire des modèles de la fuite. Ce modèle théorique sert à prédire l'activité d'une partie du circuit en faisant des hypothèses sur la clef. Le but est de trouver la bonne hypothèse qui correspond à la clef secrète. Pour comparer le degré de dépendance entre la fuite et le modèle, l'attaquant applique un test statistique appelé

distingueur. Ce test peut être par exemple un calcul de différence de moyenne, de l'information mutuelle ou de maximum de vraisemblance. L'hypothèse de clef qui maximise le score de ce distingueur correspond à la bonne hypothèse de clef.

Pour faire face à ces attaques, plusieurs contre-mesures ont été développées. Nous citons essentiellement deux catégories:

- La dissimulation "Hiding" : dont l'objectif est d'équilibrer la consommation et de la rendre constante à chaque instant quelque soient les données manipulées par l'algorithme.

- Le masquage : dont l'objectif est de "randomiser" la consommation. En effet la variable sensible portant le secret Z est divisée en deux parties: une variable aléatoire inconnue pour l'attaquant appelé le masque M et la donnée masquée qui combine le masque et la variable sensible $Z \perp M$.

Une implémentation masquée est robuste contre les attaques SCA de premier ordre. En revanche, elle est la cible d'attaques du second ordre, voire d'ordre plus élevé, permettant d'exhiber le secret en attaquant plusieurs variables simultanément. En effet, un attaquant peut toujours combiner les activités liées à la manipulation des deux variables (masque, donnée masquée) afin de trouver la clef de chiffrement. Comme le montre la figure. 2, on distingue essentiellement deux types d'attaques d'ordre élevé : les attaques multivariées (si la manipulation du masque et de la donnée masquée se produit à deux instants différents) et les attaques univariées (si la manipulation du masque et de la donnée masquée est simultanée).

Deuxième Partie : Attaques SCA d'ordre élevé

La deuxième partie de cette thèse concerne l'étude des attaques SCA d'ordre supérieur. Nous avons proposé trois nouveaux distingueurs: l'analyse de puissance basé sur la variance (VPA), l'analyse de puissance basée sur l'entropie (EPA) et l'analyse de l'information inter-classe (IIA). Nous étudions leur efficacité en présence de contre-mesure basée sur la technique de masquage.

Chapitre 1 : "Variance-based Power Attack"

La figure. 3 illustre l'architecture d'une implémentation masquée. En supposant que le modèle de la fuite est en distance de Hamming, la partie déterministe de la fuite L d'un masquage de premier ordre peut être exprimée comme : $L = \mathsf{HW}(Z \oplus M'') + \mathsf{HW}(M)$, où $Z = X \oplus S(X \oplus k)$ est la variable sensible, $M'' = M \oplus M'$ est la mise à jour du masque et X, S et k sont respectivement le plaintext, la fonction SubByte et la clef de chiffrement.

Considérant des variables codées sur 4 bits, il existe cinq distributions possibles de la fuite en fonction du poids de Hamming de la valeurs sensibles $\mathsf{HW}(Z)$, comme le montre la figure. 4. Une conséquences importante peut être soulignée sur le résultat

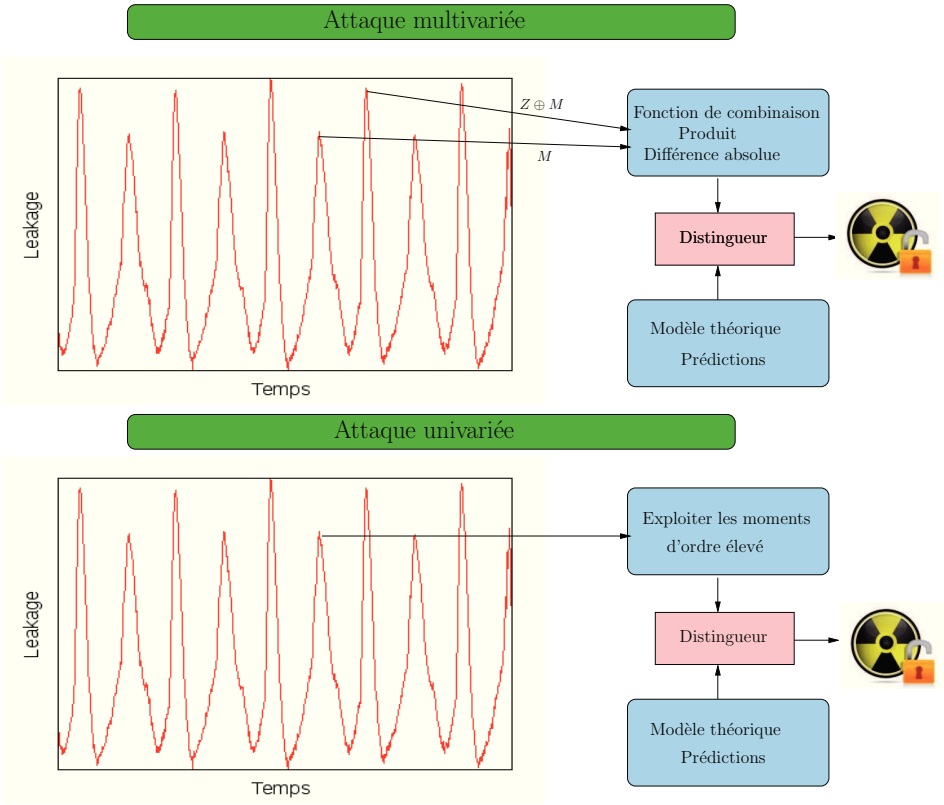


Figure 2: Illustration des attaques d'ordre élevé.

de la figure. 4 : les distributions $\mathsf{P}[L \mid \mathsf{HW}(Z)]$ pour la bonne clef ont la même valeur de moyenne et ne diffèrent que par leurs variances.

Cela nous amène à définir en conséquence l'attaque Variance-based Power Analysis. La stratégie de cette attaque est la suivante:

1. Appliquer N messages (X_i avec i dans $[1, N]$) et collecter N observations de la consommation d'énergie (traces L_i).

2. Pour chaque S-Box, faire des hypothèses sur la clef k dans $[0, 63]$:

   - Trier les traces L_i pour obtenir les cinq partitions de l'activité correspondant aux cinq valeurs possibles $\mathsf{HW}(Z)$.
   - Calculer la variance v_l pour chaque partition.
   - Calculer l'indicateur $\mathrm{VPA}(k)$ comme étant une combinaison linéaire des variances pondérées par les poids w_l: $\mathrm{VPA}(k) = \sum_{l=0}^{4} w_l \bullet v_l$.

3. La bonne hypothèse de clef k^* correspond à $\arg\max_k \mathrm{VPA}(k)$.

L'attaque VPA a été testée sur 200.000 traces de consommation acquises dur une implémentation FPGA d'un DES masquée. Les 8 sous-clefs utilisées lors du premier tour du DES masqué ont été trouvées par cette attaque ce qui prouve sa robustesse.

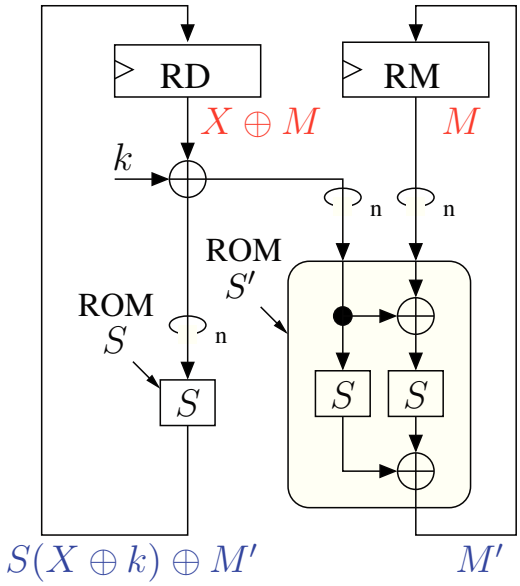


Figure 3: Architecture d'une implémentation masquée.

Chapitre 2 : "Entropy-based Power Attack"

On a également proposé une nouvelle approche d'attaque basée sur les principes de la théorie de l'information (à savoir le calcul d'entropie), appelé Entropy-based Power analysis (EPA). Cette nouvelle attaque permet d'exploiter davantage la fuite d'information dans un circuit et offre une meilleure distinction entre les hypothèses de clefs. Une évaluation empirique approfondie de l'attaque proposée confirme l'immense avantage de cette nouvelle approche par rapport aux autres attaques telles que la "Mutual Information Analysis Attack" (MIA). La stratégie de l'EPA est similaire à la VPA, sauf qu'on calculera une entropie conditionnelle au lien de la variance :

1. Appliquer N messages (X_i avec i dans $[1, N]$) et collecter N observations de la consommation d'énergie (traces L_i).

2. Pour chaque S-Box, faire des hypothèses sur la clef k dans $[0, 63]$:

   - Trier les traces L_i pour obtenir les cinq partitions de l'activité correspondant aux cinq valeurs possibles $\mathsf{HW}(Z)$.
   - Calculer l'entropie conditionnelle $\mathsf{H}[L \mid \mathsf{HW}(Z) = l]$ pour chaque partition.
   - Calculer l'indicateur $\mathrm{EPA}(k)$ comme étant une combinaison linéaire des entropies pondérées par les poids w_l: $\mathrm{EPA}(k) = \sum_{l=0}^{4} w_l \times \mathsf{H}[L \mid \mathsf{HW}(L) = l]$.

3. La bonne hypothèse de clef k^* correspond à $\arg\max_k \mathrm{EPA}(k)$.

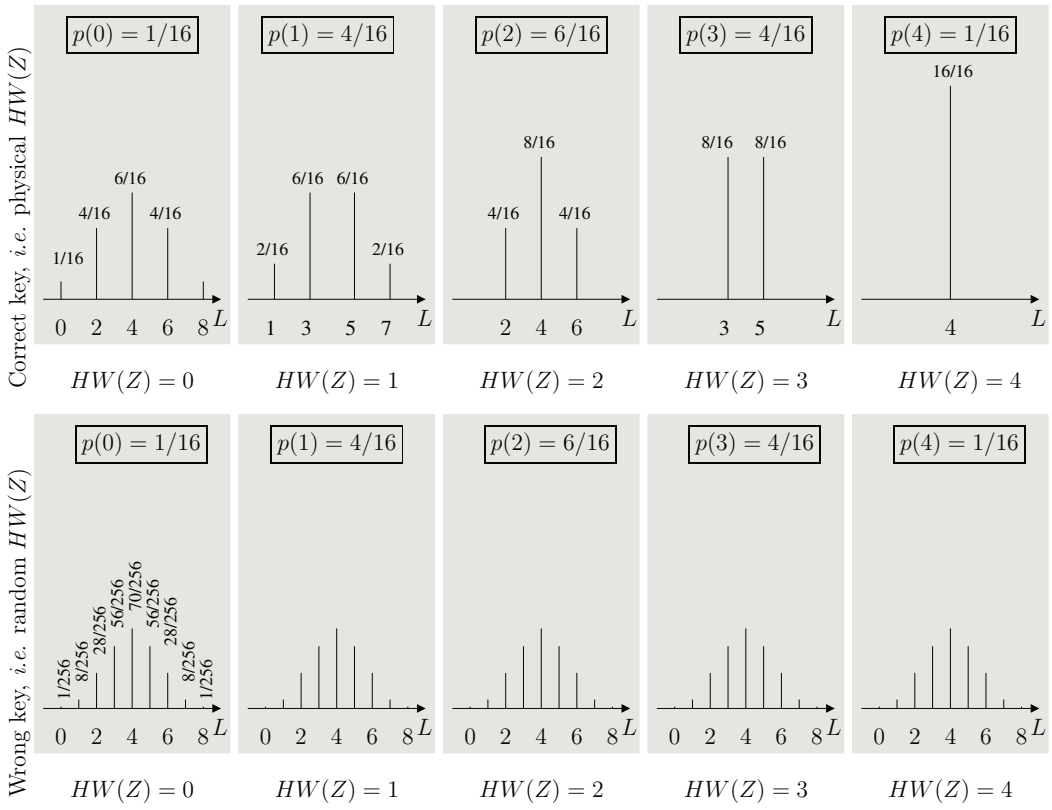


Figure 4: PDFs correspondants aux cinq valeurs possibles du $\mathsf{HW}(Z)$.

L'attaque EPA a été testée sur une implémentation d'un DES masquée et toutes les 8 sous-clefs utilisées lors du premier tour du DES masquée ont été trouvées.

Chapitre 3 : "Inter-class Information Analysis"

Nous avons proposé également un nouveau distingueur appelé "Inter-class Information Analysis" (IIA). A l'inverse de MIA ou KSA, ce distingueur consiste à comparer les fuites conditionnelles entre elles, deux à deux comme le montre la figure. 5.

Nous avons prouvé théoriquement que le distingueur IIA partage les mêmes propriétés mathématiques qu'un calcul de l'information mutuelle. A savoir, le distingueur IIA vérifie:

- **Symétrie**: $\mathsf{II}[L;Z] = \mathsf{II}[Z;L]$.
- **Dépendance**: $\mathsf{II}[L;Z] = 0 \iff L$, Z sont indépendants.
- **Relation avec l'information mutuelle**: $2\mathsf{II}[L;Z] \geq \mathsf{I}[L;Z]$.
- **Soundness**: $\mathsf{II}[L;Z] \leq \mathsf{II}[L;Z^*]$.

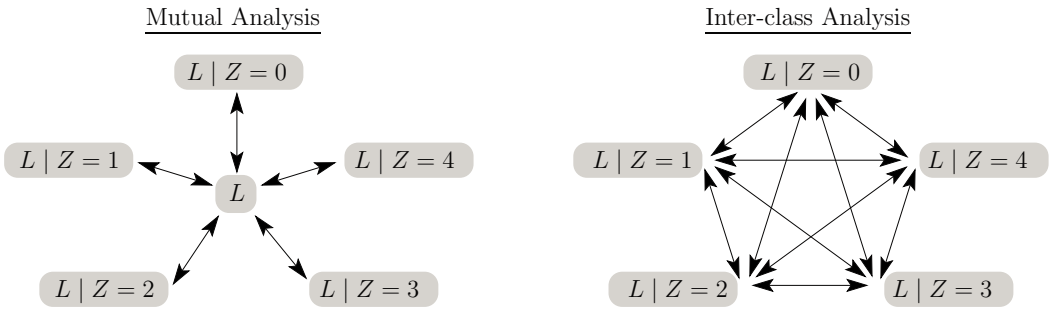


Figure 5: Le distingueur "Inter-class Information Analysis"

Pour comparer le distingueur IIA à l'attaque MIA, nous nous sommes basés sur le calcul de la métrique du taux de succès. Les résultats de simulations ont montré que l'attaque IIA est plus discriminante que l'attaque MIA en utilisant les mêmes techniques d'estimation (la même erreur d'estimation). Dans un deuxième temps, nous avons appliqué cette notion d'inter-classe au distingueur KSA. Nous avons prouvé que l'attaque IKSA résultante permet d'obtenir des taux de succès beaucoup plus élevés qu'une simple attaque KSA.

Enfin pour comparer tout les distingueurs que nous avons proposés, nous avons suggéré une méthode de réduction de l'erreur d'estimation sur le taux de succès. Le principe consiste à augmenter le nombre d'essais utilisés pour le calcul du taux de succès. Les résultats de simulation montrent le gain significatif sur le nombre de mesures nécessaires pour réussir l'attaque si on applique la notion d'inter-classe. Par conséquent, à travers cette nouvelle notion, nous définissons une nouvelle classe de distingueur non-équivalente à celle de MIA/KSA.

Troisième Partie : Caractérisation des procédés de masquage

La troisième partie présente les schémas de masquage de premier ordre les plus utilisés dans la littérature et discute leurs vulnérabilités aux attaques SCA de second ordre. Ensuite, nous proposons une nouvelle mesure appelée "HO-CPA Immunity" (HCI). Cette métrique intervient pour évaluer à la fois l'efficacité des attaques HO-CPA et les fuites d'ordre supérieur d'une implémentation masquée.

Chapitre 1 : Nouvelle Métrique d'évaluation

Pour évaluer la robustesse des implémentations masquées contre les attaques SCA d'ordre élevé, François-Xavier Standaert *et al.* ont montré qu'il est nécessaire d'enquêter soigneusement sur les fuites d'information et les adversaires qui exploitent ces fuites, séparément. Ainsi, les évaluations des implémentations protégées par masquage devrait tenir en deux étapes. Tout d'abord, une analyse de théorie de l'information détermine la réelle quantité d'information fuite par le circuit. Ensuite, une analyse de la sécurité détermine l'efficacité de divers distingueurs pour exploiter

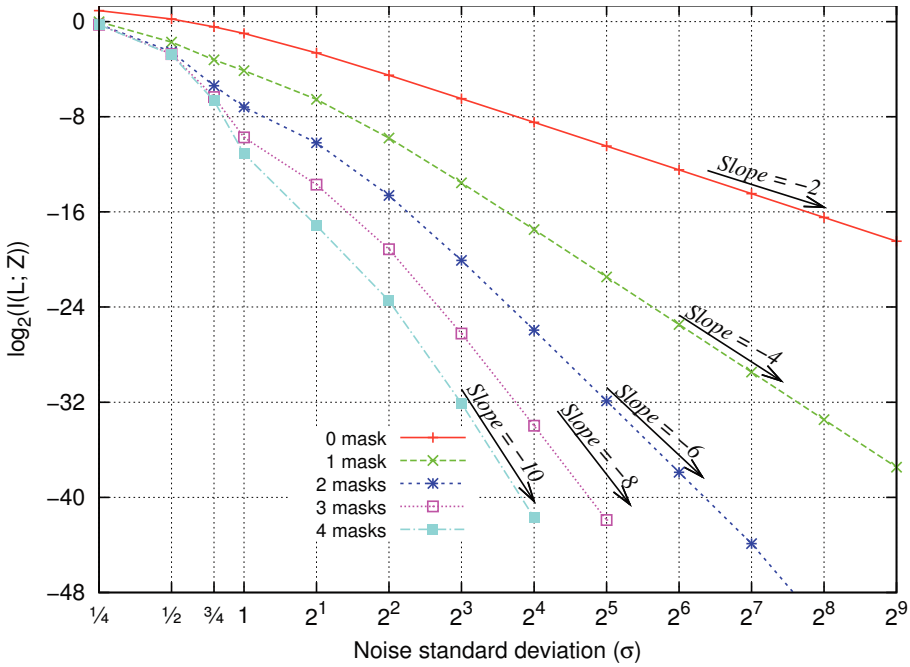


Figure 6: Le HCI une nouvelle métrique d'évaluation.

cette fuite. En appliquant cette méthodologie en utilisant des traces simulées ou des mesures réelles, nous obtenons une évaluation juste et complète du niveau de le sécurité qu'un système protégé par la technique de masquage peut assurer.

Dans le cas d'une implémentation matérielle (sur FPGA) d'un algorithme de chiffrement masquée, nous avons proposé une nouvelle métrique appelée "HO-CPA immunité" qui permet d'évaluer à la fois la résistance contre les attaques d'ordre supérieur et la quantité de l'information fuite. En tant que métrique de sécurité, la "HO-CPA immunité" peut être définit comme étant le nombre de moments statiques de la distribution $L|\mathsf{HW}(Z)$ qui sont égaux plus un. Ensuite, nous avons prouvé mathématiquement que la quantité de l'information mutuelle fuite par le circuit décroît en $\sigma^{-2\times\mathsf{HCI}}$ quand l'écart type du bruit σ tend vers l'infini. Les résultats de simulation reportés sur la figure. 6 confirment les résultats théoriques que nous avons démontrés pour les différents ordres de masquage étudié.

Chapitre 2 : Évaluation des Contre-mesures par Masquage

Dans ce chapitre, nous avons appliqué la métrique HCI dans le cas d'un masquage de premier ordre. Nous avons évalué la sécurité de cette contre-mesure dans le cas d'une implémentation matérielle et logicielle. En effet, nous avons montré que:

- pour les implémentations matérielles : l'information mutuelle fuite est équivalente à σ^{-4} ($\mathsf{HCI} = 2$)

- pour les implémentations logicielles : l'information mutuelle fuite est équivalente à σ^{-2} ($\mathsf{HCI} = 1$),

Dans le cas des attaques CPA multivariées combinées par le produit, nous avons exhibé une relation entre l'information mutuelle et cette attaque. Asymptotiquement, nous avons obtenu que $\frac{\mathrm{MIM}}{\mathrm{CPA}} = constant$. Ce résultat nous a permis d'établir le lien entre le nombre de messages qui permettent d'avoir un taux de succès de 90% et l'écart type de bruit σ : $N_{90\%} \propto \sigma^4$.

Quatrième Partie : Nouvelles Contre-mesures masquées

Dans cette quatrième partie, nous proposons trois nouvelles contre-mesures par masquage de premier ordre pour contrer non seulement les attaques du premier ordre, mais aussi les attaques d'ordre supérieur. Le première contre-mesure est appelée "Leak-free". Nous montrons que, avec un masquage de premier ordre, et en supposant une fonction de fuite à distance, il est possible d'annuler la fuite d'information sensible. Nous proposons également une deuxième façon d'appliquer le masquage où la mise à jour du masque est basé sur un code de Gray. Son principal avantage par rapport à la contre-mesure "Leak-free" est la réduction des besoins en ressources matérielles (mémoire) pour l'implémenter. Nous présentons une nouvelle contre-mesure appelée "Leakage Squeezing". Elle consiste à manipuler le masque à travers une bijection de F, visant à réduire la dépendance entre le masque et la donnée masquée. Enfin nous étendons cette contre-mesure dans le contexte du second ordre.

Chapitre 1 : La Contre-mesure "Leak-Free"

Nous avons démontré dans les parties précédentes que la distribution de la fuite $L = \mathsf{HW}(Z \oplus M'') + \mathsf{HW}(M'')$ (et en particulier sa variance) dépend de la variable sensible Z. Dans ce chapitre, nous proposons une méthode pour annuler cette dépendance en remplaçant l'opération de masquage des données $X \oplus M$ par une nouvelle notée $X@M$.

Une solution simple, profondément analysée dans ce travail, est de choisir une fonction @ tel que $X \,@\, M = X \oplus F(M)$ pour une certaine fonction F bien choisie. Dans ce qui suit, on désigne par p la dimension de M et on suppose que F est une (p, n)-fonction, *i.e.* $F : \mathbb{F}_2^p \mapsto \mathbb{F}_2^n$. Dans ce cas, la fonction de la fuite s'écrit:

$$A(X@M, X'@M') + A(M, M') = A(Z \oplus F(M) \oplus F(M')) + A(M'')\,,$$

où A est une fonction de l'activité. En vue de cette équation, deux conditions suffisantes sont à satisfaire pour garantir que L est indépendant de Z:

1. $M \oplus M'$ est constant et
2. $F(M) \oplus F(M')$ est uniforme.

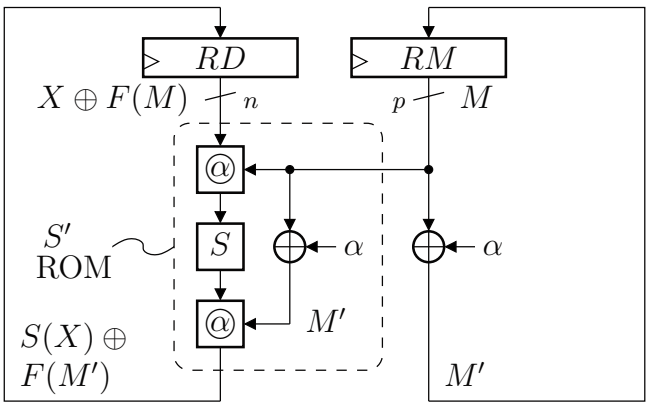


Figure 7: L'architecture de la contre-mesure "Leak-free".

Pour satisfaire la première condition, nous avons fixé $M' = M \oplus \alpha$ avec α une constante non nulle. Ensuite, nous avons proposé deux constructions de F pour satisfaire que $F(M) \oplus F(M \oplus \alpha)$ soit uniforme pour cette constante α. Cependant, ces deux constructions imposent que la dimension du masque p soit strictement supérieure à celle de la donnée n. Nous montrons dans la figure. 7 l'architecture de notre contre-mesure.

En ce qui concerne l'évaluation de la sécurité de cette contre-mesure, nous avons montré que la fuite d'information mutuelle est nulle et par conséquent notre solution permet de se protéger contre toutes les attaques univariées de tout les ordres. Lorsque la fonction de l'activité s'écarte légèrement du modèle de distance de Hamming, nous avons montré par simulation que notre solution fuit de l'information mais reste plus performante qu'un masquage classique de premier ordre.

Chapitre 2 : La Contre-mesure utilisant un Code de Gray

En dépit de ses avantages (protection contre toutes les attaques univariées dans le modèle de distance de Hamming, l'efficacité, la simplicité, *etc*), la contre-mesure "Leak-free" présente deux inconvénients. Tout d'abord, une seule paire de masques (M, $M' = M \oplus \alpha$) est utilisée durant tout le calcul. Cette propriété n'a aucun impact sur la sécurité de la contre-mesure contre les attaques univariées, mais peut la rendre vulnérable aux attaques multivariées. Un deuxième problème avec cette contre-mesure, c'est que la dimension du masque p doit être strictement supérieur à celle de la donnée n pour construire une fonction F satisfaisant les deux conditions suffisantes précédentes ce qui induit un sur-coût temps/mémoire par rapport à un masquage classique du premier ordre.

Dans ce chapitre, nous présentons une alternative à cette contre-mesure. Plus précisément, nous avons montré qu'en utilisant un compteur de Gray pour mettre à jour les valeurs du masque il est possible de définir un nouveau schéma de masquage qui partage toutes les propriétés de la solution "Leak-free" avec un atout supplé-

mentaire est que le masque et la donnée ont la même taille ($p = n$). Ensuite, nous avons prouvé qu'il existe une construction de la fonction F qui satisfait la deuxième condition. La figure. 8 illustre une l'architecture de l'algorithme AES tirant profit de cette nouvelle contre-mesure. Ce nouveau schéma de masquage a été évalué avec des vraies mesures FPGA. Les résultats de cette étape d'évaluation ont confirmé nos résultats théoriques, à savoir la protection contre les attaques SCA d'ordre élevé.

Chapitre 3 : La Contre-mesure "Leakage Squeezing"

Une troisième contre-mesure que nous avons proposée est le "Leakage squeezing". L'objectif est de réduire la dépendance entre la distribution de la fuite et la variable sensible et de résister aux attaques d'ordre élevé bornées par un certain degré d. Une approche simple consiste à modifier le masque M par une transformation bijective F avant de manipuler la valeur $F(M)$ dans le registre de masque. La figure. 9 illustre ce nouveau schéma de masquage. La fuite de cette contre-mesure est $L = \mathsf{HW}(Z \oplus M \oplus M')) + \mathsf{HW}(F(M) \oplus F(M'))$

Notre but est de trouver des bijections F telles que la fonction optimale qui maximise une attaque CPA d'ordre d :

$$f_{\text{opt}}(z) = \mathbb{E}[\big(\mathsf{HW}(Z \oplus M'') + \mathsf{HW}(F(M) \oplus F(M'))\big)^d \mid Z = z]$$

soit indépendante de la variable sensible Z. En développant $f_{\text{opt}}(z)$, on fait apparaître des termes : $\mathsf{Term}[p,q](f_{\text{opt}})(z) \doteq \mathbb{E}[\mathsf{HW}^p[z \oplus M''] \times \mathsf{HW}^q[F(M) \oplus F(M')]]$, avec p et q deux entiers positifs tels que $p + q \leq d$. Par conséquent, pour résister aux attaques HO-CPA, les termes $\mathsf{Term}[p,q](f_{\text{opt}})(z)$ doivent être indépendants de z pour tout p et q. Nous avons fixé donc trois conditions équivalentes, que la bijection F doit satisfaire :

- **Condition en terme de transformation de "Walsh-Hadamard"** : Pour tous entiers positifs P et Q, $f_{\text{opt}}(z)$ est constante pour $p \in [\![0, P]\!]$ et $q \in [\![0, Q]\!]$ si et seulement si: $\forall a, b \in \mathbb{F}_2^n, 0 \leq \mathsf{HW}(a) \leq P, 0 \leq \mathsf{HW}(b) \leq Q$ la transformée de Walsh-Hadamard $\widehat{(b \cdot F)}_\chi(a) = 0$.

- **Condition en terme de fonction "Correlation-Immunity"" ** : $f_{\text{opt}}(z)$ est constante si et seulement si l'indicateur du graphe C de la fonction F est "correlation-immune" d'ordre d.

- **Condition en terme de code** : $f_{\text{opt}}(z)$ est constante si et seulement si le code C égal au graphe de la fonction F a la plus grande distance duale.

Nous avons démontré que tout code linéaire de rendement 1/2 qui s'écrit sous la forme $[2n, n, \delta]$, avec $\delta > d$ sa distance minimale, permet de résister aux attaques HO-CPA d'ordre $d = \delta - 1$. Avec des F linéaires, il est possible de protéger:

- DES contre toute attaque HO-CPA d'ordre $d \leq 3$, et

- AES contre toute attaque HO-CPA d'ordre $d \leq 4$.

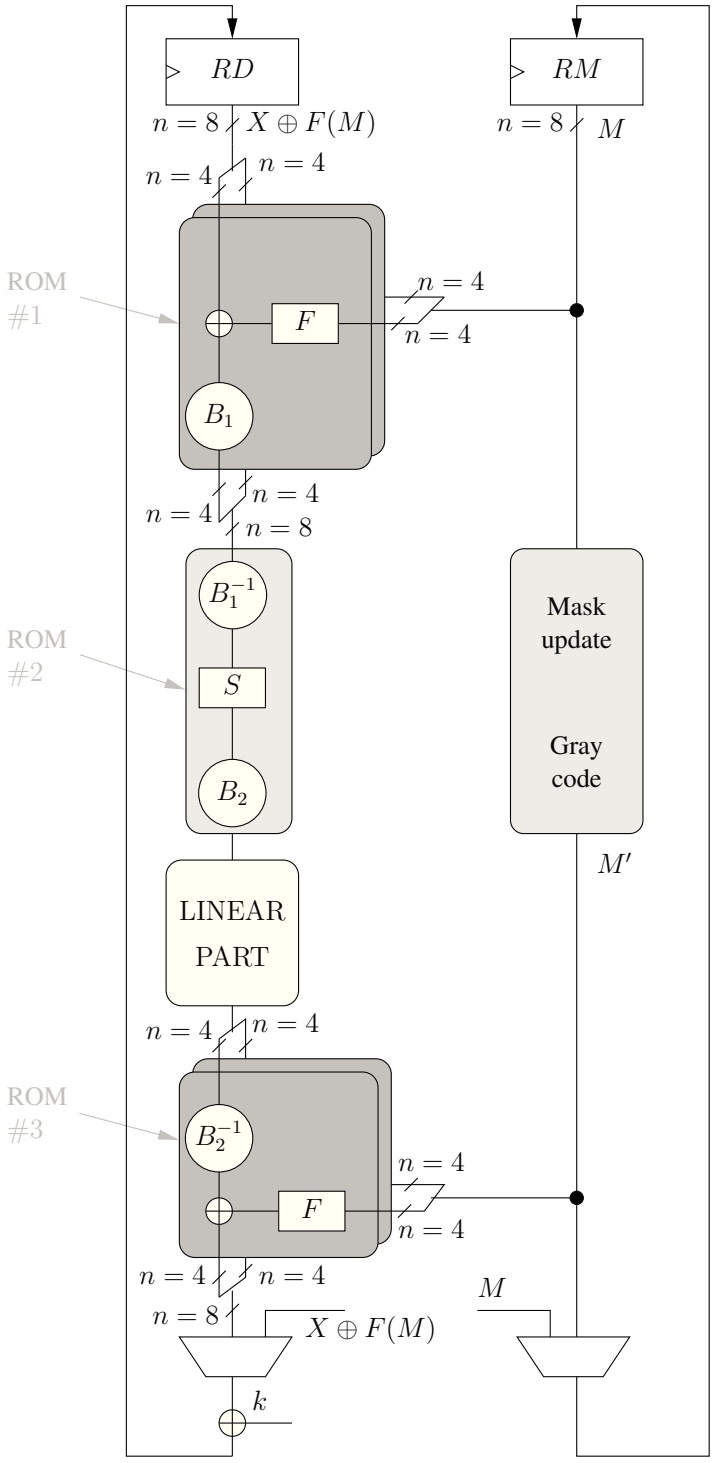


Figure 8: L'architecture de l'AES protégé par le masquage avec le code de Gray.

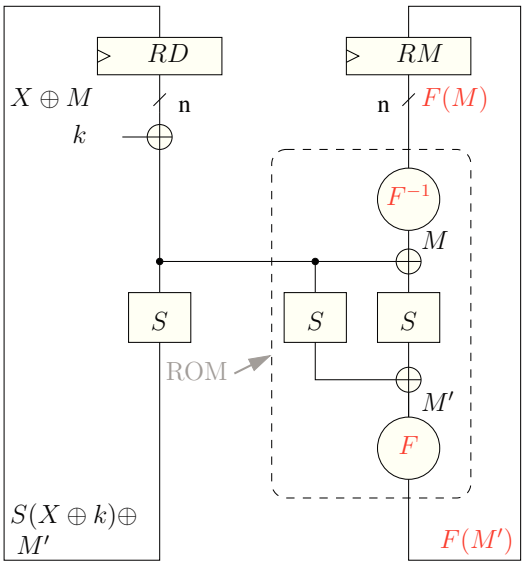


Figure 9: L'architecture de la contre-mesure "Leakage Squeezing".

Nous avons aussi étudié les bijections non-linéaires et nous avons prouvé qu'elles permettent de mieux résister. Par exemple, le code Nordstrom-Robinson $(16, 256, 6)$ a une distance duale plus grande que les codes linéaires et par la suite permet de protéger l'AES contre toute HO-CPA d'ordre $d \leq 5$.

Les phases d'évaluation de la technique "Leakage Squeezing" ont montré, d'une part, la réduction de la quantité d'information mutuelle fuite par le circuit. D'autre part, nous avons montré que cette contre-mesure, contrairement au masquage "Leak-free", est peu sensible aux déviations du modèle.

Nous avons proposé deux implémentations de DES maqués basées sur le principe du "Leakage Squeezing". Ces deux architectures ont été synthétisées sur une carte FPGA STARTIX II et comparées en termes de complexité et de débit aux architectures d'un masquage classique. Les résultats ont montré que la technique "Leakage Squeezing" n'introduit pas de sur-coût par rapport aux solutions existantes. L'évaluation avec des vraies mesures acquises sur la carte FPGA confirme nos résultats théoriques obtenus.

Chapitre 4 : La Contre-mesure "Leakage Squeezing" d'ordre deux

La dernière étape de ce travail a consisté à étendre la technique "Leakage squeezing". La figure. 10 illustre l'architecture de cette solution.

Pour résister aux attaques HO-CPA d'ordre élevé, nous avons considéré des bijections linéaires F_1 et F_2. Ensuite, nous avons prouvé que les codes linaires de rendement $1/3$ avec trois ensembles d'information disjoints permettent d'atteindre des niveaux de protection assez élevés.

En effet, il est possible de protéger:

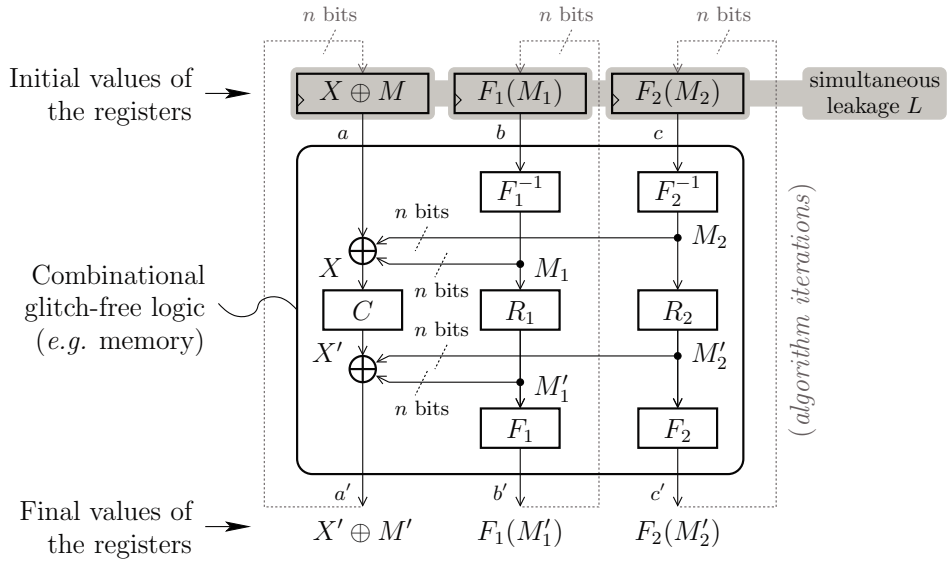


Figure 10: L'architecture de la contre-mesure "Leakage Squeezing" d'ordre deux.

- DES avec le code $[12, 4, \mathbf{6}]$ contre toute attaque HO-CPA d'ordre $d \leq 5$.
- AES avec le code $[24, 8, \mathbf{8}]$ contre toute attaque HO-CPA d'ordre $d \leq 7$.

Cinquième Partie : Conclusions et Perspectives

Chapitre 1 : Conclusions

Nous listons les axes de recherches qui ont été développés durant ces trois années de thèse :

- Étude de l'état de l'art sur les contre-mesures basées sur le masquage Booléen.
- Élaboration de trois attaques de second ordre testées sur des circuits protégés.
- Proposition d'une métrique d'évaluation de la sécurité.
- Mise en place de trois contre-mesures contre les attaques d'ordre élevé.
- Caractérisation des implémentations masquées d'ordre élevé.

Chapitre 2 : Perspectives

En termes de perspectives, il est important de souligner qu'il reste beaucoup de pistes d'amélioration. Quelques grands défis à relever sont listés ci-après:

- Trouver les poids optimaux pour l'attaque EPA afin d'améliorer sa robustesse.

- Comparer les attaques VPA et EPA avec des attaques SCA multivariées (par exemple l'attaque MMIA), en utilisant plusieurs capteurs (par exemple deux sondes magnétiques) placés à différents (X, Y, Z, ϑ) emplacements sur un cryptoprocesseur masqué (cartographie).
- Analyser la robustesse du distingueur IIA sur des mesures réelles.
- Comparer nos contre-mesures basées fondamentalement sur le masquage booléen avec d'autres solutions telles que le masquage multiplicatif ou affine qui offrent également un bon compromis performance/sécurité.
- Trouver des nouvelles primitives de mise à jour du masque pour la contre-mesure "Leak-free' pour obtenir une protection également contre les attaques SCA multivariées.
- Étendre la contre-mesure "Leakage squeezing" à l'ordre $d > 2$ en étudiant les codes de rendements $1/d$.
- Étudier l'effet de "glitches" sur nos contre-mesures.

Contents

List of Figures

List of Tables

Acronyms

AES	Advanced Encryption Standard
ASIC	Application Specific Integrated Circuit
CDF	Cumulative Distribution Function
CHES	Cryptographic Hardware and Embedded Systems
CMOS	Complementary Metal Oxide Semi-conductor
CPA	Correlation Power Analysis
DES	Data Encryption Standard
DoM	Difference of Means
DPA	Differential Power Analysis
DSP	Digital Signal Processor
EMA	Electro-Magnetic Analysis
EPA	Entropy-based Power Analysis
FA	Fault Analysis
FPGA	Field Gate Programmable Array
HD	Hamming Distance
HW	Hamming Weight
IC	Integrated Circuit
IIA	Inter-class Information Analysis
IKSA	Inter-class Kolmogorov-Smirnov Analysis
ISO	International Organization for Standardization
KSA	Kolmogorov-Smirnov Analysis
LFSR	Linear Feedback Shift Register
MIA	Mutual Information Analysis
MTD	Minimum Traces to Disclosure
NIST	National Institute of Standards and Technology
NSA	National Security Agency
PDF	Probability Distribution Function
RAM	Read-Access Memory
ROM	Read-Only Memory
RSA	Rivest Shamir and Adleman
S-box	Substitution box
SCA	Side Channel Analysis
SNR	Signal-to-Noise Ratio
SPA	Simple Power Analysis
SPN	Substitution Permutation Network
TA	Template Attacks
VPA	Variance-based Power Analysis
XOR	eXclusive OR

Thesis Context

Motivations

Side channel attacks are a serious threat against modern cryptographic implementations. They exploit information that leaks from physical implementations of cryptographic algorithms. This leakage (*e.g.* the power consumption or the electromagnetic emanations) may indeed reveal information on the secret data manipulated by the implementation. During the two last decades, the development of the smart card industry has urged the cryptographic research community to carry on with this new concept of attacks and many papers describing either countermeasures (CMs) or attacks improvement have been published (see [CJRR99, BCO04, OMHT06, NRS08] for instance). In particular, the original attacks in [KJJ96, KJJ99] have been improved and the concept of higher-order side channel analysis (HO-SCA) has been introduced in [Mes00b].

Masking is one of the most efficient countermeasures to thwart SCA attacks. The idea of masking is to conceal the sensitive variable through arithmetic or Boolean operations with random values in order to avoid the correlation between the cryptographic device's power consumption and the data being processed [CJRR99, GP99, AG01].

The masking can be characterized by the number of random masks used per sensitive variable. So, it is possible to give a formal definition for a higher-order masking scheme: a d^{th}-order masking scheme involves $d+1$ shares. The security is reached at order d provided that any combination of d shares during the entire computation conveys no information about the sensitive variable.

We must concur that computing with $d+1$ shares without revealing information from any set of size d of intermediate values can be challenging. In fact, masked implementations can always be attacked, since all shares [GBPV10] or a judicious combination [PRB09] of them unambiguously leaks information about the sensitive variable. The construction of an efficient masking scheme thus became of great interest.

In this thesis, focus is laid upon the study of masking schemes and the enhancement of their security against higher-order SCA attacks. Our objectives are:

- To develop novel higher-order SCA distinguishers.
- To efficiently evaluate the security of masking countermeasures.
- To devise new masking functions which resist higher-order SCA attacks.

Organization

This thesis is organised as follows:

The first part gives a general background about modern cryptography and physical attacks. Chapter 1 discusses the state-of-the-art of the most common cryptographic algorithms. These algorithms are designed to be used in real-life applications and implemented in electronic device. This physical implementation leaks information about the secret information that can be exploited by side channel attacks. Chapter 2 gives a general introduction to side channel attacks and presents some classical models and metrics used for their analysis. A general survey of the main existing attacks and countermeasures is described. Moreover, the higher-order SCA context is introduced.

The second part of this thesis deals with the study of higher-order SCA attacks. We proposed three novel distinguishers: the Variance-based Power Analysis (VPA), the Entropy-based Power Analysis (EPA) and the Inter-class Information Analysis (IIA). The VPA attack, introduced in Chapter 3, is based on a variance analysis of the observed power consumption. This distinguisher was tested against an FPGA implementation of a masked DES. The EPA attack is described in Chapter 4. It is an information-theoretic attack that uses a weighted sum of conditional entropies as a distinguisher. It is designed to ease the distinguishability between key hypotheses. Like VPA, the EPA is tested in a real context. Finally, the IIA is presented in Chapter 5. Unlike the most common information-theoretic distinguisher (*e.g.* the Mutual Information Analysis (MIA)), the IIA consists in comparing the conditional leakages between themselves, pairwise. We study its efficiency in the presence of masking countermeasures.

The third part is divided in two chapters. Chapter 6 presents the most widely used first-order masking schemes in the literature and discusses their vulnerabilities against second-order SCA attacks. Moreover, we propose a new SCA metric called HO-CPA immunity. It intervenes to assess both the efficiency of HO-CPA attacks and the amount of leakage of higher-order masking countermeasures. In Chapter 7, we deal with a formal security evaluation of Boolean hardware masking schemes. Mainly, we propose to extend the results presented in [SVCO+10] to the case of hardware implementations.

In the fourth part, we propose three novel masking countermeasures to counteract not only first-order but also higher-order SCA attacks. The first CM is called leakage-free masking and described in Chapter 8. We show that with a first-order masking, and assuming any distance leakage function (or even some small variations of it), it is possible to zero the sensible information leaked. In Chapter 9, we propose a second way to apply masking to secure hardware implementations of block ciphers. The mask update is based on a Gray code and its main advantage over the leakage-free CM is the reduction of the memory requirements. This countermeasure has been applied on an AES hardware implementation and evaluated in a real context. In Chapter 10, we present a novel masking countermeasure called leakage squeezing. It consists in manipulating the mask through a bijection F, aimed at reducing

the dependency between the shares' leakage. We mathematically demonstrate that optimal choices for F relate to optimal binary codes (in the sense of communication theory). This countermeasure protects against all HO-CPA attacks of order $d \leq 5$ and is shown to be resilient to imperfect leakage models. In Chapter 11, we extend the leakage squeezing countermeasure to the second-order context. We proved that with two masks, we provide resistance against all HO-CPA attacks of order $d \leq 7$.

Finally, Chapter 12 gives a general conclusion and opens some perspectives for future work.

Part I
Preliminaries

Chapter 1

General Background

In this chapter, we discuss the state-of-the-art of cryptographic algorithms. We will revisit the main branches of modern cryptography: the symmetric cryptography and the asymmetric cryptography. Even if these cryptographic algorithms are secure from a mathematical point of view, they become a target of physical attacks when they are implemented on electronic devices. Therefore, a theoretically secure cryptographic algorithm could leak information about the secret information due to its physical implementation.

Contents

1.1 Introduction to Cryptography

Cryptography is defined as the science of protecting sensitive information. It refers to the study of methods for sending messages with the help of a secret information *key* (*ciphering / encryption*) so that only the intended recipient can read the message (*deciphering / decryption*). The original message is called *plaintext* and under an encrypted form is called *ciphertext* or *cryptogram*. *Cryptanalysis* confronts cryptography and aims at breaking cryptographic means and reading the sensitive information. Namely, it consists in analyzing the ciphertexts in order to find the plaintexts without a priori knowledge of the deciphering process. Both cryptography and cryptanalysis make up *cryptology*: the *"science of secrecy"*.

Depending on the system to secure and the nature of the secret information, usually one or all of the following security aspects, that cryptography can provide, are required:

- **Confidentiality**: is to make information unintelligible to others as the allowed parties (or users).
- **Authenticity**: is to ensure that the corresponding to each partner is who he believed to be.
- **Integrity**: is to guarantee that the message has been correctly transmitted.
- **Non repudiation**: is to guarantee that the sender (respectively the receiver) cannot deny sending (respectively receiving) the message.

Theoretically cryptographic algorithms can be broken by using a *brute force* attack, *i.e.* an exhaustive key search. The attacker needs a single plaintext-ciphertext pair to retrieve the secret key. Then, he can test every possible key and check if the plaintext encryption is similar to the ciphertext. But, the cryptographic algorithms are based on *computational impossibility*: a cipher uses a large key space so that such exhaustive research is computationally impossible. For instance, if a key is coded in 128 bits, then 2^{128} key values have to be tested; such an exhaustive research require about one billion billion years when using the faster supercomputer on the market today.

Cryptographic algorithms are sorted into two main branches: the *symmetric cryptography* and the *asymmetric cryptography*. In next sections, we present the outlines of those cryptographic algorithms.

1.2 Symmetric Cryptography

Also called *secret key cryptography*, it relies on a unique secret key k shared between two communicating entities. The encryption of a message m is defined as $c = E_k(m)$, where E_k is an invertible function indexed by the shared key k. Then, to decipher the ciphertext c, the inverse function $m = E_k^{-1}(c)$ is computed. Symmetric cryptography is classified into two main types which are presented hereafter: *stream ciphers* and *block ciphers*.

1.2.1 Stream Ciphers

Stream ciphers are based on the *one-time pad cipher* (also called *Vernam cipher*), which encrypts every plaintext with a pseudo random cipher bit stream (*key-stream*) using a simple XOR[1] operation ($\oplus$). Basically, the key-stream is generated at random using one or several *Linear Feedback Shift Register* (LFSR) parametrized by a fixed length secret key k whose content is filtered by for instance a boolean function to produce the necessary non linearity. In the open literature, stream ciphers are split in two common types. There are synchronous stream ciphers where the key-stream is computed independently of the ciphertext, and asynchronous stream ciphers, called also *self-synchronizing*, where the key-stream is computed as a function of the ciphertext.

[1] XOR stands for exclusive OR

1.2.2 Block Ciphers

The block cipher algorithms aim at dividing an arbitrary length plaintext into several blocks of fixed length (*i.e.* 64 bits, 128 bits, ...) and then each is encrypted according to a particular *mode of operation* chosen depending on the context (the needed requirements and security features). Commonly, block ciphers are related to four modes of operation [Dwo01]:

- **Electronic Code Book (ECB) mode**: Each block is encrypted separately. From a security point of view, the ECB mode has a flaw since identical plaintexts yield identical ciphertexts.
- **Cipher Block Chaining (CBC) mode**: During encryption, the value of each input block is XORed with the previous ciphertext block. To guarantee the uniqueness of the message, the initial input block is XORed with an initialization vector. But, the use of a fixed initialization vector results a potential security flaw (two plaintexts with the same prefix yield to two identical ciphertexts).
- **Cipher Feedback (CFB) mode**: It operates in the same way as the self-synchronizing stream cipher.
- **Output Feedback (OFB) mode**: It operates in the same way as the synchronous stream ciphers.

Commonly, a block cipher processes an encryption through a number of iterations called *rounds*. Each round operates on a fixed length block and involves the computation of linear operations (*i.e.* permutations) and non-linear operations (*i.e.* substitutions). A *round subkey* used during these iterative rounds, is derived from a secret master key by applying a *key scheduling function*. We present hereafter the two main architectures of block ciphers: a *Feistel Network* with the *Data Encryption Standard* (DES) and a *Substitution Permutation Network* (SPN) with the *Advanced Encryption Standard* (AES). Both standard block ciphers are widely used.

1.2.3 Standard Block Ciphers

1.2.3.1 Feistel Network: the Data Encryption Standard

The Feistel network was designed by Horst Feistel in 1970 and originally called *Lucifer*. It is a symmetric structure based on complex mathematical functions used since in the construction of several block ciphers including the DES. The round transformation of a Feistel network is shown in Fig. 1.1.

In practice, the plaintext is divided into two parts L_0 and R_0. If we denote by f the round function and K_i the subkey for the round i, then at the $(i+1)^{\text{th}}$ round

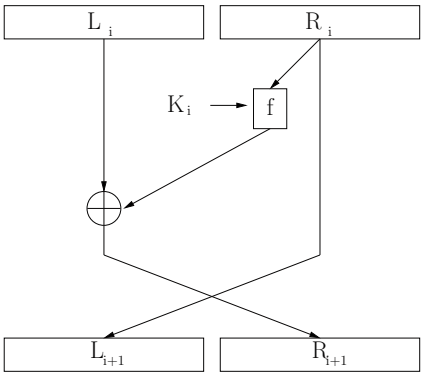


Figure 1.1: Feistel global structure.

we compute:

$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= L_i \oplus f(R_i, K_i) \ . \end{aligned}$$

By iterating this round transformation we obtain the ciphertext (L_r, R_r), where r denotes the total rounds number. The choice of the round function f depends on the security of the scheme.

The DES is the most famous Feistel scheme [NIS99]. It processes on 64-bit blocks and uses a 56-bit key (represented on 64 bits including 8 parity check bits). The encryption starts with an initial bit permutation on 64-bit blocks followed by 16 iterations applying the same round function f and finally a permutation which is the inverse of the initial one.

Every round function f takes a 48-bit round subkey as a parameter, derived from the secret master key by applying the key scheduling function, and operates on a 32-bit block. After the initial permutation, the 64-bit block is split into two 32-bit parts L (left part) and R (right part). The round function operates only on one half of the block and can be defined at the $(i+1)^{\text{th}}$ round by the following equation:

$$f(R_{i+1}, K_{i+1}) = P(S(E(R_i) \oplus K_{i+1})),$$

where:

- E: is the *expansion* function that expands the 32-bit input to a 48-bit output by duplicating 16 of them. Then, the outputs are XORed with the round subkey.
- S: is the *substitution* function composed of 8 different *substitution box* (S-box). The key mixing operation output is split into 8 blocks of 6 bits, each entering into a different S-box produces a 4-bit output block.
- P: is the bit permutation function applied on the 32-bit output of the substitution layer.

The DES algorithm is illustrated in Fig. 1.2.

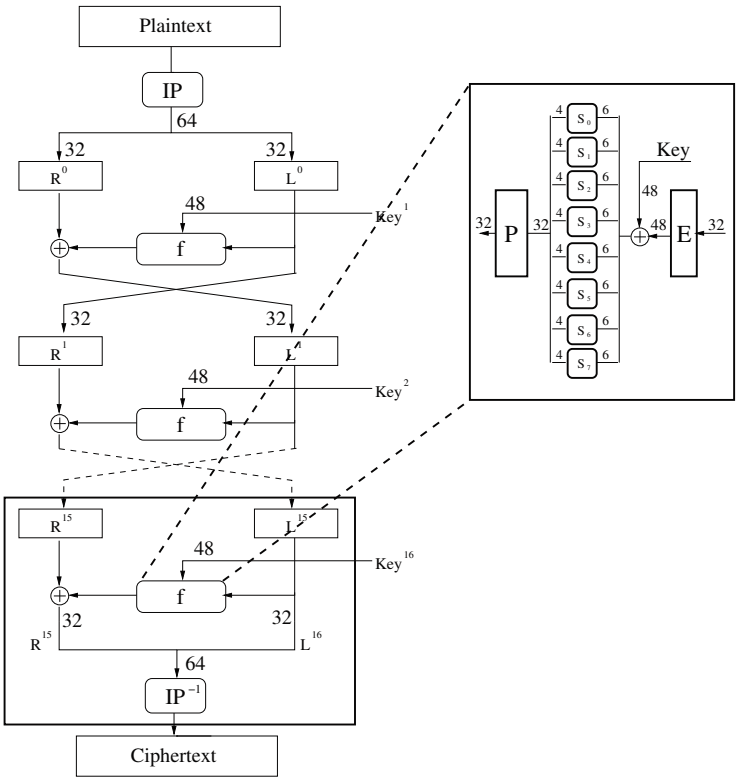


Figure 1.2: Data encryption standard.

1.2.3.2 Substitution Permutation Network: the Advanced Encryption Standard (AES)

The Substitution Permutation Network (SPN) is the second most common block cipher architecture. Unlike the Feistel network, the SPN round function operates on the hole input block. Each round function consists of three stages, or layers: the key-mixing stage, the substitution stage and the linear transformation stage. Figure 1.3 describes the SPN structure.

The Advanced Encryption Standard (AES) [NIS01] is an SPN based algorithm. It processes data using blocks of 128 bits and a variable secret key block size (128, 192 or 256 bits). Hence, the standard [14001] specifies three different block-ciphers: AES-128, AES-192, AES-256. Depending on the key size, we need 10 rounds for AES-128, 12 rounds for AES-192 and 14 rounds for AES-256 to compute the ciphertext. The plaintext, the key and the ciphertext can be represented by a 4x4 matrix of bytes, the same for the intermediate values of the rounds that we call state matrix.

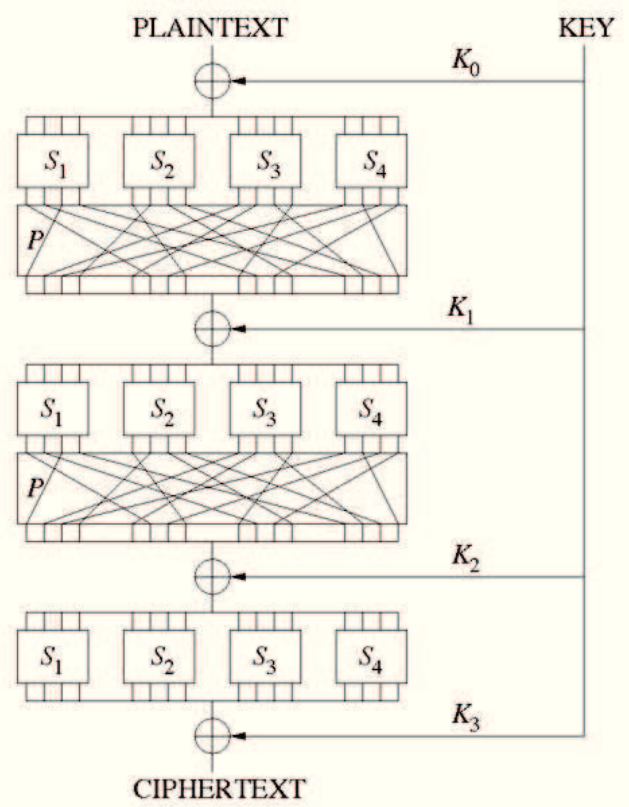


Figure 1.3: Substitution permutation network (this figure is taken from [web]).

Each round of AES is composed of four stages:

- **SubBytes**: modifies each byte in the state using an 8 bits non-linear S-box.
- **ShiftRows**: rotates the bytes in each row of the state. The shift value is 1 for the second row, 2 for the third and 3 for the fourth.
- **MixColumns**: performs a matrix multiplication on the state.
- **AddRoundKey**: mixes the state with a subkey by a bit-wise XOR.

We noticed that the AES-128 encryption starts with an AddRoundKey operation computed between the input key and the plaintext followed by 9 round functions. Finally, the tenth round omits the MixColumns operation. The AES encryption is illustrated in Fig. 1.4.

Symmetric ciphers are computationally strong but the *key exchange management* is an important issue. Asymmetric ciphers provide a solution to this problem.

1.3 Asymmetric Cryptography

Asymmetric cryptography is also called *public-key cryptography* as it does not use the same key for the encryption and decryption process. It was invented in 1976 by

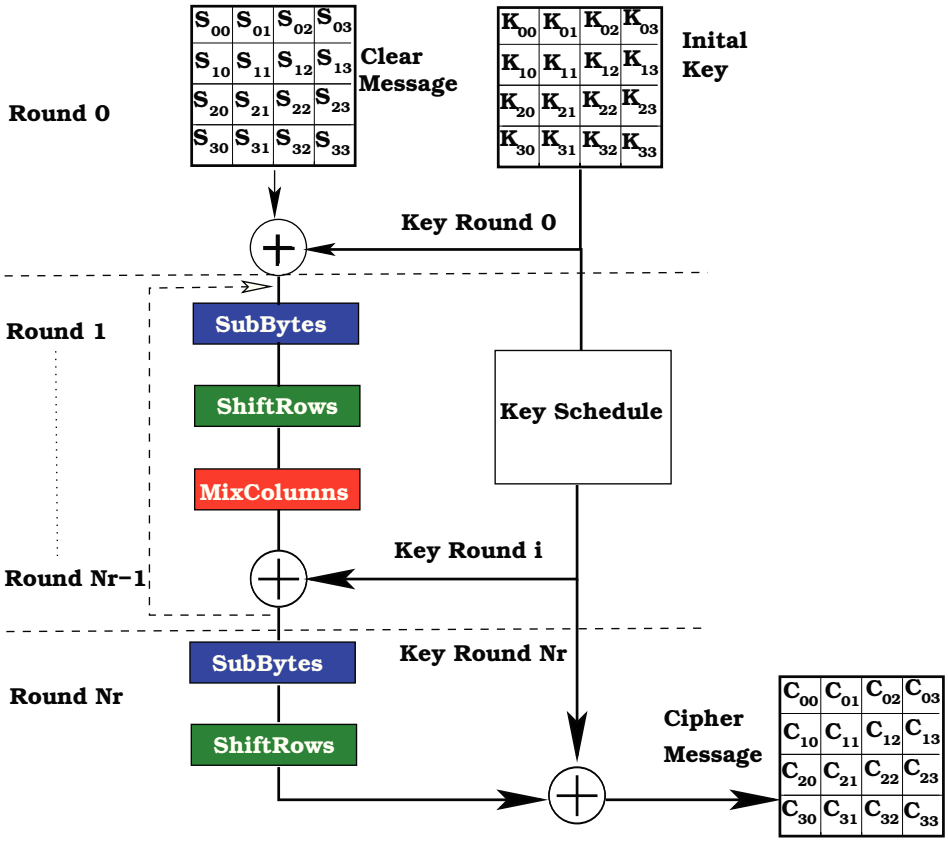


Figure 1.4: Advanced encryption standard.

Diffie and Hellman as a solution to the key exchange issue. They proposed the use of two separate but dependent keys: *the public key* and *the private key*. More precisely, the public key is used for encryption (or signature); and alternatively the private key is used for decryption (or signature verification). Anyone can use the public key to cipher a message since it is publicly deployed. However, the private key is kept secret by its owner. In the literature, several asymmetric cryptography algorithms have been proposed. The RSA cryptosystem, developed by R. Rivest, A. Shamir, and L. Adleman, is the most commonly used public-key cryptosystem. Essentially, *"it is based on the intractability of the integer factorization problem"* [Yan09]. Recently, public-key algorithms have started using the elliptic curve cryptography, introduced by Neal Koblitz and Victor Miller, they provide smaller key sizes and faster enciphering/ deciphering operations.

The interested reader is refereed to [Sal96, DM09] for detailed explanations about public-key cryptography.

1.4 Cryptographic Devices and Physical Attacks

1.4.1 Cryptographic Devices

Cryptographic modules are *"the set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module"* [14009]. They are currently used in many electronic devices of everyday's life. The most common cryptographic module is the smart card. This device was invented in 1970 as a practical answer to the *key storage issue*. In fact, secret key used in modern cryptography must be stored in a physical device such as a personal computer memory. The first smart cards were *memory cards* using an inhibitors (*e.g.* password) to protect the memory access. Since 1986, the microprocessor cards were widely used in many everyday life applications. Those are composed of a microprocessor to execute the cryptographic operations and several types of memory (ROM, EEPROM, RAM). In this type of device, operations are sequentially performed by the microprocessor at a speed dependent on its clock frequency. Nowadays, smart cards are used in different applications of everyone's daily life (*e.g.* SIM (Subscriber Identity Module) card, credit card, electronic passport, *etc.*).

The cryptographic cipher can also be implemented in a hardware device such as FPGA (*Field Programmable Gate Array*) and ASIC (*Application Specific Integrated Circuit*). FPGAs offer high performance and are used in many domains including software-defined radio, aerospace, medical imaging, cryptography and in a growing range of other areas. In this thesis, we try to answer the question: Do cryptographic module really provide the secrecy? Do hardware implementations of cryptographic ciphers protect our secret key?

1.4.2 Physical Attacks

Modern cryptography is mathematically secure against several analytical cryptanalysis attacks. But, the cryptographic algorithms are designed to be used in real-life applications. In fact, these algorithms are implemented as explained in the previous section in cryptographic modules or devices using some program codes (software implementation) or logic components (hardware implementation). So, the mathematical representation of the algorithm is converted to a physical implementation where the secret key is embedded or processed. The cryptographic device, while running cryptographic computation, leaks some information about the sensitive variable. The leakage can be the execution time, the power consumption and the electromagnetic radiation produced during the computation. The attacks that target the physical implementation and not the weakness of the algorithm in order to extract the secret key are called *physical attacks*. Such kinds of attack are very powerful and must be considered when designing a cryptographic device.

Physical attacks include two main types: *fault attacks* (FA) [BS97] and *side channel analysis* (SCA) [KJJ99]. The fault attack is an *active attack*: aims at shifting the attacked device from its normal behavior and analysing its response. A fault

attack consists in injecting hardware malfunctions (*e.g.* power spikes, clock glitches) in order to produce erroneous results exploitable to recover the secret information from electronic devices. Unlike FA, a side channel attack is a *passive attack*: does not disturb the system resources and behavior. It analyses and exploits the physical leakage from the cryptographic device while running cryptographic computations.

Physical attacks can be further divided into three classes [MOP06] depending on how the adversary breaches the cryptographic boundary:

- **Invasive attacks**: First, the chip is depackaged, then the passivation layer is removed at least part to perform *probing attacks* [HPS99].
- **Semi-invasive attacks**: Like invasive attacks, the chip is removed. However, the system surface is not altered.
- **Non-invasive attacks**: The adversary does not alter the physical integrity of the device; he performs power measurements (SCA) or disturbs its behaviour by injecting faults (FA).

1.5 Conclusions

In this chapter, we introduced few basics of modern cryptography. We described the most used cryptographic algorithms and especially block cipher standards. Then, we concluded that these algorithms are designed to be used in real-life applications and implemented in electronic devices. This physical implementation leaks information about the sensitive secret, even if the algorithm is mathematically secure.

In this thesis, we focus mainly on the protection against side channel attacks. Therefore, we explore the know-how of SCA in the next chapter.

Chapter 2

Introduction to Side Channel Analysis

In this chapter, we give a general introduction to side channel analysis. Then, we present some classical SCA models and metrics used for the security evaluation. Thereafter, we provide a general survey of the main existing attacks and countermeasures. Finally, we introduce the higher-order SCA context.

Contents

2.1 SCA: General Background

SCA is a branch of cryptography that exploits the information leaking from the physical implementation of a cryptographic cipher. The first historical example of analysing the electromagnetic emanations is the US project, so-called *TEMPEST*. It provides solutions and some standards to avoid the physical emanations based attacks. In 1985, Wim Van Eck published the first paper [Eck85] that demonstrated how to display a nearby computer electromagnetic emission on a video monitor. In the mid-1990s, Paul Kocher *et .al* [KJJ99] proposed the first side channel attack that broke several classical implementations including DES and RSA. These publications gave rise to many developments of new attacks and countermeasures. SCA became an established scientific discipline: several conferences and workshops took place and present considerable works in this field. Side channel can be divided depending on the nature of the exploited physical information: execution time, power consumption and electromagnetic radiations.

2.1.1 Timing Attacks

The timing attacks were introduced by Kocher *et al.* in [KJJ96]. Every logical instruction takes time to execute which can differ depending on the input. An attacker can select the plaintext inputs and analyze the computation time. For instance, in case of RSA algorithm the execution time depends on the key bit values: the circuit will perform a square operation when the key bit is 0 else a square operation followed by a multiplication. Therefore, the attacker by detecting the difference of execution time can reveal the secret information.

2.1.2 Power Attacks

Most recent electronic devices mainly consist of *Complementary Metal Oxide Semiconductor* (CMOS) cells. The root of vulnerability to side channel attacks is related to the behaviour of those CMOS cell. In fact, the power consumption of these components depends on the outputs transitions. There is a clear difference, in terms of power consumption, between transition from $0 \rightarrow 1$ or $1 \rightarrow 0$ where the bit value changes and from $0 \rightarrow 0$ or $1 \rightarrow 1$ where it doesn't. An adversary, by observing these transitions via the global power consumption of the cryptographic module, can reveal information about the sensitive variable.

2.1.3 Electromagnetic Attacks

Electromagnetic (EM) attacks are similar to power consumption based attacks. The attacker captures the EM radiations from the target device using an EM *probe*. Depending on the used probe, the EM radiations could be global or localized. The main advantages of localised EM are to isolate some specific part of the circuit and to reduce the noise level.

The power consumption and the EM emanations depend on the variable being processed. Both attacks shall be treated together and referred to as power attacks in the rest of this thesis. After identifying the physical leakage, the attacker collects a large set of power consumption measurements (*traces*) with different plaintexts. Figure 2.1 shows a power consumption trace of the DES algorithm with its 16 rounds.

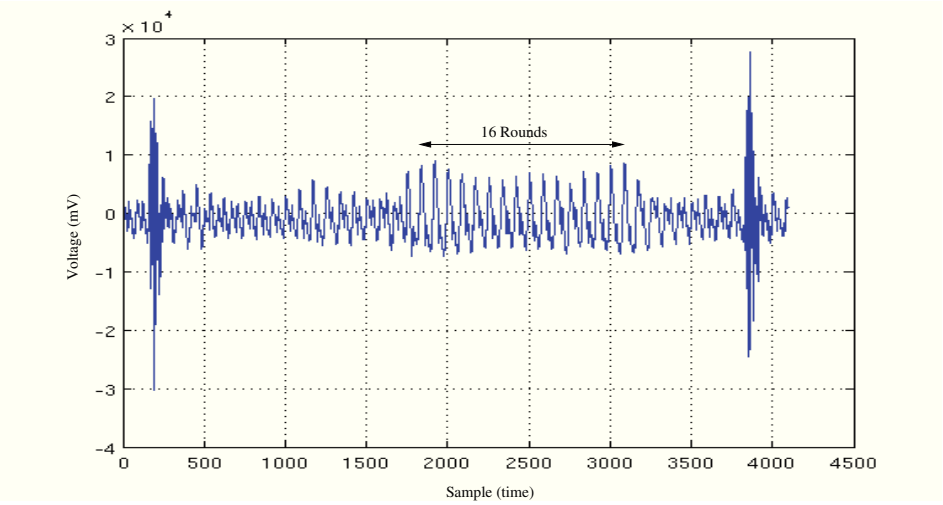


Figure 2.1: Power consumption of an FPGA implementation of the DES algorithm.

Thereafter, the adversary makes hypotheses about the secret key and builds an appropriate *leakage model*. Finally, he uses a *distinguisher* to identify the statistical dependency between the power measurements and the leakage model to extract the secret key. In next sections, we discuss further side channel models and some classical distinguishers commonly used.

2.2 Side Channel Analysis Models

2.2.1 Attack Model

The goal of this section is to formalize the attack model and introduce some notation needed to explain the state-of-the-art of some classical attacks. Let us denote:

- L: a random variable (RV) that represents the leakage (*e.g.* the measured current drawn by a cryptographic design).
- K: the secret cryptographic key, an n-bit data; unknown to the attacker and assumed to be uniformly distributed on $\mathbb{F}_2^n$.
- X: the input or the output of the cryptographic design (*i.e.* its plaintext or ciphertext); known by the attacker.

- $Z = f(X, K)$ for a given function f: a sensitive variable used internally, that depends only on X and K. We assume that this sensitive variable Z can be computed exhaustively from all possible K by the attacker, furthermore Z causes the leakages; put differently, when the key guess is correct, Z and L are dependent.

SCA consists in estimating, for every key guess (*i.e.* for every value k of K), whether the RVs Z and L are dependent. The analysis is said "sound" if the greatest dependence is obtained for the correct value of the key, noted k^*. In this case, the key can be extracted successfully from the device. In practice, the values taken by L are noisy, because they consist in physical measurements and thus the link between Z and L is imperfect. Therefore, many couples (X, Z) are required for the 2^n estimations (for each value k of K) to find the correct key.

2.2.2 Leakage Models

In SCA, the attacker tries to find the relation between the data being processed and the leakage measurements. The leakage can be represented as the sum of two parts: a deterministic part $\varphi(.)$ (depending on the sensitive variable Z) and an independent noise N such that:

$$L = \varphi(Z) + N\,,$$

where $\varphi(.)$ represents the leakage function. We assume that the noise N has a Gaussian distribution. This approximation is fairly realistic and commonly used in the literature [CRR02, SLP05, MOP06]. As discussed in Sec. 2.1.2, the leakage results from logical transitions during the computation. So, it is reasonable to assume that the leakage comes from the state update of each bit. This model is the so-called *Hamming distance model* (HD) and the leakage can be expressed as follows:

$$\begin{aligned} L &= \mathsf{HD}(Z, R) + N \\ &= \mathsf{HW}(Z \oplus R) + N\ , \end{aligned}$$

where R is the reference state. A particular case is when the reference state is initialized to zero. This yields the *Hamming weight model* (HW) which is expressed as:

$$L = \mathsf{HW}(Z) + N\,.$$

After choosing the appropriate leakage model, the adversary has to select a statistical tool (distinguisher) in order to determine the dependency between the leakage measurements and the model. We review in what follows the most common distinguishers used in side channel key recovery attacks.

2.3 Classical Side Channel Distinguishers

2.3.1 Correlation

Distinguishers based on correlation are the most commonly deployed to achieve a successful recovery attack. We give hereafter two examples of those distinguishers: the *Difference of Means* (DOM) which is used in the original *Differential Power Analysis* (DPA) [KJJ99] and the *Pearson correlation coefficient* used in the *Correlation Power Analysis* (CPA) [BCO04].

2.3.1.1 Difference of Means

In practice, the attacker collects the power traces corresponding to the encryption of many randomly selected plaintexts (or context's). Then, he makes a guess on the key k and predicts one bit value of the sensitive variable $Z = f(X, k)$, let's say the bit Z_j. The leakage measurements L are separated in two partitions according to the hypothetically predicted value $Z_j = 0$ or $Z_j = 1$ computed for each key hypothesis. Finally, the adversary computes the difference of means of the two partitions:

$$\Delta(k) = \mathbb{E}[L \mid Z_j = 0] - \mathbb{E}[L \mid Z_j = 1] \ ,$$

where $\mathbb{E}[.]$ denotes the expectation (the mean). The correct key k^* is identified as the argument that maximizes the difference of means $\Delta(k)$. Later, the so-called *mono-bit DPA*, was generalised to the *multi-bit DPA*. This distinguisher targets the consumption activity of several bits of the sensitive variable Z and computes a sum of the centered weights of the considered partitions sets.

2.3.1.2 Pearson Correlation Coefficient

This distinguisher was introduced by Éric Brier *et al.* in [BCO04] as a natural generalisation of DPA. The CPA attack based on this distinguisher is the most widely used technique in side channel field. The attacker predicts the appropriate leakage model function $\varphi(Z)$, then estimates the Pearson correlation coefficient ρ_k for every key guess k:

$$\rho(k) = \frac{\mathsf{Cov}[L; \varphi(Z)]}{\sigma_L \times \sigma_{\varphi(Z)}} \ ,$$

where $\mathsf{Cov}[.;.]$ is the covariance and σ_L and $\sigma_{\varphi(Z)}$ are respectively the standard deviation of the physical leakage and the leakage model. We notice that CPA and DPA are almost equivalent. The main advantage of the CPA over DPA is that it reduces the noise affecting the leakage measurements.

2.3.2 Mutual Information

In 2008, Gierlichs *et al.* propose a new side channel distinguisher called Mutual Information Analysis (MIA) [GBTP08]. It is an attractive alternative to the previous distinguishers. In fact, it exploits any kind of dependency between the leakage measurements and the predicted data, as opposed to correlation distinguishers which require

a linear relationship to successfully recover the good key. The MIA has been largely studied and tested on several implementations [GBTP08, VCS09, PR09, BGP+11]. Once the leakage model is chosen, the adversary estimates the mutual information between the leakage measurements and the leakage model for every key guess k:

$$\mathrm{MIA}(k) = \mathsf{H}[L] - \mathsf{H}[L \mid \varphi(Z)]\,,$$

where $\mathsf{H}[.]$ is the entropy. The correct guess of the key k^* corresponds to $\arg\max_k \mathrm{MIA}(k)$. The issue with mutual information is that it is based on the *Probability Density Function* (PDF) estimation techniques. Several methods have been proposed in the literature: histograms, kernel density function, parametric estimation. The choice of the appropriate estimation method is a real challenge. We further discuss those methods in Chapter 4.

2.3.3 Kolmogorov-Smirnov

In the context of SCA, the Kolmogorov-Smirnov (KS) test has been suggested first in [VCS09] as a non-parametric statistical tool to distinguish between distributions. Then, the authors in [WOM11] investigate the potential of the Kolmogorov-Smirnov Analysis (KSA) and compare it with MIA. The KSA distance is a simple measure which is defined as the maximum value of the absolute difference between the *cumulative distribution functions* (CDFs) of two RVs X_1 and X_2: $D_{\mathrm{KSA}} = \sup_{x\in\mathcal{X}} |F_{X_1}(x) - F_{X_2}(x)|$, where F_{X_1} and F_{X_2} are the empirical CDFs (ECDFs). By definition an ECDF is a step function defined as: $F_X(x) = \frac{1}{n}\sum_{i=1}^{n} I_{x_i \leq x}$, where the tuple $\{x_i\}_{i\in[\![1,n]\!]}$ denotes the values realized by the RV X.

Like MIA, the KSA distinguisher measures the maximum distance between the leakage L and the hypothesis-dependent conditional observations $L \mid \varphi(Z)$:

$$\mathrm{KSA}(k) = \mathbb{E}_Z \sup_{l\in\mathcal{L}} \left| F_L(l) - F_{L|\varphi(Z)}(l) \right| \quad .$$

The KSA returns the largest difference when the key is correct, *i.e.* when $k = k^\star$.

2.3.4 Likelihood

The likelihood distinguisher is used in the context of the so-called *profiled* attack. In such SCA, the attacker proceeds in two steps. First, a profiling phase where an adversary has access to a clone device that allows him characterizing its physical leakage model. Second, he exploits these information on a similar target device in order to perform a key recovery based on the maximum likelihood principle. Profiled attacks include template attacks [CRR02] and stochastic models [SLP05]. Despite the disadvantage of the necessity of the profiling phase, these attacks are very powerful and very useful for evaluating SCA countermeasures.

2.4 Side Channel Metrics

Many tools have been devised to analyse the leakage of cryptographic devices and to attack them. Notably, leakage metrics are used to quantify the leakage of a device, and the security metrics are used to measure the strength of an attack [SMY09]. We list hereafter the most common metrics used in side channel.

2.4.1 Leakage Metrics

2.4.1.1 Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) is a relevant metric to characterize any analog phenomena, such as the side channel measurements. The SNR is defined as:

$$SNR = \frac{\mathsf{Var}[\varphi(Z)]}{\mathsf{Var}[N]} \ ,$$

where $\mathsf{Var}[.]$ is the variance. The signal is characterised by the leakage model $\varphi(Z)$. The noise is traditionally estimated by computing the variance of all the traces. The noise of a side channel acquisition campaign has at least two kinds of origins as stated in [MDS02, §3.1], namely:

1. the quantization (vertical and horizontal) and the environmental parasitics (intrinsic and external), and
2. the algorithmic noise (*i.e.* the rest of the circuit).

2.4.1.2 Correlation

Another metric used to evaluate the physical leakage is to compute the Pearson correlation coefficient between the leakage and the leakage model. Eventually, the Pearson correlation coefficient involved in CPA [BCO04] is proportional to SNR; thus, it is directly connected to the quality of the acquisition. Moreover, It has been demonstrated that the number of waveforms required to break a cryptographic implementation by CPA is equal to [Man04]:

$$3 + 8\left(\frac{Z_{1-\alpha}}{\ln\left(\frac{1+\rho}{1-\rho}\right)}\right)^2 \ ,$$

where $Z_{1-\alpha}$ is a quantile of a normal distribution for the 2-sided confidence interval with error $1-\alpha$. Some values of quantiles are given in Tab. 2.1.

2.4.1.3 Mutual Information Metric (MIM)

This metric is based on an *information-theoretic* evaluation of the leakage. To quantify the information contained in the leakage measurements, the attacker computes the mutual information between the physical leakage L and the secret key K (or

Table 2.1: Some precomputed values for the quantiles of a normal distribution.

Confidence level [%] $(1-\alpha)$	$Z_{1-\alpha/2}$
60	0.842
80	1.282
90	1.645
95	1.960
98	2.326
99	2.576

the sensitive variable Z). Mutual information metric measures the quality of an implementation; higher the information leaks, lesser is the resistance of the implementation. Thus, MIM is often used to compare the strength of countermeasures.

2.4.2 Attack Metrics

2.4.2.1 The Minimum Traces to Disclosure (MTD)

The MTD quantifies the side channel attack efficiency. It indicates the average number of measurements needed to perform a successful attack.

2.4.2.2 Success Rate

The success rate is defined as the probability that the attack's best guessed key is the correct key given a set of leakage measurements. In [SMY09], Standaert *et al.* extend the notion of success rate to the higher-order context. For instance, if we consider success rate of order d, this implies that the attacker still has at most d key candidates to test after the attack in order to recover the good one. In this thesis, we use the first-order success rate in our security evaluation.

2.4.2.3 Guessing Entropy

Success rate is useful only when the attack is successful. Sometimes, when analyzing countermeasures the attack does not find the good key. In such cases, *guessing entropy* is a useful metric [SMY09]. It measures the average number of key guesses to test before finding the correct key. It can be defined also as the position of the target key in a list of key hypotheses ranked by a distinguisher.

2.5 Side Channel Countermeasures

We review hereafter the most common countermeasures proposed to hinder SCA attacks.

2.5.1 Noise Generators

One classical countermeasure consists in adding a hardware module, called *noise generator* [Man04], aiming at introducing some noise to the leakage measurements. To amount a successful key recovery attack, the adversary needs more power consumption traces which is time consuming (and needs much more memory to be stored). The amount of noise added to the implementation allowed reaching a high security level. For these reasons, noise generators are considered as a sound countermeasure in practice. But theoretically, it is not sufficient to counteract practical attacks.

2.5.2 Shuffling

Shuffling is one of the most frequently considered countermeasures to improve the security of cryptographic devices in software against side channel attacks. It consists in randomizing the execution order of independent operations [RPD09].

2.5.3 Hiding

One common countermeasure widely used against side channel attack is hiding. In literature, we can find several papers dealing with this solution also called *dual-rail* [CZ06, BZ08, AKM+08, GCS+08, GSF+10, KV10]. It is applied at the logic gate level and consists in making the activity of the physical implementation constant by adding complementary logic to the existing logic. Therefore, making this activity constant would theoretically remove the correlation between the leakage measurements and the data being processed (the secret key). However, in practice it is very difficult to guarantee a symmetry hypothesis in the hardware due to small load imbalances between the two complementary circuits, process variations, routing, *etc.* This imbalance always leaks some information that can be exploited by a SCA attacker.

2.5.4 Masking

Masking is one of the most widespread countermeasures. It can be implemented at the gate level or at the algorithmic level. When masking is used to secure a circuit at the gate level, every input of the logic gate is XORed with a random value called the *mask*. Hence, the logic gate outputs are statistically independent of the input value. This technique suffers several problems such as the occurrence of *glitches* [FG05, NRS08]. In this thesis, we are interested in the study of the algorithmic masking technique. We refer the interested reader to [Tri03, SSI04, FG05, CZ06] for more detailed explanations about masking solutions at the logic level.

When applied at the algorithmic level, a d^{th}-order masking scheme aims at replacing the manipulation of the sensitive variable Z by the manipulation of a vector of $d+1$ variables $S_0, \cdots, S_d$ called shares, such that $Z = S_0 \perp \cdots \perp S_d$, where $\perp$ is a group operation. Obviously, a d^{th}-order masking can always be theoretically

defeated by a $(d+1)^{\text{th}}$-order SCA attack that jointly involves all the $d+1$ shares. The design of masking scheme is further discussed in the third part of this thesis.

2.6 Higher-Order Side Channel Analysis

We call higher-order side channel attack [CJRR99, Mes00b, WW04, JPS05, PSDQ05, OMHT06, SP06] (abridged HO-SCA), any attack that combines all shares [GBPV10] or a judicious combination [PRB09] of them in order to turn a successful key recovery. Later, the attacker computes the dependency between the combined leakage and the leakage model for every key guess k using a distinguisher. When the correlation (respectively the mutual information) is used as a distinguisher, the attack is called higher-order differential power analysis (HO-DPA/ HO-CPA) (respectively higher-order MIA (HO-MIA)).

Several *combining functions* are used in HO-SCA. The choice of the combination depends on the implementation (*e.g.* software or hardware implementation). In fact, the SCA attacker of a d^{th}-order masking scheme gets direct observations of the $(d+1)$-tuple L in the software case since the variables are evaluated sequentially. So, the leakage L is a multivariate signal and such an attack is called *multivariate attack*. Then, the adversary can thus choose adequate combining functions using several measurements [PR07]. Mainly two combining functions have been previously studied:

- In [CJRR99], Chari suggests performing the product of the shares.
- In [Mes00b], Messerges combines the shares using the absolute difference.

In [PRB09], Prouff *et al.* analyzed both the product and the absolute difference combining functions and they show that the product combining is the best published function to perform a second-order DPA when devices leak the Hamming weight of the processed data.

In hardware implementations, ASIC or FPGA, the variables are evaluated in parallel. So, the only combination function available to the attacker is the arithmetic sum of individual leakages of the shares. This is done physically because of the parallelism in hardware devices execution. Hence, the leakage L corresponds to a univariate signal and such an attack is called *univariate attack*.

2.7 Conclusions

In this chapter, we described the general background about side channel metrics, attacks and the corresponding countermeasures. Amongst hiding and masking, which are the two major countermeasures against SCA (see respectively Chapter 7 and 9 of [MOP06]), the latter is certainly the simplest one to implement when applied at the algorithmic level. As shown by Chari *et al.* in [CJRR99], it moreover enables to have a lower bound on the achieved security level in terms of the number of measurements needed to retrieve secret information with a given success probability.

In this thesis, the focus is laid upon the study of Boolean hardware masking countermeasure and the enhancement of its resistance against higher-order attacks. In the next part, we propose new higher-order distinguishers that can defeat masking countermeasures.

Part II
Higher-Order Side Channel Attacks

Chapter 3

Variance-based Power Analysis

In this chapter, we propose a new distinguisher, called *Variance-based Power Analysis* (VPA). This attack is based on a variance analysis of the observed power consumption. We performed it against an FPGA implementation of a masked DES. The experimental results confirm the overwhelming advantage of this distinguisher. As for the most common univariate SCA attacks, VPA can be reformulated to reveal a correlation coefficient computation by changing the leakage model.

Parts of the results presented in this chapter have been published in collaboration with Sylvain Guilley, Florent Flament and Jean-Luc Danger in the international conference on *Signals, Circuits and Systems (SCS 2009)* [MDFG09].

Contents

3.1 Vulnerability of Masking in Practice

Masking can be defeated if the attacker knows how to combine the leakages corresponding to the masked data $Z \oplus M$ and its mask M. This is known as second-order power analysis (abridged 2O-DPA) and was originally suggested by Messerges in [Mes00b]. Investigating 2O-DPA is of major importance for practitioners as it remains a good alternative that is powerful enough to break real-life DPA-protected security products. In the sequel, we overview univariate second-order attacks since we are interested in the hardware setting.

In [WW04], Jason Waddle and David Wagner proposed a second-order attack that consists in computing the difference of means of the square of the power consumption traces in order to obtain key-dependent measurements, instead of the difference of means of the traces as in DPA. Peeters *et al.* present in [PSDQ05] another attack against a masked FPGA countermeasure based on the maximum likelihood distinguisher.

In the next section, we review the attack proposed in [PSDQ05]. Then, we propose a new second-order attack called Variance-based Power Analysis. The VPA is similar to the *variance test* proposed by Standaert *et al.* in [SGV08] where the variance is used as substitute to the entropy used in the computation of the mutual information, albeit with little variations.

3.2 Evaluation of the VPA Attack

3.2.1 Peeters's Attack

Assuming a Hamming weight model, the deterministic part of the leakage L of a first-order masking scheme can be expressed as: $L = \mathsf{HW}(Z \oplus M) + \mathsf{HW}(M)$. Considering 4-bit variables, there are five possible distributions of the leakage depending on the Hamming weight of the secret state values $\mathsf{HW}(Z)$, when the key is correct, as shown in the top of Fig. 3.1. For instance, if $\mathsf{HW}(Z) = 0$ (*i.e.* Z is equal to the hexadecimal value `0x0`), then the corresponding leakage is $L = 2 \times \mathsf{HW}(M)$ and thus takes value in $\{0, 2, 4, 6, 8\}$ since the mask M is random and unknown by the attacker. If $\mathsf{HW}(Z) = 4$ (*i.e.* Z is equal to the hexadecimal value `0xf`), the corresponding leakage is $L = \mathsf{HW}(\overline{M}) + \mathsf{HW}(M)$ and is constantly equal to 4. When the key is incorrect (bottom of Fig. 3.1), the leakage corresponds to that of the function L where:

- Z is uniformly distributed in $[\texttt{0x0}, \texttt{0xf}]$, because the guessed key is wrong,
- M is uniformly distributed in $[\texttt{0x0}, \texttt{0xf}]$.

Two important consequences can be emphasized from Fig. 3.1:

- The distributions $\mathsf{P}[L \mid \mathsf{HW}(Z)]$ for the correct key have the same mean value and only differ in their variances.
- Knowing the Hamming weight value of the secret intermediate variable $\mathsf{HW}(Z) = a$, the attacker knows the probability density function $\mathsf{P}[L \mid \mathsf{HW}(Z) = a]$.

The attack proposed in [PSDQ05] consists in analysing the probability density function based on a maximum likelihood approach to amount a successful key recovery. The attack algorithm is the following:

1. Encrypt n plaintexts (x_i, $i \in [1, n]$) and collect n observations of power consumption leakages (traces L_i).
2. For each assumption about the key k_j with $j \in [0, 63]$, compute a set of secret states according to $\mathsf{HW}(Z_j) = \mathsf{HW}(S(X \oplus k_j))$, where S is the DES S-box function:

$$\begin{cases} \mathsf{HW}(Z_0) = \mathsf{HW}(S(x_0 \oplus k_0)), \mathsf{HW}(S(x_1 \oplus k_0)), \cdots, \mathsf{HW}(S(x_n \oplus k_0)), \\ \mathsf{HW}(Z_1) = \mathsf{HW}(S(x_0 \oplus k_1)), \mathsf{HW}(S(x_1 \oplus k_1)), \cdots, \mathsf{HW}(S(x_n \oplus k_1)), \\ \qquad\qquad \cdots \\ \mathsf{HW}(Z_{63}) = \mathsf{HW}(S(x_0 \oplus k_{63})), \mathsf{HW}(S(x_1 \oplus k_{63})), \cdots, \mathsf{HW}(S(x_n \oplus k_{63})). \end{cases}$$

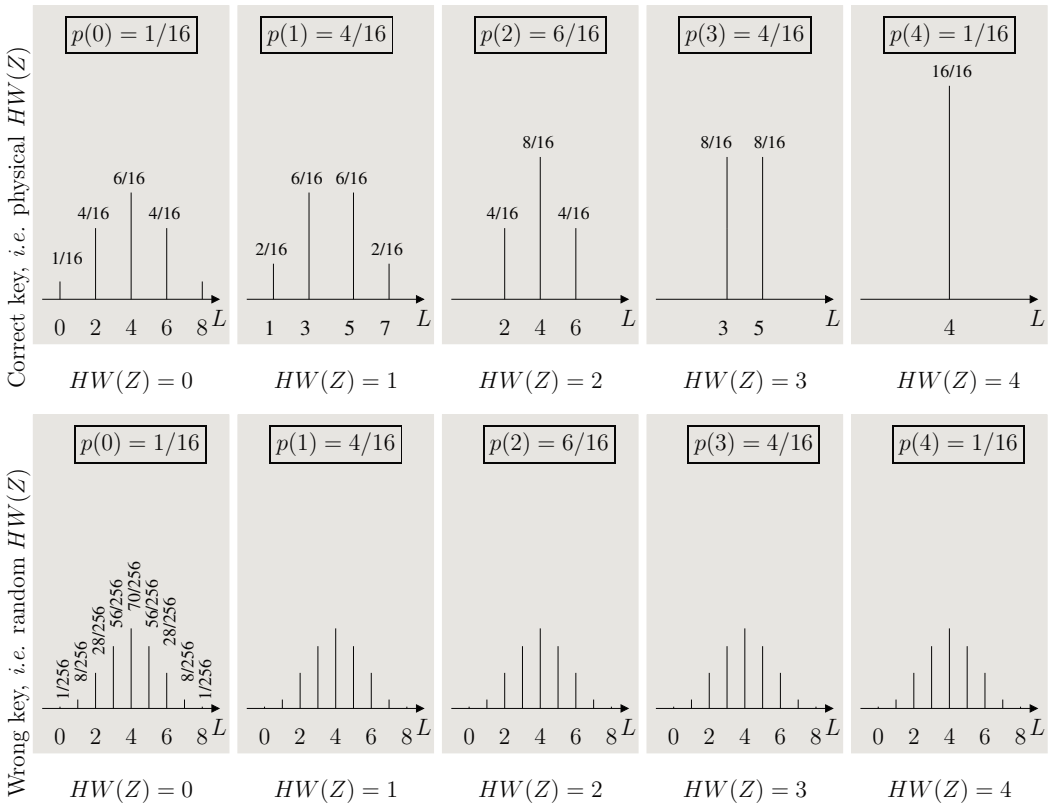


Figure 3.1: PDFs corresponding to the five possible values of $\mathsf{HW}(Z)$.

3. For each set $\mathsf{HW}(Z_j)$, compute the probability:

$$\mathsf{P}[L \mid \mathsf{HW}(Z_j)] = \prod_{i=0}^{n-1} \mathsf{P}[L = L_i \mid \mathsf{HW}(S(x_i \oplus k_j))] \ .$$

4. Apply the maximum likelihood approach: the correct key corresponds to the maximum probability $\mathsf{P}[L \mid \mathsf{HW}(Z_j)]$.

We reproduce this attack on a fully-fledged masked DES implementation in an Altera Stratix II FPGA on the SASEBO-B evaluation board provided by the RCIS [Jap]. The choice of DES algorithm was motivated by the hardware area constraints. In fact, we did not target the masked AES implementation since it requires huge memories resources in FPGA. We implemented the masked DES studied in [SRQ06], whose principle is illustrated in Fig. 3.2 and used as a target scheme in [PSDQ05]. We have designed a complete System on Programmable Chip (SoC) including a master processor for inputs/outputs communications and the masked DES cryptoprocessor.

The attack platform is described in Fig. 3.3. The design communicates with a monitoring PC via a RS-232 serial link. The EM field is collected from the FPGA,

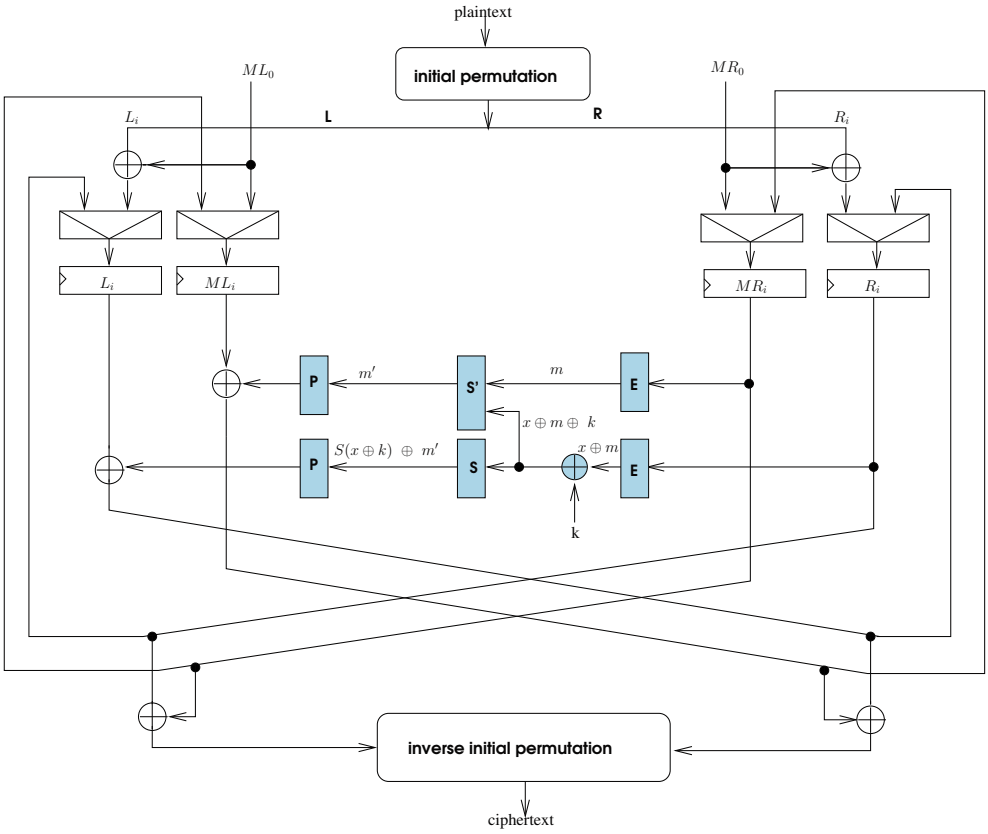


Figure 3.2: Masked DES implementation.

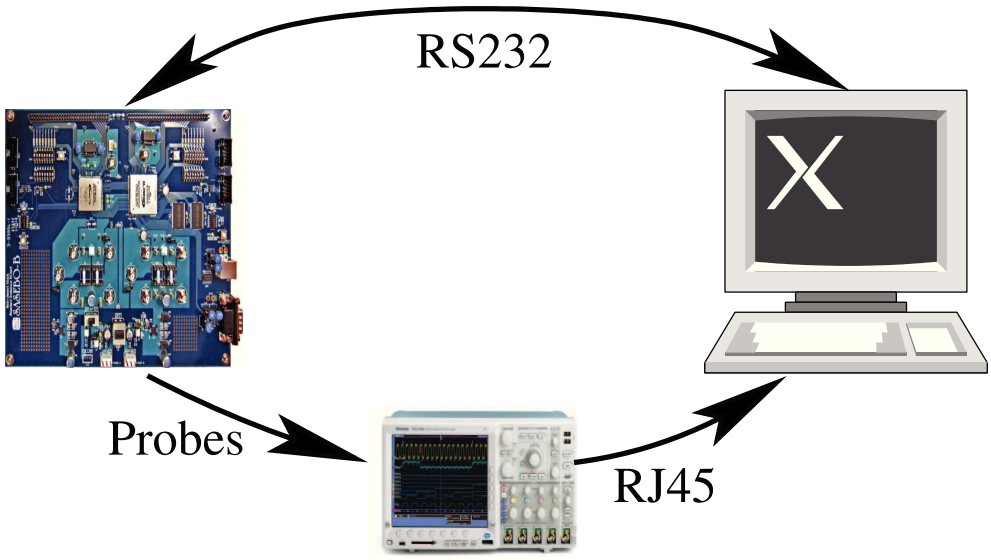


Figure 3.3: Experimental attack platform.

while encrypting messages, with an 54855 Infinium Agilent oscilloscope with 6 GHz bandwidth and a maximal sampling rate of 40 GSa/s and with an antenna of the "HZ-15 kit" from "Rohde and Schwarz' [Roh].

The attack implementation succeeded on noisy simulated traces, but failed when applied to our real world DES implementation even when using 200,000 power consumption traces. This comes mainly from the high level of noise (especially the algorithmic one) when attacking a fully-fledged implementation in FPGA. In fact, the noise coming from other computing blocks and the environment shapes the PDF as a sum of Gaussian distributions. This can be intuitively illustrated through the example of Fig. 3.4. The top left image of the figure shows the PDF that corresponds to the partition $\mathsf{HW}(Z) = 0$ in a noise-free context. The rest of the images show the same leakage model partition with some increasing additive standard deviation of noise ($0 < \sigma_1 < \sigma_2 < \sigma_3$). Obviously, the shape of the leakage distribution is converging towards the normal distribution, when the noise standard deviation is getting higher. So, it is hardly possible to discriminate the different Gaussians. Thus, in this case, the Peeters's attack might lose its efficiency when computing the probability $\mathsf{P}[L \mid \mathsf{HW}(Z)]$.

3.2.2 Proposed VPA Attack

The VPA takes advantage from the fact that the distribution of power consumption has the same mean, but a different variance as shown in Fig. 3.1. For instance, the variance difference between the PDF for $\mathsf{HW}(Z) = 0$ and $\mathsf{HW}(Z) = 4$ should be enough discriminating even without the knowledge of the exact probability density functions. VPA is based on the variance computation of the power consumption

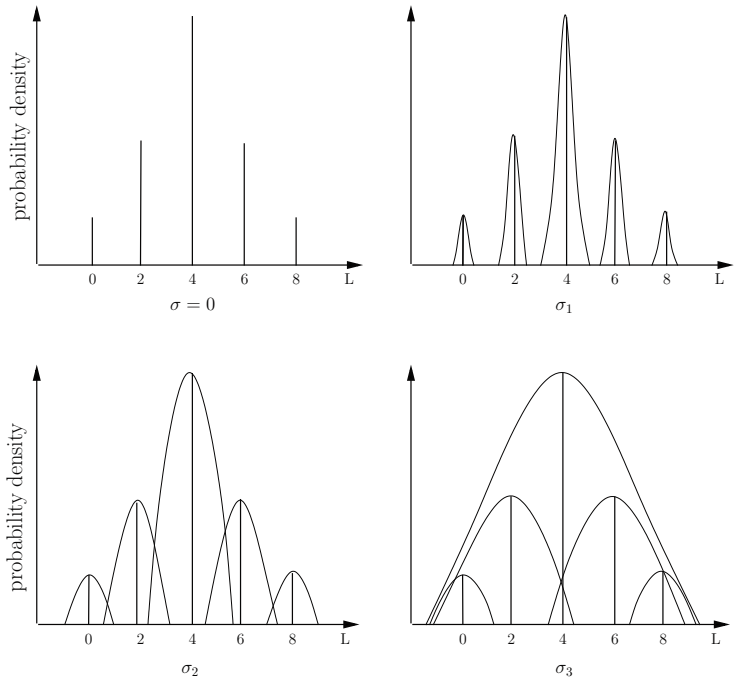


Figure 3.4: Real world PDF for $\mathsf{HW}(Z) = 0$.

traces during a time window corresponding to the first DES round, while ciphering random messages. The VPA algorithm is the following:

1. Encrypt n plaintexts (x_i, $i \in [1, n]$) and collect n observations of power consumption (traces L_i).

2. For each S-box, make assumptions about the key $k \in [0, 63]$:

   - Sort the traces L_i to get five activity sets set_l, $l \in [0, 4]$, corresponding to the five $\mathsf{HW}(Z) = \mathsf{HW}(S(x_i \oplus k))$ possible values.
   - Compute the variance v_l for each set set_l.
   - Compute a VPA indicator VPA(k) being a linear combination of the variances with weights w_l: $\mathrm{VPA}(k) = \sum_{l=0}^{4} w_l \bullet v_l$.

3. The correct guess of the key k^* corresponds to $\arg\max_k \mathrm{VPA}(k)$.

3.2.3 Experimental Results

The VPA is carried out on a masked DES implementation. It is tested on $200,000$ with different weights $(w_0, w_1, w_2, w_3, w_4)$ values. The weights of the VPA indicator producing the best results were $(\frac{1}{4}, 1, 0, -1, -\frac{1}{4})$. This result was validated empirically, *i.e.* we tested several weight vectors until we found the good combination.

Figure 3.5 shows the 10 keys having the higher VPA indicator values for each DES S-box. These indicators have been normalized such that the best key candidate has an indicator value equal to 1. Then, for each S-box, the round subkey guessed by our VPA algorithm is the key corresponding to the highest indicator value (the most left one on the figures). The eight DES S-boxes subkeys used during the first round of our DES implementation have been guessed by the VPA.

Note that the indicator illustrated by these results does not only use the sets producing maximum and minimum variance observations ($\mathsf{HW}(Z) = 0$ and $\mathsf{HW}(Z) = 4$, see Fig. 3.1). The use of such sets decreased the overall performance of the attack. The fact that there are four times less traces for $\mathsf{HW}(Z) = 0$ and $\mathsf{HW}(Z) = 4$ than for $\mathsf{HW}(Z) = 1$ and $\mathsf{HW}(Z) = 3$ could explain this behaviour.

3.3 Link between VPA and Second-Order CPA

In [DPRS11], the authors demonstrated that even the most commonly used univariate attacks (DPA, PPA [LCC+06], CPA, ...) do not share the same efficiency, they can be reformulated one in the function of the other. Especially, most univariate attacks can be rewritten as a correlation coefficient computation with different leakage models. In this section, we try to find the link between the VPA and the univariate second-order CPA attack. The results in this section have been founded in collaboration with Emmanuel Prouff.

The VPA distinguisher, is defined with respect to a family of coefficients $(\omega_i)_{i\in[0,4]}$ such that:

$$\begin{aligned}\mathrm{VPA}(k) &= \sum_i \omega_i \mathsf{Var}[L \mid \mathsf{HW}(Z) = i] \\ &= \sum_i \frac{\omega_i}{\mathsf{P}[M_h = i]} \mathsf{P}[M_h = i] \mathsf{Var}[L \mid M_h = i] \ ,\end{aligned}$$

where M_h is the Hamming weight leakage model (*i.e.* $\mathsf{HW}(Z)$). Now, if we denote by g the function defined over the set definition of M_h by $g(i) = \frac{\omega_i}{\mathsf{P}[M_h=i]}$, we eventually get:

$$\begin{aligned}\mathrm{VPA}(k) &= \sum_i g(i)\mathsf{P}[M_h = i]\mathsf{Var}[L \mid M_h = i] \\ &= \sum_{\alpha\in\mathrm{Im}(g)} \alpha\mathsf{P}[M_h \in g^{-1}(\alpha)]\mathsf{Var}[L \mid M_h \in g^{-1}(\alpha)] \\ &= \sum_{\alpha\in\mathrm{Im}(g)} \alpha\mathsf{P}[g(M_h) = \alpha]\mathsf{Var}[L \mid g(M_h) = \alpha] \\ &= \sum_{\alpha\in\mathrm{Im}(g)} \alpha\mathsf{P}[g(M_h) = \alpha]\mathbb{E}[(L - \mathbb{E}[L])^2 \mid g(M_h) = \alpha] \\ &= \sum_{\alpha\in\mathrm{Im}(g)} \mathsf{P}[g(M_h) = \alpha]\mathbb{E}[(L - \mathbb{E}[L])^2\alpha \mid g(M_h) = \alpha] \ .\end{aligned}$$

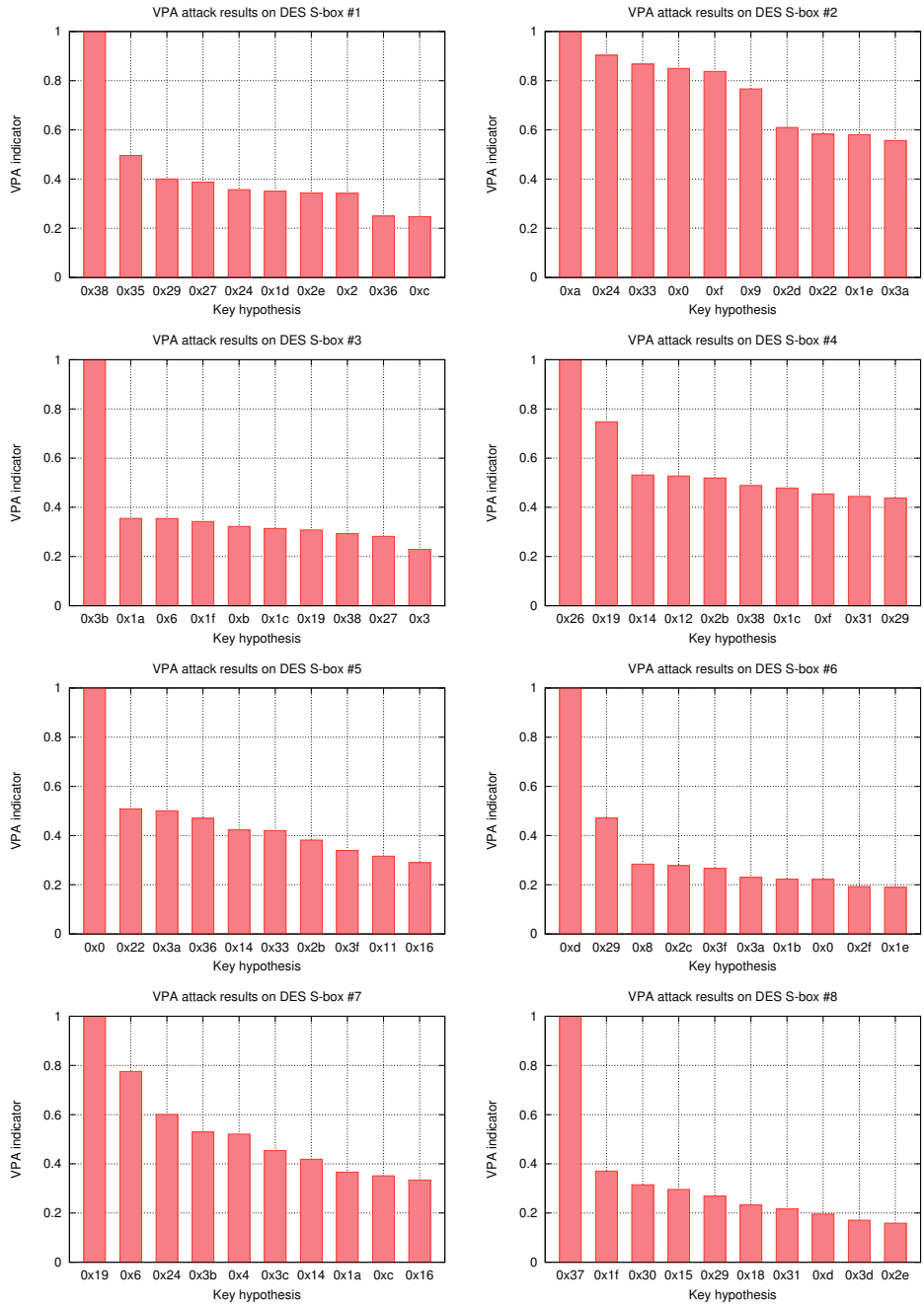


Figure 3.5: VPA results on 200,000 power consumption traces of a masked DES implementation.

Hence, the VPA attack can be written as:

$$\mathrm{VPA}(k) = \mathbb{E}[(L - \mathbb{E}[L])^2 g(M_h)] \ . \tag{3.1}$$

On the other hand, the second-order CPA attack can be expressed as:

$$\rho((L - \mathbb{E}[L])^2, g(M_h)) = \frac{\mathbb{E}[(L - \mathbb{E}[L])^2 g(M_h)] - \mathbb{E}[(L - \mathbb{E}[L])^2]\mathbb{E}[g(M_h)]}{\sqrt{\mathsf{Var}[(L - \mathbb{E}[L])^2]}\sqrt{\mathsf{Var}[g(M_h)]}} \ .$$

In most of (if not all) cryptographic contexts, the terms $\mathsf{Var}[g(M_h)]$ and $\mathbb{E}[g(M_h)]$ are not key-hypothesis dependent (*i.e.* do not depend on k) and are therefore constant with respect to k. Hence, the coefficient $\rho((L - \mathbb{E}[L])^2, g(M_h))$ viewed as a function of k can be written:

$$\rho((L - \mathbb{E}[L])^2, g(M_h)) = a\mathbb{E}[(L - \mathbb{E}[L])^2 g(M_h)] + b \ , \tag{3.2}$$

where a and b are some constant terms (*w.r.t* k). We eventually conclude on the following relation:

$$\mathrm{VPA}(k) = \frac{1}{a}\left(\rho\left((L - \mathbb{E}[L])^2, g(M_h)\right) - b\right) \ . \tag{3.3}$$

The relation above says that a VPA is equivalent to a second-order correlation power attack thanks to a change of model g. It moreover implies that the choice of the coefficients ω_i that optimizes the efficiency of the VPA is that corresponding to the function g such that the correlation coefficient $\rho\left((L - \mathbb{E}[L])^2, g(M_h)\right)$ is maximal, or in other terms, *s.t.* $g(M_h)$ is the best affine approximation of the random variable $(L - \mathbb{E}[L])^2$. Methods exist to design such a function g.

Example 1. *Let us assume that X is a 4-bit variable with uniform distribution. Let us assume that $\omega_0 = 1/4$, $\omega_1 = 1$, $\omega_2 = 0$, $\omega_3 = -1$ and $\omega_4 = -1/4$ and that Z has a uniform distribution. In this case the definition set of M_h is $\{0, 1, 2, 3, 4\}$ and g is the function defined such that:*

$$\begin{array}{rcl} g(0) & = & \frac{1}{4} \\ g(1) & = & 4 \\ g(2) & = & 0 \\ g(3) & = & -4 \\ g(4) & = & -\frac{1}{4} \end{array} \ .$$

Since M_h equals by definition $\mathsf{HW}(Z)$, we deduce that $g(M_h) = g \circ \mathsf{HW}(Z)$ can be viewed as a new model M'_h defined such that:

$$M'_h = \begin{cases} \frac{1}{4} & if \mathsf{HW}(Z) = 0 \\ 4 & if \mathsf{HW}(Z) = 1 \\ 0 & if \mathsf{HW}(Z) = 2 \\ -4 & if \mathsf{HW}(Z) = 3 \\ -\frac{1}{4} & if \mathsf{HW}(Z) = 4 \end{cases} \ .$$

Our previous analysis here implies that the VPA distinguisher is affinely equivalent with $\rho((L - \mathbb{E}[L])^2, M'_h)$ for the model function defined above.

3.4 Conclusions

In this chapter we proposed a second-order attack based on variance analysis which is powerful enough to attack a masked implementation in an FPGA using a reasonable number of traces. VPA attack can be expressed as a second-order CPA by changing the leakage model. Both distinguishers have the same soundness if the leakage model is perfectly known. In the next chapter, we discuss another interesting distinguisher: the mutual information which is a *generic* one since it detects any kind of statistical dependency between the leakage measurements and the secret key.

Chapter 4

Entropy-based Power Analysis

In this chapter, we study the mutual information analysis (MIA) in the context of both unprotected and protected implementations. Then, we propose a new approach called the *Entropy-based Power Analysis* (EPA) using a weighted sum of conditional entropies as a distinguisher. It is designed to promote partitions of high informative content and to ease the distinguishability between key candidates. It is carried out on a masked DES coprocessor which is part of a SoC programmed in an FPGA. EPA attack is compared with the MIA and VPA attacks.

The results presented in this chapter have been published in collaboration with Sylvain Guilley and Jean-Luc Danger in the international symposium on *Hardware-Oriented Security and Trust (HOST 2010)* [MGDF10].

Contents

4.1 Probability Density Function Estimation

In order to compute the mutual information, one has to estimate the probability density functions for which several solutions exist. For instance, the probability density function can be estimated non-parametrically, by using histograms or kernel density estimation, or parametrically by using the Expectation Maximization (EM) algorithm. In the context of SCA attacks, the authors of the original MIA attack [GBTP08] used the histogram method for density estimation. In [PR09], Emmanuel Prouff and Matthieu Rivain emphasize the use of parametric PDF estimation tools since they are more relevant to side channel key recovery attacks.

Table 4.1: Some kernel functions for PDF estimation.

Kernel name	Function $\Theta(t)$	Optimal bin width h
Uniform	$\frac{1}{2}I(t)$	$\sigma(\frac{12\sqrt{\pi}}{n})^{1/5}$
Triangle	$(1-\lvert t\rvert)I(t)$	$\sigma(\frac{64\sqrt{\pi}}{n})^{1/5}$
Epanechnikov	$\frac{3}{4}(1-t^2)I(t)$	$\sigma(\frac{40\sqrt{\pi}}{n})^{1/5}$
Triweight	$\frac{35}{32}(1-t^2)^3I(t)$	$\sigma(\frac{25200\sqrt{143\pi}}{n})^{1/5}$
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}t^2)$	$\sigma(\frac{4}{3n})^{1/5}$

The main disadvantages of using advanced estimation tools are the memory requirements and the higher computation load. Hereafter, we detail these widely used PDF estimation tools.

4.1.0.1 Histogram Method

A very common method to estimate a PDF is to calculate a histogram. First, we split the data into fixed size bins. Second, we compute the frequency of each bin to find the probability. For relatively simple distribution, reasonable choices of the *bin width* h can be done using Scott's rule ($h = 3.49 \times \sigma \times n^{-1/3}$, where σ is the empirical data standard deviation and n is the number of bins) and Freedman-Diaconis' rule [Wan96].

4.1.0.2 Kernel Density Estimation

The probability density function is estimated as:

$$f(x) = \frac{1}{nh}\sum_{i=0}^{n}\Theta\left(\frac{x-x_i}{h}\right),$$

where the function Θ is the kernel function. Some commonly used kernel functions are listed in Tab. 4.1 (all refer to [Sil86]), where I is a step function defined as $I(t) = 1$ if $|t| \leq 1$, 0 otherwise.

4.1.0.3 Parametric Estimation

If we consider the observations x_i to be a mixture of Gaussians, the parametric method models the probability density function as:

$$f(x) = \sum_{i=0}^{n-1} w_i \cdot N(x, \mu_i, \sigma_i),$$

where the w_i, μ_i and σ_i are respectively the weight, the mean and the standard deviation of each component. An efficient algorithm called the Expectation Maximization algorithm [DLR77] allows one to give a good approximation of a probability density function in the form of a finite mixture.

Table 4.2: Conditional entropy estimation.

Conditional entropy	**Good key**	**Bad key**
Histogram method	$-9.16542 \pm 2 \times 10^{-5}$	$-9.16367 \pm 2 \times 10^{-5}$

4.2 Practical MIA Attack

To test the MIA in a real-life context, we performed it against two DES hardware implementations. The first one is an unprotected DES and the second one is the same masked DES implementation used to test the VPA attack (see Chapter 3 Sec. 3.2.1).

4.2.1 MIA Attack on Unprotected DES

Before performing the MIA attack, we tried to estimate the conditional entropy of the first DES S-box when $\mathsf{HW}(Z) = 0$. We use two hypotheses of the key, the first is right and the second is false. We carried out the conditional entropy estimation using real power consumption measurements of our circuit. We summarize the result in Tab. 4.2 and we validated that the estimated conditional entropy is minimum for the good key, so the MIA attack can be carried out on the unprotected DES. We have estimated the accuracy of 2×10^{-5} bit as the quadratic error with respect to the theoretical value $\log_2(\sigma\sqrt{2\pi e})$, where σ is the empirical standard deviation.

Assuming a Hamming weight leakage model[1], the MIA attack is described in the following procedure:

1. Encrypt n plaintexts (x_i, $i \in [1, n]$) and collect n observations of power consumption (traces L_i).

2. Compute the entropy of the observations $\mathsf{H}[L]$.

3. For each S-box, make assumptions about the key $k \in [0, 63]$:

   - Sort the traces L_i to get five activity partitions set_l, $l \in [0, 4]$, corresponding to the five possible values $\mathsf{HW}(Z) = \mathsf{HW}(S(x_i \oplus k))$, where S is the DES S-box function.
   - Compute the conditional entropy $\mathsf{H}[L \mid \mathsf{HW}(Z) = l]$ for each set_l.
   - Compute the mutual information $\mathrm{MIA}(k)$, as the difference between the observations entropy and the sum of the conditional entropy weighted

[1]Since the distribution of the leakage in unprotected implementation only depends on $\mathsf{HW}(Z)$, therefore $\mathsf{I}[L; \mathsf{HW}(Z)] = \mathsf{I}[L; Z]$. One can choose the identity leakage model (*i.e.* Z) to perform the MIA attack and obtain the same results. The same argument holds for masking countermeasure at any order.

with the probability $p_l = \mathsf{P}[\mathsf{HW}(Z) = l]$:

$$\mathrm{MIA}(k) = \mathsf{H}[L] - \sum_{l=0}^{4} p_l \times \mathsf{H}[L \mid \mathsf{HW}(Z) = l] \ .$$

4. The correct guess of the key k^* corresponds to $\arg\max_k \mathrm{MIA}(k)$.

We performed the MIA attack using the histogram estimation method. For the choice of bins, we follow the rule used in the original MIA attack described in [GBTP08, BGP+11]: the number of bins is equal to the number of the distinct leakage model values (*i.e.* 5 in our case).

The MIA is tested on 50,000 traces of an unprotected DES implementation. Figure 4.3 shows the mutual information values according to each subkey predicted for the first DES S-box. Consequently, the round subkey guessed by the MIA attack is the key corresponding to the highest mutual information.

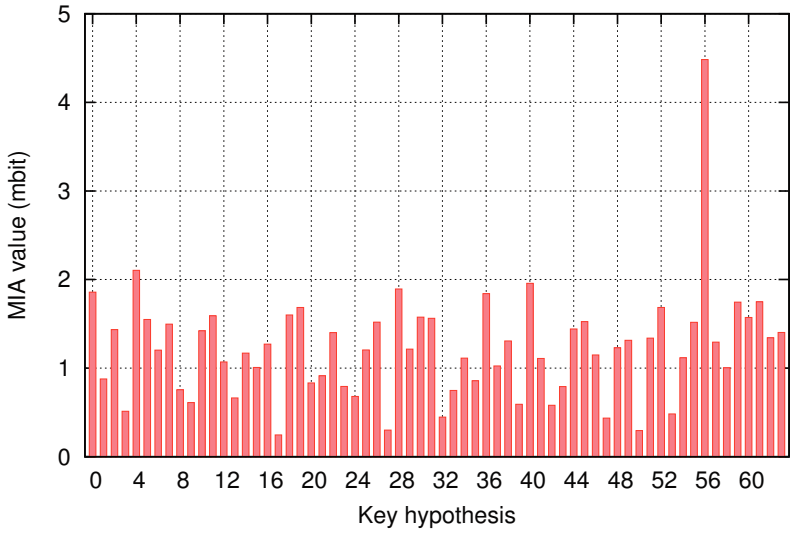


Figure 4.1: MIA results on 50,000 power consumption traces of an unprotected DES implementation. The correct key is $k = 56$.

The eight subkeys used during the first round of our DES implementation have been guessed by the MIA attack. The number of traces to break the first round subkeys is given in Tab. 4.3. In the next subsection we try to answer the question: Is the masked DES sensitive to the MIA attack?

4.2.2 MIA Attack on Masked DES

In software implementations, the masked data $Z \oplus M$ and the mask M are manipulated sequentially. Then, the leakage is a multivariate signal (*i.e.* $L = (L_1, L_2)$,

Table 4.3: Minimum traces to disclose (MTD) the subkey for each S-box of the unprotected DES module on the first round.

S-box #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
MTD	12,133	10,827	12,974	12,317	11,034	13,578	10,651	11,635

where L_1, L_2 denote respectively the masked variable and the mask leakage) and combined attacks can be carried out. Of special interest is the multivariate MIA (MMIA) introduced recently by Gierlich *et al.* in [GBPV10] and deeply analysed in [BGP+11]. It uses $\mathsf{I}[(L_1, L_2); \mathsf{HW}(Z)]$ as a distinguisher. However, in a hardware implementation (our study), $Z \oplus M$ and M are used simultaneously. In this case, the MMIA cannot be applied since the leakage is univariate. In the sequel, we consider the univariate MIA attack.

Table 4.4 summarizes the theoretical values of the conditional entropy of each values of $\mathsf{HW}(Z)$, when considering 4-bit variables, in the two cases good and bad key. The entropy, in the case of a bad key, is equal to: 2.5442 bit. Hence a contrast in mutual information of:

$$\begin{aligned} \mathsf{I}[L; \mathsf{HW}(Z^*)] - \mathsf{I}[L; \mathsf{HW}(Z)] &= \cancel{\mathsf{H}[L]} - \mathsf{H}[L \mid \mathsf{HW}(Z^*)] - \cancel{\mathsf{H}[L]} + \mathsf{H}[L \mid \mathsf{HW}(Z)] \\ &= -1.3922 + 2.5442 \\ &= 1.1520\text{bit} \ , \end{aligned}$$

where Z^* denotes the sensitive variable using the good key k^*. So, theoretically, with ideal S-boxes, the univariate MIA attack can succeed on a first-order masking implementation: it does distinguish the correct key guess from the wrong ones. However, it clearly appears that the partitions do not contribute equally to disambiguate the correct key from the incorrect ones: the smaller the value $\mathsf{H}[L \mid \mathsf{HW}(Z)]$ is, the better the entropy difference is. This noting is the first motivation to devise an improved version of the MIA.

Additionally, we simulated the computation of the conditional entropy $\mathsf{H}[L \mid \mathsf{HW}(Z)]$ for all key hypotheses. Figure 4.2 describes the result for the 4^{th} DES S-box using good key equal to 38. We observe that the conditional entropy $\mathsf{H}[L \mid \mathsf{HW}(Z)]$ for some keys is under the theoretic value 2.5442 bit. We explain this result by the fact that the activity of the S-box itself leaks information and decreases the entropy. This phenomenon had already been observed in [BCO04] and referred to as "ghost peaks", characterized in [GHP04]. It is a second compelling reason to define an upgraded version of the MIA. So, in practice, it is harder than expected to discriminate the right key in the context of masked implementations.

We reproduce the MIA attack described in Sec. 4.2.1 on the masked DES implementation; the attack failed even with up to 200,000 power consumption traces. This result is in line with that of Standaert *et al.* in [SVCO+10, §7]. More pre-

Table 4.4: Theoretical conditional entropy of the masked DES.

Theoretical entropies	**The correct key**	**Any wrong key**
$\mathsf{H}[L \mid \mathsf{HW}(Z) = 0]$	2.0306 bit	2.5442 bit
$\mathsf{H}[L \mid \mathsf{HW}(Z) = 1]$	1.8113 bit	2.5442 bit
$\mathsf{H}[L \mid \mathsf{HW}(Z) = 2]$	1.5000 bit	2.5442 bit
$\mathsf{H}[L \mid \mathsf{HW}(Z) = 3]$	1.0000 bit	2.5442 bit
$\mathsf{H}[L \mid \mathsf{HW}(Z) = 4]$	0.0000 bit	2.5442 bit
$\mathsf{H}[L \mid \mathsf{HW}(Z)]$	**1.3992 bit**	**2.5442 bit**

cisely, the univariate MIA attack (MIA with the sum combining function) is strongly affected by the algorithmic noise addition.

4.3 Evaluation of the EPA Attack

4.3.1 Proposed EPA Attack

Table 4.5: Conditional entropy estimation of the masked DES.

Conditional entropy	**Good key**
$\mathsf{HW}(Z) = 0$	23.2842549755
$\mathsf{HW}(Z) = 1$	23.2460651542
$\mathsf{HW}(Z) = 2$	23.2185655678
$\mathsf{HW}(Z) = 3$	23.189564065
$\mathsf{HW}(Z) = 4$	23.1286079923

Table 4.5 shows that the distribution of power consumption has different conditional entropy. For instance the conditional entropy difference between the PDF for $\mathsf{HW}(Z) = 0$ and $\mathsf{HW}(Z) = 4$ (see Tab. 4.5) should be enough discriminating, since it is maximum when using the good key. This leads us to define accordingly the EPA attack, which is an improved partition distinguisher. The EPA is a combination of conditional entropies of the power consumption traces computed during the first DES round, while ciphering random messages. The *EPA* algorithm is made explicit below:

1. Encrypt n plaintexts (x_i, $i \in [1, n]$) and collect n observations of power consumption (traces L_i).

2. For each S-box, make assumptions about the key $k \in [0, 63]$:

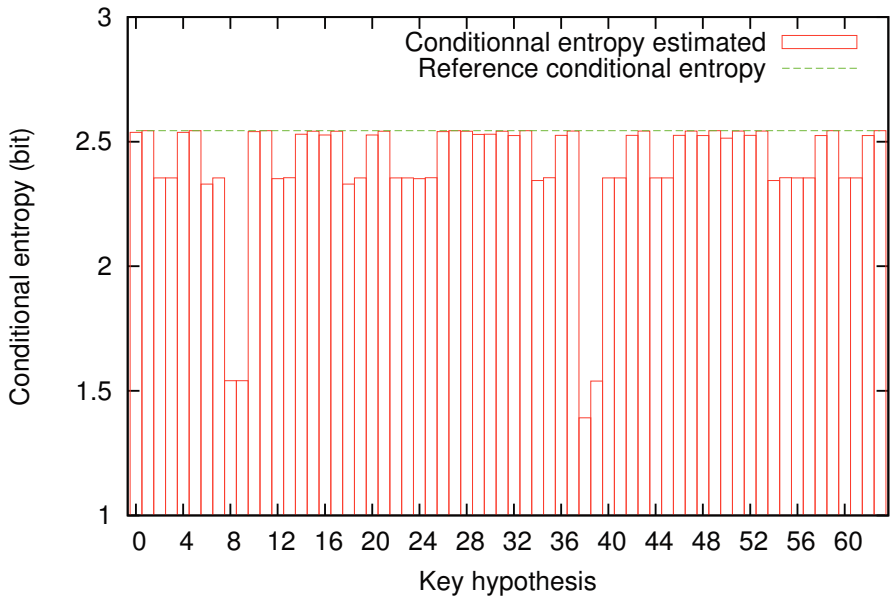


Figure 4.2: Comparison between the reference and the estimated conditional entropy for each key in S-box #4. The correct key is $k = 38$.

- Sort the traces L_i to get five activity partitions set_l, $l \in [0, 4]$, corresponding to the five $\mathsf{HW}(Z) = l$ possible values.
- Compute the conditional entropy $\mathsf{H}[L \mid \mathsf{HW}(Z) = l]$ for each set set_l.
- Compute an EPA indicator $\mathrm{EPA}(k)$ being a linear combination of the conditional entropy with weights w_l:

$$\mathrm{EPA}(k) = \sum_{l=0}^{4} w_l \times \mathsf{H}[L \mid \mathsf{HW}(L) = l] \ .$$

3. The correct guess of the key k^* corresponds to $\arg\max_k \mathrm{EPA}(k)$.

4.3.2 Experimental Results

The *EPA* is carried out on a masked DES implementation. It is tested on $200,000$ power measurements with different weights $(w_0, w_1, w_2, w_3, w_4)$ values. The weights of the EPA indicator producing the best results are $(0.25, 1, 0, -1, -0.25)$, the same empirical result as for the VPA attack.

Figure 4.3 shows the EPA indicator values according to each key hypothesis for the first DES S-box. Then, for each S-box, the round subkey guessed by the EPA attack is the key corresponding to the highest indicator value. The eight DES S-boxes subkeys used during the first round of our masked DES have been guessed by the EPA attack.

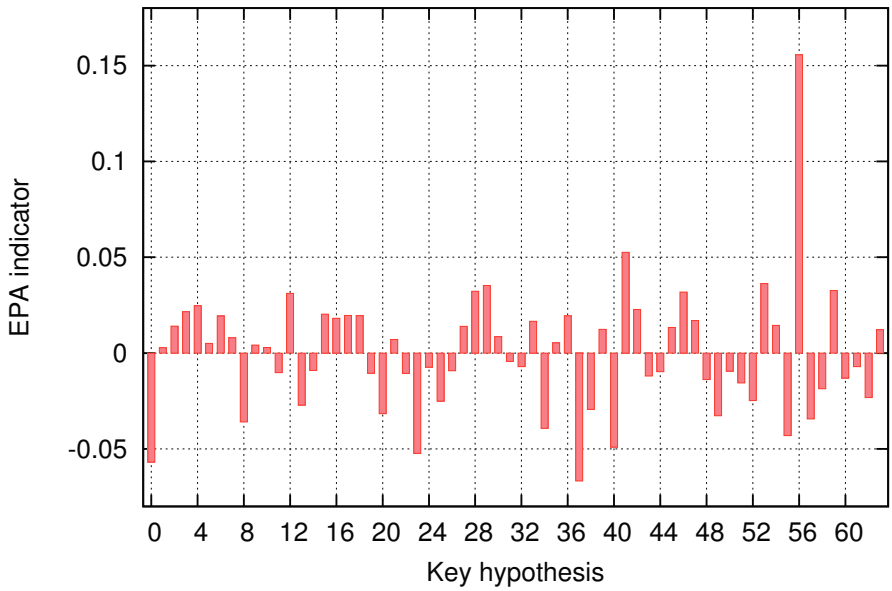


Figure 4.3: EPA results on 200,000 power consumption traces of a masked DES implementation. The correct key is $k = 56$.

4.3.3 EPA *Vs* VPA *Vs* MIA

In this subsection, we compare the EPA attack to the MIA [GBTP08] and VPA attacks. Following the recent advances concerning the comparison of univariate side-channel distinguishers [SGV08], we apply the first-order success rate to assess the performance of the three attacks. Figure 4.4 shows our experimental results for those three attacks on the masked DES implementation in FPGA.

We can see that VPA performs well in this scenario. About 5,000 traces suffice to achieve a success rate of 50% and starting from about 14,000 traces the VPA attack reveals the correct key with success rate of 95%. The EPA attack performs well also. The success rate stays well above 50% even when using 11,000 measurements, but eventually reaches success rate of 95% using 18,000 traces. The MIA attack performs much worse. The success rate stays under 10% even when using 25,000 measurements. We conclude that the distinguisher based on the computation of the difference between entropy is more efficient than the MIA attack in the context of attacking a masked implementation. This is expected since the algorithmic noise level is high. The VPA remains the best attack, since the leakage model is well known.

This result is in line with that in [PR09, SVCO+10]: CPA-like attacks enable a better discrimination of the correct key than the MIA attack even if masking is used to ensure the protection. However, in the context of an unknown model (*e.g.* algorithms protected with logic style that do not exhibit a simple Hamming weight

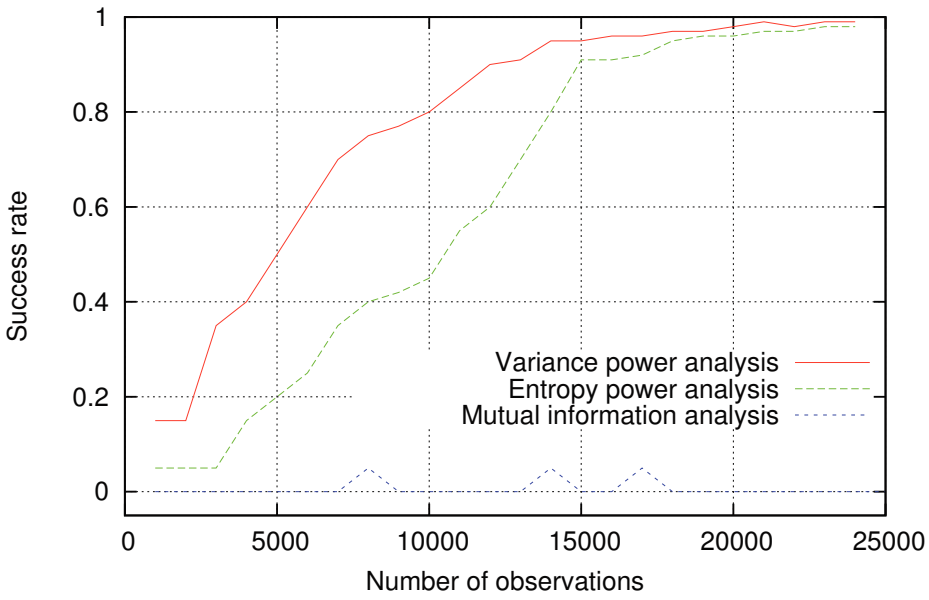


Figure 4.4: First-order success rate of 3 distinguishers on a masked DES implementation in FPGA.

leakage model), an information-theoretic attack would be necessary; the EPA would be more appropriate, since it would outperform the MIA.

4.4 Conclusions

In this chapter we showed the limitations of the mutual information attack when masking is used to protect the implementation. We presented a new attack based on entropy analysis, which succeeds in breaking a hardware masked DES implementation. This attack is quite efficient (all the S-boxes are cracked) and requires a reasonable number of traces (15K).

In the next chapter, we propose another information-theoretic distinguisher which compares also favorably to the MIA attack.

Chapter 5

Inter-class Information Analysis

In this chapter, we suggest a new information-theoretic distinguisher, termed *Inter-class Information Analysis* (IIA). Conversely to MIA or KSA, it consists in comparing the conditional leakages between themselves, pairwise. First, we theoretically compare IIA and MIA, including a proof of soundness in the case of arbitrary noise and a theoretical analysis about the ability to discriminate. Second, we confirm our results using simulations of a protected PRESENT implementation. Then, we extend the notion of *inter-class* to KSA attack and we propose finally a simulation-based fair framework to evaluate and compare several distinguishers.

Parts of the results presented in this chapter have been published in collaboration with Sylvain Guilley, Olivier Rioul and Jean-Luc Danger in the *International Conference on Information and Communications Security (ICICS 2012)* [MRGD12].

Contents

5.1 Inter-class Distinguishers

In this section we first highlight methods to generally compare two probability distributions and, second, describe strategies in order to compare conditional probability distributions, which are required in order to decide which key is the most likely one. Note that, the following statements consider discrete RVs, however, they can be easily adapted for continuous RVs. The term PMF is used for the *Probability Mass Function* of discrete RVs.

5.1.1 On Comparing Conditional Probability Distributions

Definition 1. *Let us consider two PMFs* $\mathsf{P}[X]$ *and* $\mathsf{Q}[X]$ *of a discrete RV* X*, then the Kullback-Leibler divergence is defined as:*

$$D_{\mathsf{KL}}[P \parallel Q] = \sum_x \mathsf{P}[X = x] \cdot \log \frac{\mathsf{P}[X = x]}{\mathsf{Q}[X = x]} \ .$$

Other examples for divergences D are total variation $D(P, Q) = \sum \mid P - Q \mid$ for discrete probabilities, Hellinger distance $D(P, Q) = \int \mid P - Q \mid^2$ for densities, Renyi α-divergence, Wasserstein distance, Kolmogorov-Smirnov distance (L^∞-norm between cumulative distribution functions), *etc.*

In the area of side channel analysis one is not only interested in the PMF of one RV but the conditional PMF between two RVs, *e.g.* the probability of measuring a specific current consumption while processing a certain value. Let us therefore consider the two RVs L and Z representing respectively the leakage measurements and the sensitive variable.

5.1.2 Conditional-to-Unconditional

As introduced in [GBTP08], one option for a side channel distinguisher $\mathcal{D}_0$ is to compare the average Kullback-Leibler distance D_{KL} between the conditional probability $\mathsf{P}[L \mid Z = z]$ for each value z and its average $\mathsf{P}[L]$:

$$\mathcal{D}_0 = \mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L] \parallel \mathsf{P}[L \mid Z = z]]\big) = \sum_z \mathsf{P}[Z = z] D_{\mathsf{KL}}[\mathsf{P}[L] \parallel \mathsf{P}[L \mid Z = z]] \ .$$

Accordingly, this approach compares the conditional probabilities $\mathsf{P}[L \mid Z = z]$ with the unconditional probability $\mathsf{P}[L]$ (see the left of Fig. 5.1).

Remark 1. *It is straightforward to prove that the distinguisher* $\mathcal{D}_0 = \mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L] \parallel \mathsf{P}[L \mid Z]]$ *results in MIA, more precisely:*

$$\mathsf{I}[L; Z] = \mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L] \parallel \mathsf{P}[L \mid Z]]\big) \ .$$

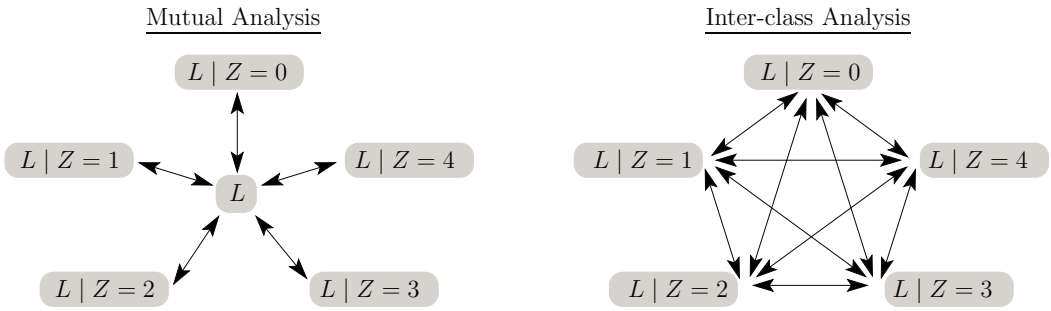


Figure 5.1: Reflecting different strategies to compare classes of conditional probability distributions (the distance between A and B is depicted with an arrow).

5.1.3 Conditional-to-Conditional

Instead of taking the average $\mathbb{E}(\mathsf{P}[L \mid Z = z]) = \mathsf{P}[L]$ into account, another possibility is to directly consider the distance between two classes z, z':

$$D_{\mathsf{KL}}[\mathsf{P}[L \mid Z = z] \parallel \mathsf{P}[L \mid Z = z']] \ .$$

Accordingly, the distinguisher $\mathcal{D}_I$ would be defined as the average distance between all possible tuples of classes (z, z') with $z \neq z'$:

$$\begin{aligned} \mathcal{D}_I &= \frac{1}{2}\,\mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L \mid Z = z] \parallel \mathsf{P}[L \mid Z = z']]\big) \\ &= \frac{1}{2}\sum_{z,z'} \mathsf{P}[Z = z]\mathsf{P}[Z = z']D_{\mathsf{KL}}[\mathsf{P}[L \mid Z = z] \parallel \mathsf{P}[L \mid Z = z']] \ . \end{aligned}$$

Note that, due to symmetry each term for $z \neq z'$ in the sum is counted twice, hence we add the factor $1/2$. Thus, in contrast to MIA, this strategy only determines the average distance between two conditional probabilities, without the comparison to the unconditional distribution of $\mathsf{P}[L]$ which is not a key-hypothesis dependant (see the right of Fig. 5.1).

Remark 2. *Let X be a RV, then the divergence:*

$$D_{\mathsf{KL}}[\mathsf{P}[X = x] \parallel \mathsf{P}[X = x']] + D_{\mathsf{KL}}[\mathsf{P}[X = x'] \parallel \mathsf{P}[X = x]] \ ,$$

for different classes x, x' is named as inter-class divergence (see for instance [SP00]).

In our case, we keep this inter-class notion and define the inter-class information distinguisher as:

Definition 2. *Let L, Z be (discrete) random variables with PMF $\mathsf{P}[L = l, Z = z]$ and marginal PMFs $\mathsf{P}[L = l], \mathsf{P}[Z = z]$, then the inter-class information analysis is*

defined as:

$$\begin{aligned} \mathsf{II}[L;Z] &= \frac{1}{2}\,\mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L \mid Z=z] \parallel \mathsf{P}[L \mid Z=z']]\big) \\ &= \frac{1}{2}\sum_{z\neq z'} \mathsf{P}[Z=z]\mathsf{P}[Z=z']D_{\mathsf{KL}}[\mathsf{P}[L \mid Z=z] \parallel \mathsf{P}[L \mid Z=z']] \\ &= \sum_{z<z'} \mathsf{P}[Z=z]\mathsf{P}[Z=z']D_{\mathsf{KL}}[\mathsf{P}[L \mid Z=z] \parallel \mathsf{P}[L \mid Z=z']] \ , \end{aligned} \tag{5.1}$$

where the expectation is taken on the probability distribution of Z.

5.1.4 More than One Definition

In the following we provide the reader with equivalent definitions of $\mathsf{II}[L;Z]$ since they are required later on. It is well known that mutual information is symmetric in the sense that $\mathsf{I}[L;Z] = \mathsf{I}[Z;L]$. Yet, Kullback divergence is not as $D_{\mathsf{KL}}[L \parallel Z] \neq D_{\mathsf{KL}}[Z \parallel L]$ in general. For this reason Kullback (and Hajek) introduced a symmetric divergence:

Definition 3. *Let P, Q be two probability distributions, then the symmetric Kullback-Leibler divergence is defined as:*

$$\delta(P \parallel Q) = \frac{D_{\mathsf{KL}}[P \parallel Q] + D_{\mathsf{KL}}[Q \parallel P]}{2} = \frac{1}{2}\sum (P-Q)\log\frac{P}{Q} \ .$$

So, the IIA metric can also be expressed as:

Definition 4. *Let L, Z be RVs with probability distribution $\mathsf{P}[L=l], \mathsf{P}[Z=z]$, and $\mathsf{P}[L=l, Z=z]$ then the inter-class information is defined as:*

$$\begin{aligned} \mathsf{II}[L;Z] &= \mathbb{E}\big(\delta(\mathsf{P}[L \mid Z] \parallel \mathsf{P}[L])\big) \\ &= \frac{\mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L \mid Z] \parallel \mathsf{P}[L]]\big) + \mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L] \parallel \mathsf{P}[L \mid Z]]\big)}{2} \qquad (5.2) \\ &= \frac{\sum_{l,z}(\mathsf{P}[L=l, Z=z] - \mathsf{P}[L=l]\mathsf{P}[Z=z])\log\frac{\mathsf{P}[L=l,Z=z]}{\mathsf{P}[L=l]\mathsf{P}[Z=z]}}{2} \ , \qquad (5.3) \end{aligned}$$

where the expectation is taken over Z.

Moreover, we give an entropic definition of $\mathsf{II}[L;Z]$. Let the *conditional cross-entropy* $\mathsf{H}'[L;Z]$ be defined as:

$$\mathsf{H}'[L;Z] = -\sum_{l}\sum_{z} \mathsf{P}[L=l]\mathsf{P}[Z=z]\log \mathsf{P}[L=l \mid Z=z] \ . \tag{5.4}$$

Then, $\mathsf{II}[L;Z]$ can also be expressed as:

Definition 5.

$$\mathsf{II}[L;Z] = \frac{\mathsf{H}'[L;Z] - \mathsf{H}[L \mid Z]}{2} \ . \tag{5.5}$$

Theorem 1. $\mathsf{II}[L;Z]$ *in Definitions 2, 4, and 5 coincide.*

Proof. **Equivalence of Definition 2 and 4:**
We start from Eqn. (5.2):

$$
\begin{aligned}
&= \frac{\mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L=l \mid Z] \parallel \mathsf{P}[L=l]]\big) + \mathbb{E}\big(D_{\mathsf{KL}}[\mathsf{P}[L=l] \parallel \mathsf{P}[L=l \mid Z]]\big)}{2} \\
&= \frac{1}{2}\sum_{l}\sum_{z}\mathsf{P}[Z=z]\mathsf{P}[L=l \mid Z=z]\log\frac{\mathsf{P}[L=l \mid Z=z]}{\mathsf{P}[L=l]} \\
&\quad + \frac{1}{2}\sum_{l}\sum_{z'}\mathsf{P}[Z=z']\mathsf{P}[L=l]\log\frac{\mathsf{P}[L=l]}{\mathsf{P}[L=l \mid Z=z']} \\
&= \frac{1}{2}\sum_{l}\sum_{z}\sum_{z'}\mathsf{P}[Z=z]\mathsf{P}[Z=z']\mathsf{P}[L=l \mid Z=z]\log\frac{\mathsf{P}[L=l \mid Z=z]}{\mathsf{P}[L=l]} \\
&\quad + \frac{1}{2}\sum_{l}\sum_{z}\sum_{z'}\mathsf{P}[Z=z']\mathsf{P}[Z=z]\mathsf{P}[L=l \mid Z=z]\log\frac{\mathsf{P}[L=l]}{\mathsf{P}[L=l \mid Z=z']} \\
&= \frac{1}{2}\sum_{z}\sum_{z'}\mathsf{P}[Z=z]\mathsf{P}[Z=z']\sum_{l}\mathsf{P}[L=l \mid Z=z]\log\frac{\mathsf{P}[L=l \mid Z=z]}{\mathsf{P}[L=l \mid Z=z']} \\
&= \frac{1}{2}\sum_{z\neq z'}\mathsf{P}[Z=z]\mathsf{P}[Z=z']D_{\mathsf{KL}}[\mathsf{P}[L \mid Z=z] \parallel \mathsf{P}[L \mid Z=z']] \ .
\end{aligned}
$$

Equivalence of Definition 4 and 5:
We start from Eqn. (5.3). Since we can remove $\mathsf{P}[L=l]$ inside the logarithm as $\sum_z \mathsf{P}[L=l,Z=z] - \mathsf{P}[L=l]\mathsf{P}[Z=z] = 0$ and furthermore $\mathsf{P}[L=l,Z=z]/\mathsf{P}[Z=z] = \mathsf{P}[L=l \mid Z=z]$, we reformulate:

$$
\begin{aligned}
&\frac{1}{2}\sum_{l,z}(\mathsf{P}[L=l,Z=z] - \mathsf{P}[L=l]\mathsf{P}[Z=z])\log\frac{\mathsf{P}[L=l,Z=z]}{\mathsf{P}[L=l]\mathsf{P}[Z=z]} \\
= \ &\frac{1}{2}\sum_{l,z}(\mathsf{P}[L=l,Z=z] - \mathsf{P}[L=l]\mathsf{P}[Z=z])\log\mathsf{P}[L=l \mid Z=z] \\
= \ &\frac{\mathsf{H}'[L;Z] - \mathsf{H}[L \mid Z]}{2} \ .
\end{aligned}
$$

□

The following theorem highlights properties for $\mathsf{II}[L;Z]$ that are well-known for $\mathsf{I}[L;Z]$ and a relation between both distinguishers.

Theorem 2. *Let L, Z be two RVs, then:*

1. *(Symmetry):* $\mathsf{II}[L;Z] = \mathsf{II}[Z;L]$.
2. *(Dependence):* $\mathsf{II}[L;Z] = 0 \iff$ *L, Z are independent.*
3. *(Relation to Mutual Information):* $2\mathsf{II}[L;Z] \geq \mathsf{I}[L;Z]$.

Proof. **Symmetry:** Since Eqn. (5.3) is symmetric in l, z, one can easily see that symmetry holds for $\mathsf{II}[L;Z]$. Alternatively, symmetry is obvious from:

$$2\mathsf{II}[L;Z] = \mathsf{I}[L;Z] + \sum_l \sum_z \mathsf{P}[L=l]\mathsf{P}[Z=z] \log \frac{\mathsf{P}[L=l]\mathsf{P}[Z=z]}{p(L=l,Z=z)} \ .$$

Dependence: Dependence is a consequence of a well-known property of the symmetric Kullback-Leibler divergence δ [CT06]:

$$\delta(P \parallel Q) = 0 \iff P \text{ and } Q \text{ are independent RVs} \ .$$

Relation to Mutual Information: One can see from

$$2\mathsf{II}[L;Z] = \mathsf{I}[L;Z] + \sum_l \sum_z \mathsf{P}[L=l]\mathsf{P}[Z=z] \log \frac{\mathsf{P}[L=l]\mathsf{P}[Z=z]}{p(L=l,Z=z)} \ ,$$

that $2\mathsf{II}[L;Z] \geq \mathsf{I}[L;Z]$. □

5.1.5 Non-Equivalence of MIA and IIA

Even though $\mathsf{I}[L;Z]$ and $\mathsf{II}[L;Z]$ have the same properties (*cf.* Theorem 2), we show in the following that they are not equivalent. We therefore use the following definition of equivalence.

Definition 6. *Two distances $\mathcal{D}(P,Q)$ and $\mathcal{D}'(P,Q)$ are said to be equivalent if there exist finite constants $\alpha, \beta \geq 0$ such that:*

$$\mathcal{D}(P,Q) \ \leq \ \alpha\mathcal{D}'(P,Q) \quad \text{and} \quad \mathcal{D}'(P,Q) \ \leq \ \beta\mathcal{D}(P,Q) \, .$$

In particular, whenever one of two distances becomes small, so does the other and both distances define the same "topology". For the sake of illustration, we consider the following example:

Example 2. *Let us consider linear correlation $\rho(L;Z) = \frac{\mathsf{Cov}[L;Z]}{\sigma_L \sigma_Z}$ vs. mutual information $\mathsf{I}[L;Z]$. Since it is possible that L and Z are linearly uncorrelated while being dependent (take* e.g. *$Z = L^2$ where $L \sim \mathcal{N}(0,1)$), it follows that an inequality of the form $\mathsf{I}[L;Z] \leq \alpha\rho(L;Z)$ cannot hold and therefore, that correlation and mutual information are not equivalent. The same conclusion goes unchanged for higher-order or nonlinear correlation,* i.e. *$L \sim \mathcal{N}(0,1)$ and $Z = \pm L$. Even L, Z being dependent they are (non-linearly) uncorrelated.*

In case of comparing inter-class and mutual information the inequality $\mathsf{I}[L;Z] \leq 2\mathsf{II}[L;Z]$ (*cf.* Theorem 2) is satisfied. However, we formulate the following theorem in case of Gaussian noise.

Theorem 3. *Let (L,Z) be jointly Gaussian. Then, $\mathsf{I}[L;Z]$ and $\mathsf{II}[L;Z]$ are not equivalent.*

Proof. Let $\lambda = \frac{1}{1-\rho^2}$. Since (L,Z) are jointly Gaussian, $2\mathsf{I}[L;Z] = \log\lambda$ and $2\mathsf{II}[L;Z] = (\lambda - 1)\log e$. So as $\rho \to 1$ the inequality $\mathsf{II}[L;Z] \leq \beta\mathsf{I}[L;Z]$ cannot hold for a constant $\beta \geq 0$. □

5.2 Side Channel Analysis Scenario

Let the noise N be modeled by an arbitrary RV that is not necessarily Gaussian. Note that, the measured leakage $L = Z(k^*)+N = f(X, k^*)+N$ is continuous, while the sensitive variable Z is discrete. To be general, we should write $L = \varphi(Z(k^*))+N$, however, our theoretical outcomes are independent of the leakage model function φ. So, we assume $\varphi = \mathsf{Id}$, while we choose $\varphi =$ Hamming Weight in our simulation in Sec. 5.4.3. The conditional PMF $\mathsf{P}[L = l \mid Z = z]$, *i.e.* the distribution of the measured current L knowing the model Z, is given by:

$$\begin{aligned}
\mathsf{P}[L = l \mid Z = z] &= \textstyle\sum_{x\ ;\ f(x,k)=z} \mathsf{P}[X = x \mid Z(k) = z] \cdot \mathsf{P}[L = l \mid Z = z, X = x] \quad \text{(Bayes Rule)} \\
&= \textstyle\sum_{x\ ;\ f(x,k)=z} \frac{\mathsf{P}[X=x, Z(k)=z]}{\mathsf{P}[Z(k)=z]} \cdot \mathsf{P}[X = x \mid Z = z, X = x] \quad (X \text{ the plaintext is known}) \\
&= \textstyle\sum_{x\ ;\ f(x,k)=z} \frac{\mathsf{P}[X=x]}{\mathsf{P}[Z(k)=z]} \cdot \mathsf{P}[L = l \mid X = x] \quad \text{(referring to N)} \\
&= \textstyle\sum_{x\ ;\ f(x,k)=z} \frac{\mathsf{P}[X=x]}{\mathsf{P}[Z(k)=z]} \cdot \mathsf{P}[l - f(x, k^*) \mid N] \ .
\end{aligned}$$

Next, we rearrange the last equation as the sum over all possible $Z(k^*) = z^*$:

$$\begin{aligned}
\mathsf{P}[L = l \mid Z = z] &= \textstyle\sum_{z^*} \sum_{x\ ;\ \substack{f(x,k)=z \\ f(x,k^*)=z^*}} \frac{\mathsf{P}[X=x]}{\mathsf{P}[Z(k)=z]} \cdot \mathsf{P}[(l - z^*) \mid N] \\
&= \textstyle\sum_{z^*} \frac{\mathsf{P}[Z(k)=z, Z(k^*)=z^*]}{\mathsf{P}[Z(k)=z]} \cdot \mathsf{P}[(l - z^*) \mid N] \qquad \text{(Bayes Rule)} \\
&= \textstyle\sum_{z^*} \mathsf{P}[Z(k^*) = z^* \mid Z(k) = z] \cdot \mathsf{P}[(l - z^*) \mid N] \ .
\end{aligned}$$

In short, we have:

$$\mathsf{P}[L = l \mid Z = z] = \sum_{z^*} \mathsf{P}[z^* \mid z] \cdot \mathsf{P}[(l - z^*) \mid N] \ . \tag{5.6}$$

In particular, when one selects the correct key, *i.e.* $k = k^*$, one has the Kronecker symbol: $\mathsf{P}[Z(k^*) = z^* \mid Z(k) = z] = \delta_{z,z^*}$, so that the density mixture simplifies to:

$$\mathsf{P}[L = l \mid Z = z] = \mathsf{P}[l - z^* \mid N] \ . \tag{5.7}$$

Concluding, we formulate the following analysis scenario:

- **Wrong key hypothesis**: If $k \neq k^*$ then the conditional PDF of $(L \mid Z = z)$ for all z is a PDF mixture whose coefficients depend on z, and whose modes z^* take values on a finite set (including z), *cf.* Eqn. (5.6). We shall also assume that the linear mixture is *non trivial* for any $k \neq k^*$, otherwise we would have $\mathsf{P}[z^* \mid z] = \delta_{z,z^*}$, $Z(k) = Z(k^*)$ almost surely and it would be impossible to distinguish k from k^* and therefore precludes soundness.

- **Correct key hypothesis**: In case $k = k^*$, the conditional PDF of $(L \mid Z = z)$ is simply identically distributed (i.d.) as $N + z$, *cf.* Eqn. (5.7).

5.3 Soundness Proofs for MIA and IIA

The first step in order to prove soundness is to compute MIA and IIA from our model defined in the previous section. Second, we prove soundness for arbitrary noise. Note that, since L is continuous and Z is discrete, we compute conditional entropies or cross-entropies as discrete means in z of integrals in l.

5.3.1 Soundness Proof for Mutual Information Analysis

We recall that $\mathsf{I}[L;Z] = \mathsf{H}[L] - \mathsf{H}[L \mid Z] = \mathsf{H}[L] - \sum_z \mathsf{P}[Z=z]\mathsf{H}[L \mid Z=z]$. So, in case of $k \neq k^*$, the distribution of $(L \mid Z=z)$ follows a mixture of densities of $\mathsf{P}[(l-z^*) \mid N]$ (see Eqn. (5.6)), whereas each density has the same entropy as N. Furthermore, when $k = k^*$, this mixture boils down to the noise density $\mathsf{P}[(l-z^*) \mid N]$, so that $\mathsf{I}[L;Z^*] = \mathsf{H}[L] - \mathsf{H}[N]$. This allows us to give a simpler and more general proof of soundness for MIA than in [PR09, MMPS09], since we do not necessarily require Gaussian noise.

Soundness proof 1. *Let $k \neq k^*$. Since $L \mid Z=z$ is a non trivial linear mixture of densities $\mathsf{P}[(l-z^*) \mid N]$ of the same entropy as $\mathsf{H}[N]$ as well as the property that $H(L)$ is concave [CT06, Theorem 2.7.3], we have:*

$$\mathsf{H}[N] < \mathsf{H}[L \mid Z=z] \tag{5.8}$$

for all z. Taking the expectation over Z, results in:

$$\mathsf{H}[N] < \mathsf{H}[L \mid Z],$$

which already yields the soundness:

$$\mathsf{I}[L;Z^*] > \mathsf{I}[L;Z] \ .$$

□

5.3.2 Soundness Proof for Inter-class Information Analysis

As in Definition 5, the inter-class information $\mathsf{II}[L;Z]$ can be represented as:

$$\mathsf{II}[L;Z] = \frac{\mathsf{H}'[L;Z] - \mathsf{H}[L \mid Z]}{2},$$

where $\mathsf{H}[L \mid Z]$ is as above and:

$$\mathsf{H}'[L;Z] = \sum_z \mathsf{P}[Z=z] \int_l \mathsf{P}[L=l] \log \frac{1}{\mathsf{P}[L=l \mid Z=z]} \, \mathrm{d}l, \tag{5.9}$$

where $\mathsf{P}[L=l \mid Z=z] = \sum_{z^*} \mathsf{P}[z^* \mid z]\mathsf{P}[(l-z^*) \mid N]$ and $\mathsf{P}[L=l] = \sum_z \mathsf{P}[Z=z]\mathsf{P}[L=l \mid Z=z]$. So, inserting $\mathsf{P}[L=l \mid Z=z]$ and $\mathsf{P}[L=l]$ in Eqn. (5.9) gives:

$$\mathsf{H}'[L;Z] = \sum_{z,z'} \mathsf{P}[Z=z]\mathsf{P}[Z=z'] \sum_{z'^*} \mathsf{P}[z'^* \mid z'] \int_l \mathsf{P}[(l-z'^*) \mid N] \log \frac{1}{\sum_{z^*} \mathsf{P}[z^* \mid z]\mathsf{P}[(l-z^*) \mid N]} \, \mathrm{d}l \ .$$

When $k = k^*$, this boils down to:

$$\mathsf{H}'[L; Z^*] = \sum_{z^*, z'^*} \mathsf{P}[z^*]\mathsf{P}[z'^*] \int_l \mathsf{P}[(l - z'^*) \mid N] \log \frac{1}{\mathsf{P}[(l - z^*) \mid N]} \, \mathrm{d}l \ .$$

Again, we give a general proof of soundness without any restriction on the distribution of the noise N.

Soundness proof 2. *Let $k \neq k^*$. By strict concavity of the logarithm (or convexity of $x \mapsto \log(1/x)$):*

$$\begin{aligned}
\mathsf{H}'[L; Z] &= \sum_{z, z'} \mathsf{P}[Z = z]\mathsf{P}[Z = z'] \sum_{z'^*} \mathsf{P}[z'^* \mid z'] \int_l \mathsf{P}[(l - z'^*) \mid N] \log \frac{1}{\sum_{z^*} \mathsf{P}[z^* \mid z]\mathsf{P}[(l - z^*) \mid N]} \, \mathrm{d}l \\
&< \sum_{z, z'} \mathsf{P}[Z = z]\mathsf{P}[Z = z'] \sum_{z^*, z'^*} \mathsf{P}[z'^* \mid z']\mathsf{P}[z^* \mid z] \int_l \mathsf{P}[(l - z'^*) \mid N] \log \frac{1}{\mathsf{P}[(l - z^*) \mid N]} \, \mathrm{d}l \\
&= \sum_{z^*, z'^*} \mathsf{P}[z'^*]\mathsf{P}[z^*] \int_l \mathsf{P}[(l - z'^*) \mid N] \log \frac{1}{\mathsf{P}[(l - z^*) \mid N]} \, \mathrm{d}l = \mathsf{H}'[L; Z^*] \ .
\end{aligned}$$

As for MIA, cf. *Eqn.* (5.8)*, $\mathsf{H}[L \mid Z] > \mathsf{H}[N] = \mathsf{H}[L \mid Z^*]$, we end up with:*

$$\begin{aligned}
\mathsf{II}[L; Z] = \frac{\mathsf{H}'[L; Z] - \mathsf{H}[L \mid Z]}{2} &< \frac{\mathsf{H}'[L; Z] - \mathsf{H}[L \mid Z^*]}{2} \\
&< \frac{\mathsf{H}'[L; Z^*] - \mathsf{H}[L \mid Z^*]}{2} = \mathsf{II}[L; Z^*] \ ,
\end{aligned}$$

which is the soundness statement for IIA. □

5.4 Mutual *Vs* Inter-class Information

Previously, we have shown that both, mutual and inter-class information, are sound under the stated scenario, more precisely, $\mathsf{I}[L; Z] < \mathsf{I}[L; Z^*]$ and $\mathsf{II}[L; Z] < \mathsf{II}[L; Z^*]$. Furthermore, we have shown that in case of Gaussian noise $\mathsf{I}[L; Z]$ and $\mathsf{II}[L; Z]$ are not equivalent (see Theorem 3). Concluding, we will theoretically show that in case of Gaussian noise $\mathsf{II}[L; Z]$ is more discriminating than $\mathsf{I}[L; Z]$, which will also be underlined with practical results.

5.4.1 Inter-class Information for a Gaussian Mixture

In Subsect. 5.3.2 we derived that if $k^* = k$:

$$\mathsf{H}'[L; Z^*] = \sum_{z^*, z'^*} \mathsf{P}[z^*]\mathsf{P}[z'^*] \underbrace{\int_l \mathsf{P}[(l - z'^*) \mid N] \log \frac{1}{\mathsf{P}[(l - z^*) \mid N]} \, \mathrm{d}l}_{(*)} \ .$$

Substituting $\xi = l - z'^*$ in $(*)$ and assuming $N \sim \mathcal{N}(0, \sigma^2)$, results in:

$$\begin{aligned}\int_l \mathsf{P}[\xi \mid N] \log \frac{1}{\mathsf{P}[\xi + z'^* - z^* \mid N]} \,\mathrm{d}\xi &= \frac{1}{2}\log(2\pi\sigma^2) + \frac{\log(e)}{2\sigma^2}\,\mathbb{E}\big(\big(N + z^* - z'^*\big)^2\big) \\ &= \frac{1}{2}\log(2\pi\sigma^2) + \frac{\log(e)}{2\sigma^2}\left(\sigma^2 + (z^* - z'^*)^2\right) \\ &= \frac{1}{2}\log(2\pi e\sigma^2) + \frac{\log(e)}{2\sigma^2}\left(z^* - z'^*\right)^2 \\ &= \mathsf{H}[N] + \frac{\log(e)}{2\sigma^2}\left(z^* - z'^*\right)^2 \quad .\end{aligned}$$

So, by letting Z'^* be independent and identically distributed (*i.i.d.*) as Z^*, we obtain for $\mathsf{H}'[L; Z^*]$:

$$\sum_{z^*, z'^*} \mathsf{P}[z^*]\mathsf{P}[z'^*](z^* - z'^*)^2 = \mathbb{E}\big((Z^* - Z'^*)^2\big) = 2\,\mathbb{E}\big((Z^* - \mathbb{E}[Z^*])^2\big) = 2\sigma^2_{Z^*} \quad .$$

Concluding, as $\mathsf{H}[L \mid Z^*] = \mathsf{H}[N]$, it follows that:

$$\mathsf{II}[L; Z^*] = \frac{\mathsf{H}'[L; Z^*] - \mathsf{H}[N]}{2} = \frac{\log e}{2} \cdot \frac{\sigma^2_{Z^*}}{\sigma^2} \quad . \tag{5.10}$$

5.4.2 Why IIA is more Discriminating than MIA?

In general it is not true that $\mathsf{I}[L; Z^*] < \mathsf{II}[L; Z^*]$. However, we show in the following that this holds in case of Gaussian noise. Since differential entropy (continuous entropy) is maximal for normal variables and $\log x \leq (\log e)(x-1)$, we use Eqn. (5.10) to estimate:

$$\mathsf{I}[L; Z^*] = \mathsf{H}[Z^* + N] - \mathsf{H}[N] \leq \frac{1}{2}\log\frac{\sigma^2_{Z^*} + \sigma^2}{\sigma^2} \leq \frac{\log e}{2}\frac{\sigma^2_{Z^*}}{\sigma^2} = \mathsf{II}[L; Z^*]\,.$$

Note that, the difference $\mathsf{II}[L; Z^*] - \mathsf{I}[L; Z^*]$ increases with $\frac{\sigma^2_{Z^*}}{\sigma^2}$ (high SNR).

Let us now investigate the ability to distinguish between the correct key k^* and the incorrect keys $k \neq k^*$ for MIA as well as for IIA. Of course, the practical outcome will depend on the numerous experimental conditions (*e.g.* number of traces, noise level, *etc.*). Our aim is to gain an insight that will be confirmed by simulations in the next section.

Let further ζ denote the nearest-rival distinguishability:

$$\zeta_{\mathrm{MIA}}(k) = \mathsf{I}[L; Z(k^*)] - \max_{k \neq k^*} \mathsf{I}[L; Z(k)] \tag{5.11}$$

$$\zeta_{\mathrm{IIA}}(k) = \mathsf{II}[L; Z(k^*)] - \max_{k \neq k^*} \mathsf{II}[L; Z(k)]. \tag{5.12}$$

Assume that Z depends in a non-linear way on the key k, for instance when $Z = S(X \oplus k)$ where S is a substitution box, then Z^* is nearly independent on Z for $k \neq k^*$. Therefore, the Gaussian mixture coefficients $\mathsf{P}[Z^*|Z]$ are essentially independent

of the particular value of $Z(k) = z$. So, we can deduce that $\mathsf{I}[Z; Z^*] \ll \mathsf{I}[L; Z^*]$ and also $\mathsf{II}[Z; Z^*] \ll \mathsf{II}[L; Z^*]$.

Also, due to the Markov chain condition on $Z \to Z^* \to L$, it follows $\mathsf{I}[L; Z] \leq \mathsf{I}[Z; Z^*] \ll \mathsf{I}[L; Z^*]$ and $\mathsf{II}[L; Z] \leq \mathsf{II}[Z; Z^*] \ll \mathsf{II}[L; Z^*]$, so $\mathsf{I}[Z; Z^*]$ and $\mathsf{II}[Z; Z^*]$ are negligible compared to $\mathsf{I}[L; Z] \leq \mathsf{II}[L; Z]$. Thus we have:

$$\zeta_{\mathrm{IIA}}(k) \gtrsim \zeta_{\mathrm{MIA}}(k) \ ,$$

especially for a high SNR.

5.4.3 Simulation Results

In our practical attack scenario, we considered a protected PRESENT [BKL+07] implementation, more precisely, we considered the leakage:

$$L = \mathsf{HW}(S^{-1}(X \oplus k^*) \oplus M) + \mathsf{HW}(M) + N \ ,$$

with S^{-1} being the inverse S-box operation in PRESENT, and X and M randomly chosen 4-bit values. Moreover, we used $N \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = \{1, 2\}$ for our simulations. Although the assumption of additive white Gaussian noise may not be always realistic, the investigation is justified by the numerous works carried out with this hypothesis in the community. Note that, we used $Z(k) = \mathsf{HW}(S^{-1}(X \oplus k))$ as the sensitive variable for the attack. For both distinguishers, the maximum value reflects the key prediction, more precisely:

$$k^* = \arg\max_k \mathsf{I}[L; Z] \quad \text{or} \quad k^* = \arg\max_k \mathsf{II}[L; Z] \ .$$

In order to compare the performance of MIA and IIA as side channel distinguishers, we used the first-order success rate as proposed in [SMY09], which we computed over a set of 300 independent experiments, while the secret key was chosen randomly for each experiment. Note that, in order to guarantee a fair comparison, we choose the same data set for MIA and IIA. Furthermore, we used the same histogram-based probability estimation method, using the same number of bins for both attacks. Even though, it is shown in [VCS09] that using the histogram for PDF estimation as suggested in [GBTP08] might not be optimal, our goal is to show and prove the difference in comparing conditional probability distribution and not to find the best method to estimate conditional probabilities.

Figure 5.2 shows the first order success rate for $\sigma = 1, 2$. One can see that the success rate for IIA is above that of MIA in both settings, which confirms our theoretical results made so far.

In addition to this, the ability to distinguish may also depend on the variance of $\mathsf{I}[L; Z^*]$ and $\mathsf{II}[L; Z^*]$ that is estimated from the drawn samples. So, let us recall the following statistical properties of a parameter $\theta = \mathsf{I}[L; Z^*]$ or $\theta = \mathsf{II}[L; Z^*]$ and its estimator $\hat{\theta}$:

$$\begin{aligned} Stdev(\hat{\theta}) &= \sqrt{\mathrm{Var}(\hat{\theta})} = \sqrt{\mathbb{E}[(\hat{\theta} - \mathbb{E}(\hat{\theta}))^2]} \ , \\ Bias(\theta, \hat{\theta}) &= \theta - \mathbb{E}(\hat{\theta}) \ , \\ MSE(\theta, \hat{\theta}) &= \mathrm{Var}(\hat{\theta}) + (Bias(\theta, \hat{\theta}))^2 \ . \end{aligned}$$

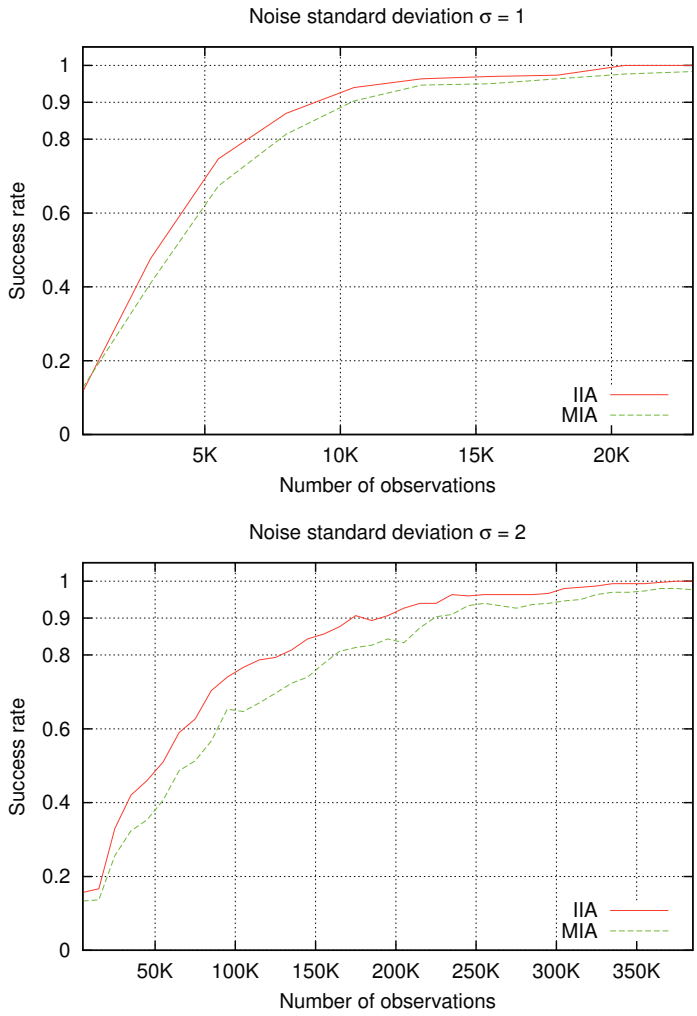


Figure 5.2: First-order success rate for MIA and IIA distinguishers when attacking one masked S-box of PRESENT.

We compute these statistical properties and we show the results for $\sigma = 2$ in Fig. 5.3. The following observations can be emphasized:

- Obviously, the averaged IIA is grater that the averaged MIA, respectively denoted by E(IIA) and E(MIA). This is in line with our previous theoretical study (*i.e.* $\mathsf{II}[L; Z^*] \geq \mathsf{I}[L; Z^*]$).

- The standard deviations, the bias and MSE are of the same degree for both mutual and inter-class information.

Through the theoretical and practical results, we conclude that IIA is more discriminating than MIA.

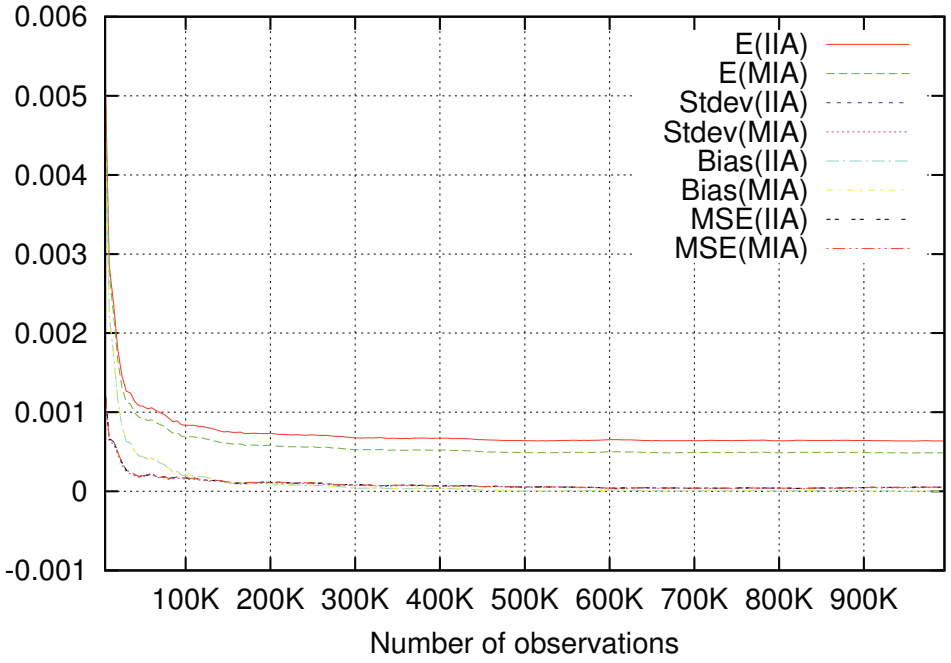


Figure 5.3: Statistical Properties of $\mathsf{I}[L; Z^*]$ and $\mathsf{II}[L; Z^*]$ (cf. Sec. 5.4.2) for $\sigma = 2$.

5.5 Inter-class Approach and the Kolmogorov-Smirnov Test

In this section, we apply the inter-class approach to the Kolmogorov-Smirnov test, introduced in Chapter 2 Sec. 2.3.3, which yield the *Inter-class Kolmogorov-Smirnov Analysis* (IKSA). In contrast to KSA, IKSA consists in comparing the conditional leakages between themselves, pairwise. The inter-class KSA distinguisher can be written as:

$$D_{\text{IKSA}} = \frac{1}{2} \cdot \mathbb{E}_{Z,Z'}\big(\sup_{l \in \mathcal{L}} \mid F_{L|Z}(l) - F_{L|Z'}(l) \mid\big) \ , \tag{5.13}$$

where F_X is the cumulative distribution function of a RV X and Z' is an independent copy of Z.

In order to compare the KSA and IKSA distinguishers, we propose a simulation-based "fair" framework which takes into account the different errors of the estimation tools used in the simulation process.

5.5.1 Existing Frameworks

In this section, we analyze previous comparison frameworks, highlight possible limitations and motivate for a new setting. The first evaluation framework was proposed by Standaert *et al.* in [SMY09]. For the comparison of distinguishers, the

authors suggest metrics like o^{th}-order success rate (with $o \in [\![1, 2^n]\!]$) or guessing entropy. In another framework [WOM11, WO11], the distance to the nearest rival is employed; it is the same definition as previously termed "Correlation Contrast" in [BP10]. Many other metrics can be invented, such as the signal (distinguisher expected value for the correct key $k^\star$) to noise (distinguisher variance over incorrect keys $k \in \mathbb{F}_2^n \backslash \{k^\star\}$) ratio [GHP04] or the norm-2 of the characterized coefficients in a stochastic profiling [HSS12].

Recent analyses [WO11] suggest pitfalls in the evaluation methodologies for distinguishers. Errors can arise from many sources:

- **Estimation bias**: The estimator does not converge to the correct value. For instance, the MIA with few bins for the PDF estimation can have a square bias significantly larger than its variance.
- **Estimation algorithm**: It can approximate the data. Whatever the kernels used in PDF constructions [PR09], the binning of the observed side channel reduces its accuracy.
- **Success rate error**: It is a random variable that has its own variance.
- **Sampling errors**: The random variables are not drawn a sufficient number of times and thus do not obey to their law. As a rule of thumb, estimations are incorrect if a discrete RV has been measured a fewer number of times than the size of its set of possible values.

In the sequel, we intend to compare KSA [WOM11] and IKSA on a fair basis.

5.5.2 Increasing the Fairness of the Estimations

We try here to eliminate or at least bound the errors listed in the previous subsection.

- The KS distance is shown to be unbiased by the Glivenko-Cantelli theorem [Wel77], (and furthermore there is a uniform convergence). This is never true for entropy estimators (for instance, all the estimation methods presented in [PR09] are biased).
- We use an estimation algorithm that keeps the data unchanged (see Eqn. (5.13)). Our estimation for KSA is the same as that of Whitnall, Oswald and Mather [WOM11].
- We quantify the success rate error. An upper bound of the variance of the success rate error is shown below to behave as $1/\sqrt{Q}$, where Q is the number of experiments.
- We consider attacks with a noise large enough for the success rate to be well below 100% for a number of queries smaller than the size of its definition set.[1]

[1] For instance, it can be seen in Fig. 5.6 that for the unprotected (resp. Boolean masked) AES, the number of traces to recover the key successfully with probability $> 80\%$ is about $2,000$ (resp. $70,000$), which is significantly greater than the number of possible plaintexts (*i.e.* $2^n = 256$) for $\sigma \geqslant 8$.

5.5.3 Bounding the Success Rate

Let S_i denote *i.i.d.* Bernoulli variables that take binary values in $\{0,1\}$ with probabilities p and $1-p$, where p is the success probability. The success rate is defined as $SR = \frac{1}{Q}\sum_{i=1}^{Q} S_i$ and has expectation $\mathbb{E}[SR] = p$, *i.e.* SR is an unbiased estimator of the success probability. According to the strong law of large numbers the success rate converges to p almost surely: $SR \xrightarrow{a.s.} p$. In addition, $\mathbb{E}[SR] = p$, *i.e.* SR is an unbiased estimator of the success rate. Now, the standard deviation of SR is easily computed:

$$\sigma(SR) = \sqrt{\frac{1}{Q^2} \cdot Q \cdot \sigma^2(S_j)} = \sqrt{\frac{p\cdot(1-p)}{Q}} \ . \tag{5.14}$$

Thus, the estimation error on the success rate is maximized when p is close to $1/2$, and is minimized when p is almost equal to 0 or 1.

In practice, one wishes to compare the success rates of two distinguishers by examining the values of intermediate p (*i.e.* $p \approx 1/2$). Note that there is a uniform bound $\sigma(SR) \leqslant \frac{1}{2\sqrt{Q}}$, but the error bars can be a function of p and Q. The criterion for analyzing experiments will be that errors bars never overlap. Otherwise (see Fig. 5.4 for $Q = 10$), more experiments must be done, so as to reach a situation such as Fig. 5.4 for $Q = 200$. The exact number of experiments depends on the distinguishers to be relatively characterized. The closer they are in success rate, the more experiments are required.

5.5.4 Simulation Results

In this section, we perform several attack experiments to compare KSA and IKSA. Our methodology allows to observe how the different attacks behave against unprotected reference and a masking scheme, and to compare their resistance for different noise's standard deviations.

In what follows, we consider the same simulation setting described in Sec. 5.4.3. For comparison purposes, we compute the same metric for other univariate distinguishers: MIA, DPA, CPA, VPA and 2O-CPA. Figure 5.6 summarizes the number of leakage measurements required to observe a success rate of 90% in retrieving $k^\star$ for those SCA attacks. This figure is the compilation of success rates curves obtained for different values of the noise standard deviation (see examples in Fig. 5.5).

The results presented in Fig. 5.6 show the significant gain in the number of measurements needed induced when using IKSA compared to KSA attack. Our new distinguisher compares favorably to KSA: the IKSA attack outperforms the KSA attack when targeting the unprotected implementation or even when the Boolean masking scheme is used for the protection. As expected, CPA performs well in both scenarios since the dependency between the leakage and the model is linear. But, we like to stress that we focus only on information-theoretic distinguishers which are generic.

In [MOS11], a notion of asymptotic equivalence (noted "$\equiv$") for side channel distinguishers is introduced: two distinguishers are said equivalent if the number of traces to overcome a given success rate (say 90%) decreases when the noise variance

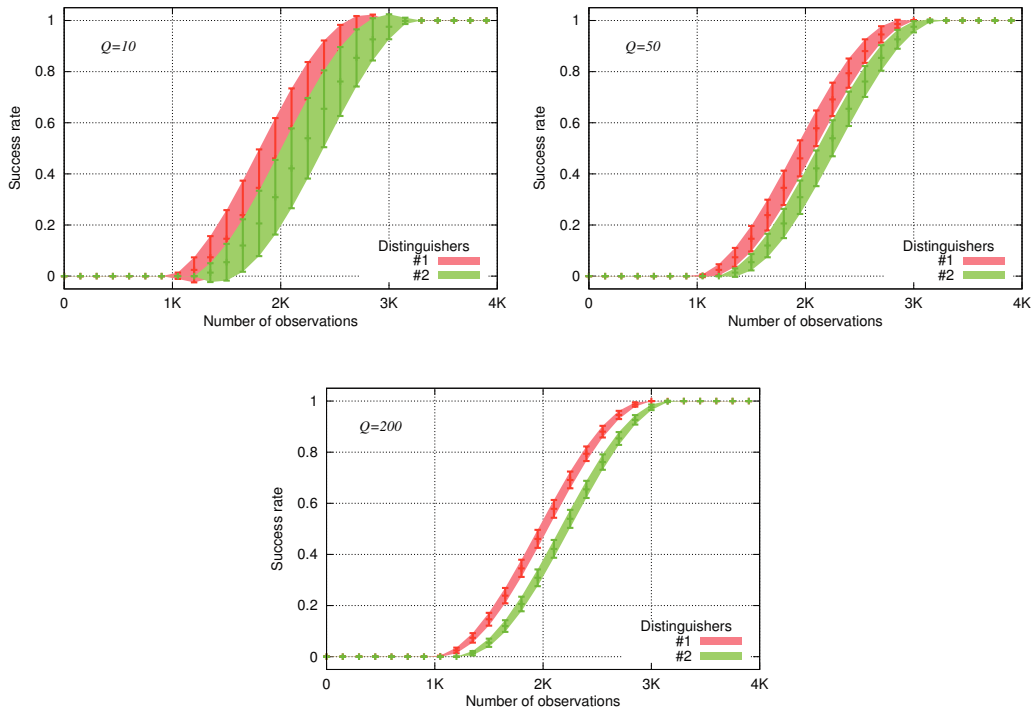


Figure 5.4: Examples of success rates errors (Eqn. (5.14)) for various numbers of experiments.

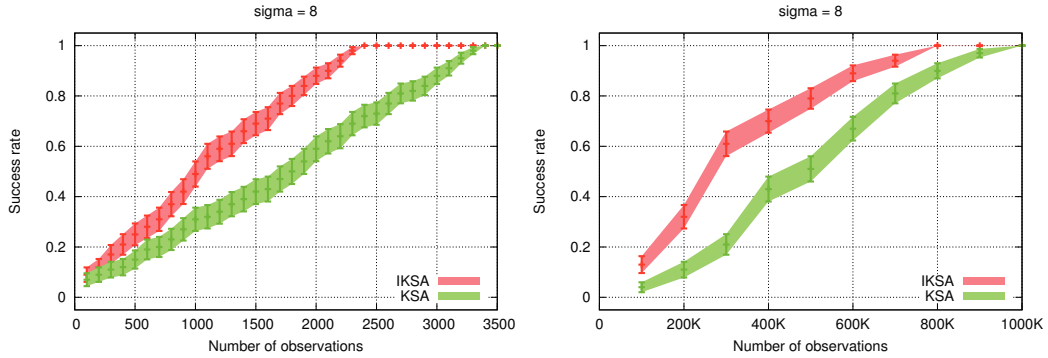


Figure 5.5: Success rate of both IKSA and KSA distinguishers when attacking one S-box of an unprotected AES (*left*) and of a Boolean masking scheme (*right*).

increases. For example, the likelihood and the Pearson correlation are equivalent in this sense.

A look at $N_{90\%}$ curves in Fig. 5.6 shows that other univariate distinguishers exhibit a similar equivalence law:

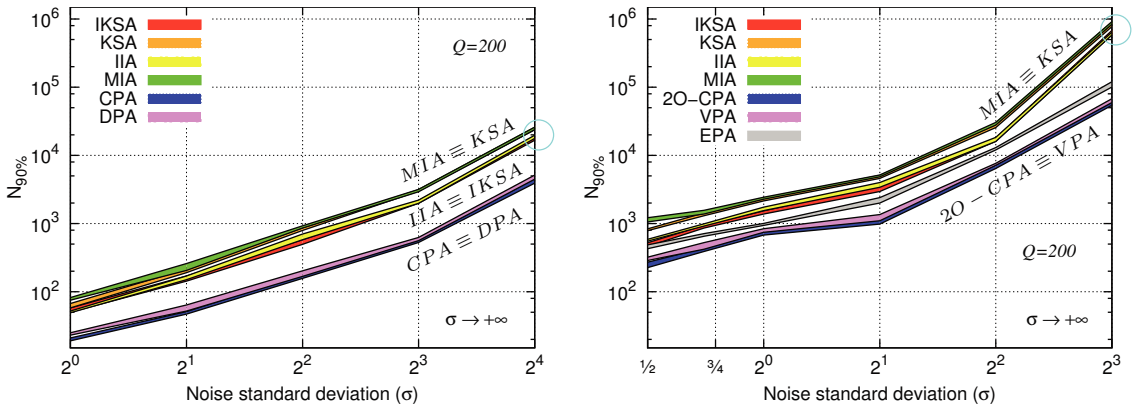


Figure 5.6: Evaluation of $N_{90\%}$, the number of messages to achieve a success rate greater than 90%, according to the noise standard deviation when attacking unprotected (*left*) and Boolean masking (*right*) AES implementation.

- DPA ≡ CPA on an unprotected implementation (*left*);
- 2O-CPA ≡ VPA on a first-order masked implementation (*right*);
- KSA ≡ MIA on both implementations (already proved in [WOM11]).

However, IKSA and KSA are *not equivalent*. The difference between IKSA and KSA ≡ MIA is materialized in Fig. 5.6 as a circle in cyan color. To the best of our knowledge, it is the first time that two distinguishers that do not become equivalent in the sense of [MOS11] are put forward. Incidentally, we note that this conclusion could not have been derived mathematically under the usual Gaussian approximation, because under this approximation equivalence holds as $\sigma \to +\infty$.

5.6 Conclusions

In this chapter, we introduced the new inter-class concept to distinguish between various partitionings. We applied this concept to the mutual information (IIA) and the Kolmogorov-Smirnov test (IKSA). Additionally, we made a first theoretical analysis why IIA may be more discriminating than MIA, which we also confirmed by simulations. We also proposed a simulation-based fair framework to compare the two distinguishers KSA and IKSA. Our framework takes in account the different sources of errors estimations. We used this framework to compare KSA to IKSA using the success rate metric. Security metrics are clearly in favor of inter-class distinguishers even when the implementation is unprotected or protected using a first-order Boolean masking countermeasure (with a linear leakage model).

Part III
Characterization of Masking Schemes

Chapter 6

Introduction to Higher-Order Masking Countermeasures

Masking [CJRR99, GP99] is a countermeasure against side channel attacks, that is suitable for both hardware and software cryptographic implementations. It consists in changing the variable representation into randomized shares [CJRR99], and can thus be qualified as an algorithmic countermeasure. Notably, masking does not rely on specific hardware properties: as opposed to dual-rail protection [MOP06, Chp. 7] that demands some physical indiscernibility. We present in this chapter a brief overview of masking theory. Then, we introduce the notion of *HO-CPA immunity* as a new SCA metric allowing us to assess both the resistance against higher-order CPA attacks and the amount of leakage. Finally, we overview some widely used first-order masking schemes.

Contents

6.1 State-of-the-Art: Higher-Order Masking

As stated in Chapter 2 Sec. 2.5.4, a d^{th}-order masking scheme consists in splitting a sensitive variable $Z \in \mathbb{F}_2^n$ (that can be deduced from either the plaintext or the ciphertext through few subkeys hypotheses) into $d+1$ random shares, noted $\vec{S} = (S_i)_{i \in [\![0,d]\!]}$, in such a way that the relation $S_0 \perp \cdots \perp S_d = Z$ is satisfied for a group

operation $\perp$ (*e.g.* the XOR operation in Boolean masking). We recall hereafter the definition of the masking soundness.

Definition 7. ***(Masking d^{th}-Order Soundness)**: The masking is sound at d^{th}-order if:*

- *Z can be deterministically reconstructed knowing the $d+1$ shares, while*
- *no information about Z can be extracted from strictly less than $d+1$ shares.*

In order to study a masking scheme resistance in a SCA context, one usually associates each share with a noisy observation of it, modeled by a noisy function $\ell_i : X \mapsto \varphi_i(X) + N_i$ where N_i is an independent and Gaussian noise and φ_i is a deterministic but unknown function sometimes approximated by the Hamming weight. We denote by L_i the RV $\ell_i(S_i)$ and summarize by $\vec{L}$ the tuple of $(L_i)_{i\in[\![0,d]\!]}$[1].

Depending on the contexts (*e.g.* software or hardware implementation), the SCA attacker of a masking scheme gets either direct observations of the $(d+1)$-tuple $\vec{L}$ (in the software case) or observations of a RV in the form $\mathcal{C}_{\text{device}}(\vec{L})$ (in the hardware case), where $\mathcal{C}_{\text{device}} : \mathbb{R}^{d+1} \to \mathbb{R}^a$ is an unknown function determined by the hardware architecture ($a \in \mathbb{N}^*$). After collecting the side channel leakage, the attacker can apply a pre-processing of his choice before performing any statistical analysis. It is denoted by $\mathcal{C}_{\text{attacker}}$ and defined on sets $\mathbb{R}^a \to \mathbb{R}^b$, where the positive integer b is chosen according to the subsequent statistical test (*e.g.* $b = 1$ for univariate analyses such as those based on a correlation coefficient, or $b = a$ for multivariate analyses such as those based on mutual information [GBPV10]). Eventually, the exploited leakage is the RV $\mathcal{C}_{\text{total}}(\vec{L})$, which is equal to $\mathcal{C}_{\text{attacker}}(\mathcal{C}_{\text{device}}(\vec{L}))$. In the sequel, we focus on univariate combinations, *i.e.* $b = 1$, that are more efficient to estimate. Thus, the function $\mathcal{C}_{\text{total}} \doteq \mathcal{C}_{\text{attacker}} \circ \mathcal{C}_{\text{device}}$ can be developed as a polynomial in $\mathbb{R}[L_0, \cdots, L_d] = \mathbb{R}[\vec{L}]$, noted $\mathcal{C}_{\text{total}}(\vec{L}) = \sum_{\vec{\alpha}=(\alpha_0,\cdots,\alpha_d)} a_{\vec{\alpha}} \vec{L}^{\vec{\alpha}}$, where $\vec{\alpha} \in \mathbb{N}^{d+1}$, $\vec{L}^{\vec{\alpha}}$ denotes the monomial term $\prod_{i=0}^{d} L_i^{\alpha_i}$ and $a_{\vec{\alpha}}$ are real coefficients. We recall that the *degree* of a multivariate polynomial is the maximum over the sum of the exponents of the variables in any monomial term. The degree of $\mathcal{C}_{\text{total}}(\vec{L})$ is denoted by δ and we denote by $\vec{\tilde{\alpha}}$ one of the vectors that satisfy $a_{\vec{\alpha}} \neq 0$ and $||\vec{\alpha}||_1 = \sum_{i=0}^{d} \alpha_i = \delta$.

6.1.1 Notation and Basics of Statistics

We introduce some notation which will be useful in the sequel. We denote by μ_h and σ_h^2 the mean and the variance of the conditional RV $\left(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h\right)$. We call $\mu_{\text{tot}} = \sum_h \mathsf{P}[\mathsf{HW}(Z) = h] \cdot \mu_h$ the mean of $\mathcal{C}_{\text{total}}(\vec{L})$. The total variance σ^2_{tot} of $\mathcal{C}_{\text{total}}(\vec{L})$ decomposes into the sum of *inter-* and *intra-class* variance, denoted by σ^2_{inter} and σ^2_{intra} respectively. Those quantities are defined using the law of total variance as: $\sigma^2_{\text{inter}} \doteq \sum_h \mathsf{P}[\mathsf{HW}(Z) = h] \cdot (\mu_h - \mu_{\text{tot}})^2$ and $\sigma^2_{\text{intra}} \doteq \sum_h \mathsf{P}[\mathsf{HW}(Z) = h] \cdot \sigma_h^2$.

[1] This definition forbids the modelling of glitches (as those put forward in [MS06]), that are a function of simultaneously at least two different shares S_i, S_j. But it complies with our strategy of allocating one dedicated hardware resource for each share.

In the presence of countermeasures, the central moment $\mu_i(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h) \doteq \mathbb{E}[\Big(\mathcal{C}_{\text{total}}(\vec{L}) - \mu_{\text{tot}}\Big)^i \mid \mathsf{HW}(Z) = h]$ of order i can be independent of h. For instance, we have seen in the previous part of this thesis that when the first-order masking is applied, the RVs $\Big(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h\Big)$ have the same mean. So in practice, the attacker will typically try to compute the moments starting from low orders $i \geq 1$, because their estimation is less affected by the measurement noise.

6.1.2 Attack Efficiency and Degree of $\mathcal{C}_{\text{total}}$

The attacker builds the quantity $\mathcal{C}_{\text{total}}$ before distinguishing between subkey hypotheses. Hence, this quantity is all the more suitable from the attacker point of view as its variance is low, since the capability of distinguishing depends on the SNR; the conclusions in [WO11] can be generalized to any order. The variance of the noise in the signal $\mathcal{C}_{\text{total}}(\vec{L})$ is equal to $\mathsf{Var}[\mathcal{C}_{\text{total}}(\vec{L}) \mid \vec{S} = \vec{s}]$. Assuming that the means of L_i are positive (which is the case in practice when the measurements correspond to the power consumption and the noise has a zero mean) and the variances are greater than 1, it can be checked that we have:

$$\begin{aligned} & \mathsf{Var}[\mathcal{C}_{\text{total}}(\vec{L}) \mid \vec{S} = \vec{s})] \\ \geq & \sum_{\vec{\alpha}=(\alpha_0,\cdots,\alpha_d)\in\mathbb{N}^{d+1}} a_{\vec{\alpha}}^2 \mathsf{Var}[\vec{L}^{\vec{\alpha}} \mid \vec{S} = \vec{s}] \\ \geq & \sum_{\vec{\alpha}=(\alpha_0,\cdots,\alpha_d)\in\mathbb{N}^{d+1}} a_{\vec{\alpha}}^2 \prod_{i=0}^{d} \mathsf{Var}[L_i^{\alpha_i} \mid S_i = s_i] \ . \end{aligned} \tag{6.1}$$

When $L_i \mid S_i = s_i$ has a normal distribution, it can be checked[2] that $\mathsf{Var}[L_i^{\alpha_i} \mid S_i = s_i] \doteq \mathbb{E}[L_i^{2\alpha_i} \mid S_i = s_i] - \mathbb{E}[L_i^{\alpha_i} \mid S_i = s_i]^2$ is always greater than $\mathsf{Var}[L_i \mid S_i = s_i]^{\alpha_i}$. So, we have:

$$\mathsf{Var}[\mathcal{C}_{\text{total}}(\vec{L}) \mid \vec{S} = \vec{s}] \geq a_{\hat{\vec{\alpha}}}^2 \prod_{i=0}^{d} \sigma^{2\hat{\alpha}_i} = a_{\hat{\vec{\alpha}}}^2 \times (\sigma^{\delta})^2 \, . \tag{6.2}$$

In view of Eqn. (6.2), it comes that the amount of noise in $\mathcal{C}_{\text{total}}(\vec{L})$ is upper bounded by a bound that increases exponentially with the degree δ of $\mathcal{C}_{\text{total}}(\vec{L})$. Therefore, it is preferable for the attacker to minimize the total degree of the polynomial $\mathcal{C}_{\text{total}}$.

6.1.3 Information-Theoretic Characterization of Masking

In order to characterize the $\mathcal{C}_{\text{total}}(\vec{L})$ polynomial from an information-theoretic point of view, we introduce the notion of *multivariate degree*[3]. The multivariate degree of

[2] In [WL03, Eqn. (16)], an analytical formula for the variance of a product normal distribution is given. It allows to check the inequality for $\alpha_i = 2$. For higher-orders, we need to refer to the difference between the *raw moment* of a Gaussian RV X: $\mathbb{E}[X^{2\alpha_i}] - \mathbb{E}[X^{\alpha_i}]^2$ is greater than $\mathsf{Var}[X]^{\alpha_i}$. The proof relies on the development of generalized Hermite polynomials.

[3] In the context of polynomials in variables $L_0, \cdots, L_d$ over the field $\mathbb{K}$ (*e.g.* $\mathbb{K} = \mathbb{R}$), our definition of *multivariate degree* coincides with the "usual" degree of polynomials in the algebra

a monomial $\prod_{i=0}^{d} L_i^{\alpha_i}$ is equal to $\sum_{i=0}^{d}(\alpha_i \neq 0)$, where α_i are natural numbers, *i.e.* $\alpha_i \in \mathbb{N}$, and where $(\alpha_i \neq 0)$ is a notation for the function that equals 1 if $\alpha_i \neq 0$ and 0 otherwise. We also emphasize that the multivariate degree is smaller than the degree. We start with this basic lemma:

Lemma 1. *If a masking scheme is d^{th}-order sound, then the mutual information between any monomial in L_i, of multivariate degree lower than or equal to d, and Z is null.*

Proof. If a masking scheme is d^{th}-order sound, then $\mathsf{I}[Z;(S_i)_{i\in I}] = 0$ if $\#I \leq d$. Now, for any function ψ, $\mathsf{I}[Z;(S_i)_{i\in I}] \geq \mathsf{I}[Z;\psi((S_i)_{i\in I})]$. So, if ψ is taken as a monomial in $(L_i)_{i\in[\![0,d]\!]} = (\ell_i(S_i))_{i\in[\![0,d]\!]}$ of multivariate degree less than or equal to d, we have $\mathsf{I}[Z;\psi((S_i)_{i\in I})] \leq 0$, hence $\mathsf{I}[Z;\psi((S_i)_{i\in I})] = 0$. □

As a consequence of Lemma 1, for any sound d^{th}-order masking schemes, every moment of order lower than or equal to d is constant. Hence, an attacker may attempt to apply the following strategy: choose a $\mathcal{C}_{\text{attacker}}$ such as $\mathcal{C}_{\text{total}}$ is of multivariate degree strictly greater than d. Together with Eqn. (6.2), this result implies that the adversary must choose $\mathcal{C}_{\text{attacker}}$ such that the multivariate degree of $\mathcal{C}_{\text{total}}$ is at least $d+1$, and that the (*regular*) degree must not be too high otherwise the SCA attack efficiency decreases. For hardware implementations of d^{th}-order masking schemes, it may happen that the $d+1$ shares $\vec{S}$ are manipulated simultaneously. This can be the consequence of some undesired synchronization issues or even this can be wanted by the hardware designer. Indeed, processing the shares in parallel allows keeping the throughput unchanged. In the following, we assume that the hardware implementation of the masking scheme which is considered indeed has this property. Under this assumption, the univariate leakage exploitable by SCA depends on all the shares which implies that $\mathcal{C}_{\text{device}}$ has output dimension equal to 1. We denote by d_{device} the multivariate degree of $\mathcal{C}_{\text{device}}(\vec{L})$. Since, according to Lemma 1, the multivariate degree of $\mathcal{C}_{\text{total}} = \mathcal{C}_{\text{attacker}} \circ \mathcal{C}_{\text{device}}$ must be at least $d+1$ to enable effective SCA attack, having the degree of $\mathcal{C}_{\text{device}}$ equal to d_{device} implies that $\mathcal{C}_{\text{attacker}}$ must be chosen with a multivariate degree d_{attacker} greater than or equal to $d+1-d_{\text{device}}$. This implies that $\mathcal{C}_{\text{attacker}}$, viewed as a polynomial over $\mathbb{R}[X]$ must contain monomials in the form X^i with i greater than or equal to $d+1-d_{\text{device}}$.

For software implementations of d^{th}-order masking schemes, the $d+1$ shares S_i are manipulated sequentially. It therefore makes sense to assume that their manipulation leaks independently, which in turn implies that $\mathcal{C}_{\text{device}}$ is the identity function (*i.e.* $d_{\text{device}} = 0$). In this case and according to Lemma 1, $\mathcal{C}_{\text{attacker}}$ must be chosen with a multivariate degree greater than or equal to $d+1$ to enable effective SCA attack.

$\mathbb{K}[L_0,\cdots,L_d]/(\prod_{i=0}^{d} L_i^2 - L_i)$, also called sometimes the *algebraic degree*.

6.2 HO-CPA Attacks and HO-CPA Immunity

In this section, we propose a new SCA metric, called HO-CPA immunity, that allows estimating both the efficiency of existing attacks and the quantity of information leaked.

6.2.1 HO-CPA Immunity

Definition 8. *The HO-CPA immunity of the RV $\mathcal{C}_{total}(\vec{L})$ is the order of the smallest (central) moment of $\left(\mathcal{C}_{total}(\vec{L}) \mid \mathsf{HW}(Z) = h\right)$ which is dependent on h.*

The HO-CPA immunity of $\mathcal{C}_{\text{total}}$ is denoted by HCI in the following. The minimal value of the HO-CPA immunity is 1 and it is reached when the distributions of the RV $\left(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h\right)$ do not have the same mean. This is the case of unprotected circuits, for which a first-order CPA [BCO04] works.

The HO-CPA immunity is larger than or equal to 2 when the distributions are balanced (*i.e.* $\forall h, \mu_h = \mu_{\text{tot}}$). In this case, the inter-class variance is null and the total variance σ^2_{tot} is equal to the intra-class variance $\sigma^2_{\text{intra}} = \sum_h \mathsf{P}[\mathsf{HW}(Z) = h] \times \mu_2(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h)$. If the $\mu_2(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h)$ are not all equal, then $\mathsf{HCI} = 2$ and a second-order CPA using the moments of order 2 (or a variance-based attack) is possible.

The motivation of the HO-CPA immunity definition is thus straightforward. All HO-CPA using the moments of order $i < \mathsf{HCI}$ will fail, because the moments are independent of h. Thus the HO-CPA immunity is equal to the smallest order of the moments for which an HO-CPA attack can be successful. This is illustrated for 4-bit variables in Tab. 6.1. In this table, the number of lines in gray is equal to $\mathsf{HCI} - 1$ (Definition 8).

6.2.2 Link between $\mathsf{I}[\mathcal{C}_{\text{total}}(\vec{L}); Z]$ and the HO-CPA Immunity

The HO-CPA reveals linear dependencies between the RVs $\mathcal{C}_{\text{total}}(\vec{L})$ and Z. Unless the RVs $\left(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h\right)$ are identically distributed for every h, the mutual information $\mathsf{I}[\mathcal{C}_{\text{total}}(\vec{L}); Z]$ will be non-zero. We show in the following theorem that HCI is also relevant to quantify the amount of information leaked when the noise level is high.

In terms of mutual information, the impact of the noise $N \sim \mathcal{N}(0, \sigma^2)$ is quantified by Theorem 4, that extends that of Waddle and Wagner [WW04] to the multibit case.

Theorem 4. *Let σ denote the standard deviation of the noise, the mutual information $\mathsf{I}[\mathcal{C}_{total}(\vec{L}); Z]$ tends towards $\mathcal{O}\left(\sigma^{-2\times\mathsf{HCI}}\right)$ when σ tends towards infinity.*

To prove the theorem, we introduce the notions of *cumulants* of the RV X, denoted by $k_i(X)$, that correspond to the monomials in the Taylor series of the function $t \in \mathbb{R} \mapsto \ln(\mathbb{E}(\exp(t \cdot X))) = \sum_{i=0}^{+\infty} k_i(X)\frac{t^i}{i!}$, for $t \in \mathbb{R}$. The proof will use the following lemma.

Table 6.1: Statistics about some leakage models on words of $n = 4$ bitwidth, without noise (*i.e.* $\sigma = 0$).

	R.V.	L	$L \mid \mathsf{HW}(Z) = 0$	$L \mid \mathsf{HW}(Z) = 1$	$L \mid \mathsf{HW}(Z) = 2$	$L \mid \mathsf{HW}(Z) = 3$	$L \mid \mathsf{HW}(Z) = 4$
Plain hardware implementation with $d = 0$ mask (unprotected reference).							
$\mathsf{HCI} = 1$	$\mu_1 = \mathbb{E}(\cdot)$	2.000	0.000	1.000	2.000	3.000	4.000
	$\mu_2 = \mathbb{E}((\cdot - \mu_1)^2)$	1.000	0.000	0.000	0.000	0.000	0.000
	$\mu_3 = \mathbb{E}((\cdot - \mu_1)^3)$	0.000	0.000	0.000	0.000	0.000	0.000
	$\mu_4 = \mathbb{E}((\cdot - \mu_1)^4)$	2.500	0.000	0.000	0.000	0.000	0.000
	Entropy [bit]	2.031	0.000	0.000	0.000	0.000	0.000
Plain hardware implementation with $d = 1$ mask.							
$\mathsf{HCI} = 2$	$\mu_1 = \mathbb{E}(\cdot)$	4.000	4.000	4.000	4.000	4.000	4.000
	$\mu_2 = \mathbb{E}((\cdot - \mu_1)^2)$	2.000	4.000	3.000	2.000	1.000	0.000
	$\mu_3 = \mathbb{E}((\cdot - \mu_1)^3)$	0.000	0.000	0.000	0.000	0.000	0.000
	$\mu_4 = \mathbb{E}((\cdot - \mu_1)^4)$	11.000	40.000	21.000	8.000	1.000	0.000
	Entropy [bit]	2.544	2.031	1.811	1.500	1.000	0.000
Plain hardware implementation with $d = 2$ masks.							
$\mathsf{HCI} = 3$	$\mu_1 = \mathbb{E}(\cdot)$	6.000	6.000	6.000	6.000	6.000	6.000
	$\mu_2 = \mathbb{E}((\cdot - \mu_1)^2)$	3.000	3.000	3.000	3.000	3.000	3.000
	$\mu_3 = \mathbb{E}((\cdot - \mu_1)^3)$	0.000	-3.000	-1.500	0.000	1.500	3.000
	$\mu_4 = \mathbb{E}((\cdot - \mu_1)^4)$	25.500	25.500	25.500	25.500	25.500	25.500
	Entropy [bit]	2.839	1.762	1.822	1.836	1.822	1.762
Plain hardware implementation with $d = 3$ masks.							
$\mathsf{HCI} = 4$	$\mu_1 = \mathbb{E}(\cdot)$	8.000	8.000	8.000	8.000	8.000	8.000
	$\mu_2 = \mathbb{E}((\cdot - \mu_1)^2)$	4.000	4.000	4.000	4.000	4.000	4.000
	$\mu_3 = \mathbb{E}((\cdot - \mu_1)^3)$	0.000	0.000	0.000	0.000	0.000	0.000
	$\mu_4 = \mathbb{E}((\cdot - \mu_1)^4)$	46.000	52.000	49.000	46.000	43.000	40.000
	Entropy [bit]	3.047	2.044	2.047	2.046	2.043	2.031
Plain hardware implementation with $d = 4$ masks.							
$\mathsf{HCI} = 5$	$\mu_1 = \mathbb{E}(\cdot)$	10.000	10.000	10.000	10.000	10.000	10.000
	$\mu_2 = \mathbb{E}((\cdot - \mu_1)^2)$	5.000	5.000	5.000	5.000	5.000	5.000
	$\mu_3 = \mathbb{E}((\cdot - \mu_1)^3)$	0.000	0.000	0.000	0.000	0.000	0.000
	$\mu_4 = \mathbb{E}((\cdot - \mu_1)^4)$	72.500	72.500	72.500	72.500	72.500	72.500
	Entropy [bit]	3.208	2.207	2.208	2.208	2.208	2.207

Lemma 2. *If $\mathcal{C}_{total}(\vec{L})$ has a HO-CPA immunity equal to* HCI*, then for every i in $[\![0, \mathsf{HCI}[\![$ and every Z in $\mathbb{F}_2^n$ we have, $k_i(\mathcal{C}_{total}(\vec{L}) \mid \mathsf{HW}(Z) = h) = k_i(\mathcal{C}_{total}(\vec{L}))$.*

Proof. First of all, we notice that $\forall i \in [\![0, \mathsf{HCI}[\![$, the cumulants $k_i(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h)$ are equal. The reason is that for any law X, $k_j(X)$ can be expressed as a function of $\mu_i(X)$ for $0 \leq i \leq j$ (and reciprocally). For instance $k_3(X) = \mu_3(X)$, $k_4(X) = \mu_4(X) - 3\mu_2^2(X)$, $k_5(X) = \mu_5(X) - 10\mu_3(X)\mu_2(X)$, *etc.* Now, according to Definition 8, if $\mathcal{C}_{\text{total}}(\vec{L})$ has HO-CPA immunity HCI, then all $\mu_j(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h)$ for $0 \leq i < \mathsf{HCI}$ are independent of $\mathsf{HW}(Z)$. Consequently, the same holds for the cumulants of orders $i \in [\![0, \mathsf{HCI}[\![$. Eventually, as:

$$\forall i < \mathsf{HCI}, \forall \mathsf{HW}(Z) = h, \mu_i(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h) = \mu_i(\mathcal{C}_{\text{total}}(\vec{L})) \ ,$$

we also have $k_i(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h) = k_i(\mathcal{C}_{\text{total}}(\vec{L}))$. □

We give hereafter the proof of Theorem 4.

Proof. After using the Edgeworth expansion to approximate the PDFs of $\left(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h\right)$, we deduce from Lemma 2 in [LB10] (extended to laws of common variance $\sigma_{\text{tot}}^2 + \sigma^2 \neq 1$) that:

$$\mathsf{I}[\mathcal{C}_{\text{total}}(\vec{L}); Z] = \tag{6.3}$$
$$\sum_{i=0}^{+\infty} \frac{1}{2i!} \sum_h \mathsf{P}[\mathsf{HW}(Z) = h] \frac{\left(k_i(\mathcal{C}_{\text{total}}(\vec{L}) \mid Z = h) - k_i(\mathcal{C}_{\text{total}}(\vec{L}))\right)^2}{\left(\sigma_{\text{tot}}^2 + \sigma^2\right)^i} .$$

Then, according to Lemma 2, the first non-zero term in the summation in Eqn. (6.3) is at index $i = \mathsf{HCI}$. So, we have:

$$\begin{aligned} \mathsf{I}[\mathcal{C}_{\text{total}}(\vec{L}); Z] &= \sum_{i=\mathsf{HCI}}^{+\infty} \frac{1}{2 \cdot i!} \sum_h \mathsf{P}[\mathsf{HW}(Z) = h] \frac{\left(k_i(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h) - k_i(\mathcal{C}_{\text{total}}(\vec{L}))\right)^2}{\left(\sigma_{\text{tot}}^2 + \sigma^2\right)^i} \\ &= \mathcal{O}\left(\sigma^{-2 \times \mathsf{HCI}}\right) \ , \end{aligned}$$

which achieves to prove Theorem 4. □

Our main interest in Theorem 4 is that it gives the dependence between the leakage, the noise variance σ^2 and the HCI order. It shows that the higher HCI (*i.e.* the more statistical moments of $\left(\mathcal{C}_{\text{total}}(\vec{L}) \mid \mathsf{HW}(Z) = h\right)$ are constant with respect to $\mathsf{HW}(Z)$, the less information is leaked by the device.

6.3 First-Order Boolean Masking Schemes in the Literature

In the rest of this chapter, we focus on $d = 1$ mask. The leakage function for the first-order masking countermeasure can be expressed as:

$$\mathcal{C}_{\text{total}}\left(\vec{L}\right) = \mathcal{C}_{\text{total}}\left(L_0, L_1\right) \ ,$$

where $L_0 = \mathrm{HW}(Z \oplus M) + N_0$ and $L_1 = \mathrm{HW}(M) + N_1$. Two cases are considered:

- **In hardware**: We note "$\mathcal{C}_{\text{device}} = \text{sum}$" to say that the device already sums the leakages, *i.e.* $\mathcal{C}_{\text{device}}(\vec{L}) = L_0 + L_1$;
- **In software**: We note "$\mathcal{C}_{\text{device}} = \text{id}$" to say that the device leaks each leakage identically *i.e.* $\mathcal{C}_{\text{device}}(\vec{L}) = (L_0, L_1)$.

When a first-order Boolean masking is involved to secure the manipulation of Z, the latter variable is randomly split into two shares S_0, S_1 such that:

$$Z = S_0 \oplus S_1 \ . \tag{6.4}$$

The share $S_1 = M$ is usually called the mask and is a random variable uniformly distributed over $\mathbb{F}_2^n$. The share $S_0 = Z \oplus M$ is called the masked variable. Variables Z and M are assumed to be mutually independent. To enable the application of a transformation S on a variable Z split in two shares, as in Eqn. (6.4), a so-called first-order masking scheme must be designed. It leads to the processing of two new shares S'_0 and S'_1 such that:

$$S(Z) = S'_0 \oplus S'_1 \ .$$

Once again, the share $S'_1 = M'$ is usually generated at random (the new mask) and the share S'_0 is defined such that $S'_0 = S(Z) \oplus M'$. The critical point is to deduce $S(Z) \oplus M'$ from the variables M, $Z \oplus M$ and M' without compromising the security of the scheme (*w.r.t.* first-order SCA attack). When S is linear for the law $\oplus$, then deducing $S(Z) \oplus M'$ is an easy task. The variable $S(Z) \oplus M'$ can be simply chosen as the direct sum of the variables M, $Z \oplus M$ and M'. When S is non-linear for the law $\oplus$ (which occurs when S is a S-box), achieving first-order security is much more difficult. The latter security indeed implies that no univariate leakage during the processing leaks information on Z and hence, particular attention must be paid on each elementary calculus or memory manipulation. Several solutions have been proposed to deal with this issue. Commonly, there are three strategies [PR07]: *the re-computation method*, *the global look-up table* and *the S-box secure calculation*.

6.3.1 The Re-Computation Method

This technique involves the computation of a precomputed table corresponding to the masked S-box and the generation of one or several random value(s) [Mes00a, AG01]. In its most elementary version, two random values M and M' are generated and a look-up table representing the function $S' : Y \mapsto S(Y \oplus M) \oplus M'$ is computed from S and stored in RAM. Then, each time the masked variable $S(Z) \oplus M'$ has to be computed from the masked input $Z \oplus M$, the RAM is accessed.

6.3.2 The Global Look-up Table Method

This method also involves the computation of a precomputed look-up table, denoted $T^\star$, associated to the function $S' : (X, Y, Y') \mapsto S(X \oplus Y) \oplus Y'$. To compute the

masked variable $S(Z) \oplus M'$, the global look-up table method (GLUT for short) performs a single operation: the look-up table $T^{\star}[Z \oplus M, M, M']$ [PR07, SVCO+10]. The main and important difference with the first method is that the value $S(X \oplus Y) \oplus Y'$ has been precomputed for every possible 3-tuple of values. Consequently, there is no need to re-compute before each algorithm processing and it can be stored in ROM[4]. In a simplified version (sufficient to thwart only first-order SCA), the output mask and the input mask are chosen equal (*i.e.* $M = M'$). In this case, the dimension of the table is $2n$ instead of $3n$ and the look-up table becomes $T^{\star}[Z \oplus M, M]$.

6.3.3 The S-box Secure Calculation Method

The S-box outputs are computed *on-the-fly* by using a mathematical (*e.*g. polynomial) representation of the S-box [CJRR99, RDJ+01, Tri03, PGA06, RP10]. Then, each time the masked value $S(Z) \oplus M'$ has to be computed, an algorithm performing S and parametrized by the 3-tuple $(Z \oplus M, M, M')$ is executed. The computation of S is split into elementary operations (bit-wise addition, bit-wise multiplication, ...) performed by accessing one or several look-up table(s).

Moreover, depending on the number of masks generated to protect the S-box calculations, we can distinguish two modes of protections:

1. *The single mask protection mode*: in this mode, every computation $S(Z)$ performed during the execution is protected with a single pair of input/output masks (M, M').

2. *The multi-mask protection mode*: in this mode, the pair of masks (M, M') is re-generated each time a computation $S(Z)$ must be protected and thus many times per algorithm execution.

In [PR07], the authors have shown that the choice between the three methods depends on the protection mode in which the algorithm is implemented. In fact, when the algorithm is protected in the single-mask protection mode, the re-computation method is more appropriate and induces a smaller timing/memory overhead. In the multi-mask protection mode, the re-computation method is often much more costly since the re-computation must be done before every S-box processing. Moreover, in both contexts it requires 2^n bytes of RAM to be free, which can be impossible in some very constrained environments. Concerning the S-box secure computation, it is secure against first-order SCA and does not need particular RAM allocation. However, it is often more time consuming than the first two methods and can only be used to secure S-boxes with a simple algebraic structure (as *e.g.* the AES or the SEED S-boxes).

Regarding the GLUT method, it seems at a first glance to be the most appropriate method. Its timing performances are ideal since it requires only one memory transfer. Moreover, it can be applied in both protection modes described above.

[4] Recall that in embedded systems, ROM is a much less costly resource than RAM.

From a security point of view, the GLUT method has however a flaw since it manipulates the masked data $Z \oplus M$ and the mask M at the same time. Actually, $Z \oplus M$ and M are concatenated to address the look-up table $T^\star$ and thus, the value $Z \oplus M \,||\, M$ is transferred through the bus. Since the latter variable is statistically dependent on Z, any leakage on it is potentially exploitable by a first-order DPA involving the higher-order moments of the concatenated random variable. It must however be noted that such a leakage on the address does not necessarily occurs during the bus transfers or the registers update. Indeed, when for instance the latter ones leak the Hamming weight between an independent and random initial state and the address $Z \oplus M \,||\, M$, then the leakage is independent on Z and no first-order DPA is hence applicable. This example shows the importance of the device architecture when assessing a countermeasure soundness.

6.4 Conclusions

This chapter deeply analyses higher-order Boolean masking countermeasures against side channel attacks in contexts where the masks are manipulated simultaneously. The relationship between the leakage characteristics and the HO-CPA attacks efficiency is focused, leading to the introduction of the notion of HO-CPA immunity as a new SCA metric. Then, we described the most widely used first-order masking schemes in the literature. We showed that the problem of securing an algorithm by using masking techniques could be reduced to the problem of securing the S-box computation.

Chapter 7

Security Evaluation of Masking Countermeasures

In this chapter, we are concerned with a formal security evaluation of Boolean hardware masking schemes. Following a practice-oriented evaluation framework introduced by Standaert *et al.* at EUROCRYPT'2009 [SMY09], we compute both leakage and attack metrics. Then, we prove that a leakage metric (namely the mutual information) allows characterizing perfectly the best attack. Moreover, we exhibit explicitly the link between leakage and attacks metrics.

The results presented in this chapter have been published in collaboration with Sylvain Guilley and Jean-Luc Danger in the international symposium on *Hardware-Oriented Security and Trust (HOST 2011)* [MGD11a].

Contents

7.1 On Comparing Side Channel Attacks

In a recent work [MOS11], Mangard *et al.* showed that under some assumptions, the standard univariate side channel attacks using a distance-of-means test, correlation analysis and Gaussian templates are essentially equivalent. So, in the important scenario of standard DPA attacks, all distinguishers are equally efficient. Hence, testing does not always require investigating all distinguishers exhaustively. It is sound to use "one distinguisher for all". Then, they established a link between the

CPA and the MIA in the first-order side channel attack scenario:

$$\mathrm{MIA} = \frac{1}{2}\log_2\left(1+\mathrm{CPA}^2\right) \approx \frac{1}{2\ln 2}\mathrm{CPA}^2 \quad (\text{if } |\mathrm{CPA}| \ll 1)\,.$$

This relationship allows linking currently used metrics to evaluate standard DPA attacks (such as the number of power traces needed to perform a key recovery) with an information theoretic metric (the mutual information).

In [SVCO+10], authors show that in the context of multivariate attacks against masked implementations, understanding second-order attacks requires to carefully investigate the information leakages and the adversaries exploiting these leakages, separately. So, the evaluations of protected implementations should hold in two steps. First, an information theoretic analysis determines the actual information leakage (*i.e.* the impact of the countermeasure, independently of the adversary). Second, a security analysis determines the efficiency of various distinguishers in exploiting this leakage. By applying such a methodology to simulations and practical experiments, we consequently obtain a fair and comprehensive evaluation of the security level that a masking scheme can ensure.

In next sections, we extend the work done in [SVCO+10] so as to conclude about the efficiency of hardware masking implementation. The first topic of interest consists in exhibiting the asymptotical evolution of the mutual information metric (MIM), as described in [SVCO+10], for high value of noise standard deviation and for several leakages combining functions (for instance: the sum, the product and the absolute difference). The second topic of interest is to bring to the fore the link between leakage and security metrics for second-order attacks on masked implementations. The goal is to decide whether the combining function leading to the greatest vulnerabilities identified by the MIM also leads to the best attack. Moreover, we show the relationship between the MIM and the number of traces to reach a certain success rate of non-profiled attack which was experimentally exhibited in [SVCO+10].

7.2 Leakage Estimation with Information Theory

7.2.1 Leakage Metric Computation

In the sequel, we denote the variable L as the combining of the individual leakages of the masked data $L_1 = \mathsf{HW}(Z \oplus M)$ and the mask $L_2 = \mathsf{HW}(M)$. Thus, the leakage can be expressed as $L = \mathcal{C}(L_1, L_2) + N$, where $\mathcal{C}$ is the combining function and $N \sim \mathcal{N}(0, \sigma^2)$ with σ is the noise standard deviation. Assuming a Hamming weight leakage model, the MIM, expressed in bits and defined as $\mathsf{I}[L; \mathsf{HW}(Z)]$, can be written as:

$$\begin{aligned}\mathsf{I}[L;\mathsf{HW}(Z)] \;=\; & -\sum_{h=0}^{n} \mathsf{P}[\mathsf{HW}(Z)=h]\cdot\Big(\log_2 \mathsf{P}[\mathsf{HW}(Z)=h] \\ & -\int_{\mathbb{R}} \mathsf{P}[L=l\,|\,\mathsf{HW}(Z)=h]\cdot\log_2 \mathsf{P}[\mathsf{HW}(Z)=h\,|\,L=l]\,\mathrm{d}l\Big). \quad (7.1)\end{aligned}$$

The MIM is represented in the case of DES (the bitwidth of the sensitive variable under analysis $n = 4$) in Fig. 7.1(a), with the same scale for the noise standard deviation σ as in [SVCO+10], and in Fig. 7.1(b) with a larger range for σ. The overall shape of the leakage metric of Fig. 7.1(a) complies with Fig. 3 page 9 of [SVCO+10]:

- the sum or the difference in absolute value combining functions leak more when the noise is low (σ small, like in "software"), whereas
- the product leaks more otherwise (σ high, like in "hardware", with large amounts of noise due to the intense algorithmic parallelism).

However, in hardware implementations, the only combining function available to the attacker is the sum (that is achieved physically by the global side channel measurements). Thus, the masking is even more relevant in hardware, because the attacker cannot choose the best attacks.

When σ is large, we observe (and will show in Sec. 7.2.1.2) that $\log(\text{MIM}) \approx -4 \cdot \log(\sigma)$ for the sum combining function. Thus, the combining by sum leaks very few information. On the contrary, $\log(\text{MIM}) \approx -2 \cdot \log(\sigma)$ for the absolute difference or the product combining function. Those ways of combining information lead to greatest leaks, and thus better extract the information despite the mask.

7.2.1.1 MIM Asymptotic Value when $\sigma \to +\infty$

When the noise gets high, the conditional distributions tend to become indiscernible. In terms of probabilities, this can be expressed by the independence of L and $\mathsf{HW}(Z)$. Thus, to study the asymptotic values of the MIM (Eqn. (7.1)), we will assume that:

$$\forall l, h, \mathsf{P}[L = l, \mathsf{HW}(Z) = h] = \mathsf{P}[L = l] \times \mathsf{P}[\mathsf{HW}(Z) = h] \ .$$

This implies that:

$$\mathsf{P}[L = l \,|\, \mathsf{HW}(Z) = h] = \frac{\mathsf{P}[L = l, \mathsf{HW}(Z) = h]}{\mathsf{P}[\mathsf{HW}(Z) = h]} = \frac{\mathsf{P}[L = l] \times \cancel{\mathsf{P}[\mathsf{HW}(Z) = h]}}{\cancel{\mathsf{P}[\mathsf{HW}(Z) = h]}} = \mathsf{P}[L = l] \ ,$$

and symmetrically that:

$$\mathsf{P}[\mathsf{HW}(Z) = h \,|\, L = l] = \mathsf{P}[\mathsf{HW}(Z) = h] \ .$$

The asymptotical MIM rewrites as:

$$\begin{aligned} \text{MIM} &= \mathsf{H}[\mathsf{HW}(Z)] - \sum_{h=0}^{n} \mathsf{P}[\mathsf{HW}(Z) = h] \int_{\mathbb{R}} \mathsf{P}[L = l] \cdot \log_2 \mathsf{P}[\mathsf{HW}(Z) = h] \, \mathrm{d}l \\ &= \mathsf{H}[\mathsf{HW}(Z)] - 1 \times \mathsf{H}[\mathsf{HW}(Z)] \\ &= 0 \text{ bit} \ . \end{aligned}$$

However, the first-order dependency of MIM with σ would be more interesting than the limit. This law is computed in the next subsection.

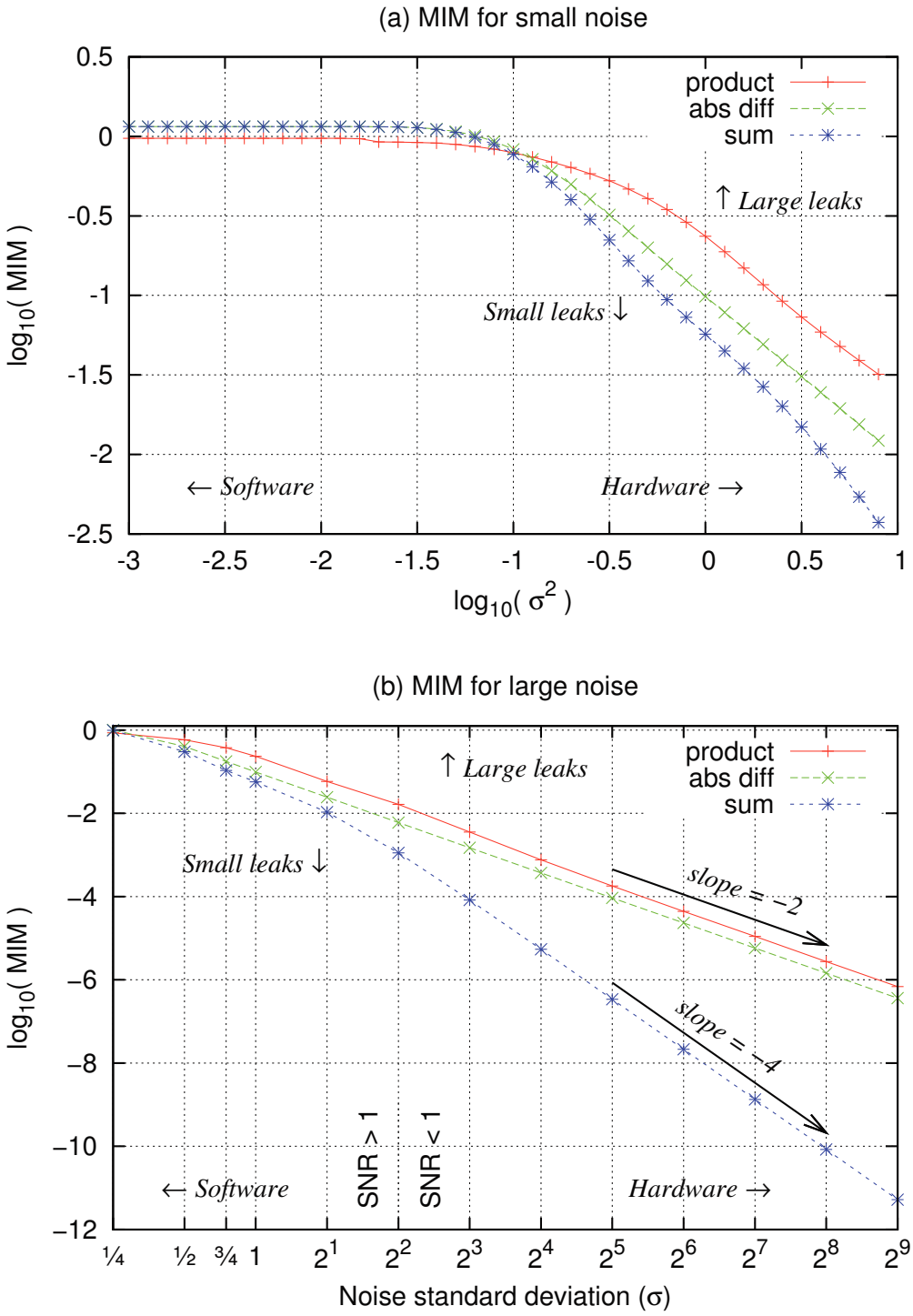


Figure 7.1: Leakage metrics for the 4-bit leakage model of an architecture protected against second-order DPA.

Table 7.1: Mean and variance of the distributions for the studied combining functions on $n = 4$ bits, in the absence of noise.

R.V.	$L \mid \mathsf{HW}(Z)=0$	$L \mid \mathsf{HW}(Z)=1$	$L \mid \mathsf{HW}(Z)=2$	$L \mid \mathsf{HW}(Z)=3$	$L \mid \mathsf{HW}(Z)=4$	L
Centered product combining function						
$\bar{\mu}_{\{h\mid\mathrm{tot}\}}$	5.000	4.500	4.000	3.500	3.000	4.000
$\bar{\sigma}^2_{\{h\mid\mathrm{tot}\}}$	17.500	12.750	8.500	4.750	1.500	9.000
Absolute difference combining function						
$\bar{\mu}_{\{h\mid\mathrm{tot}\}}$	0.000	1.000	1.000	1.500	1.500	1.094
$\bar{\sigma}^2_{\{h\mid\mathrm{tot}\}}$	0.000	0.000	1.000	0.750	1.750	0.804
Sum combining function						
$\bar{\mu}_{\{h\mid\mathrm{tot}\}}$	4.000	4.000	4.000	4.000	4.000	4.000
$\bar{\sigma}^2_{\{h\mid\mathrm{tot}\}}$	4.000	3.000	2.000	1.000	0.000	2.000

7.2.1.2 Analytical Expression of Eqn. (7.1) under the Gaussian Approximation

We assume that σ is large enough to consider that $\forall h$, the RVs $(L \mid \mathsf{HW}(Z) = h)$ and L follow normal distributions. This is obviously not the case for small values of σ. But, it is a viable assumption in an implementation with 64 bits of state computing in parallel with the $n = 4$ targeted ones. We apply the Gaussian assumption by adopting the following assimilations:

- $\forall h, L \mid \mathsf{HW}(Z) = h \sim \mathcal{N}(\mu_h, \sigma_h^2)$ and
- $L \sim \mathcal{N}(\mu_{\mathrm{tot}}, \sigma^2_{\mathrm{tot}})$.

Then, it is straightforward to show that $\mu_{\{h\mid\mathrm{tot}\}} = \bar{\mu}_{\{h\mid\mathrm{tot}\}}$ and $\sigma^2_{\{h\mid\mathrm{tot}\}} = \sigma^2 + \bar{\sigma}^2_{\{h\mid\mathrm{tot}\}}$, where the quantities over-lined are respectively the mean and variance without noise. They are given for $n = 4$ in Tab. 7.1. Therefore, we can derive an analytical formula for the MIM, as a function of the constants presented in Tab. 7.1 and of the noise variance σ^2:

$$
\begin{aligned}
&-\textstyle\sum_{h=0}^{n} \mathsf{P}[\mathsf{HW}(Z)=h] \int_{\mathbb{R}} \mathsf{P}[L=l \mid \mathsf{HW}(Z)=h] \cdot \log_2 \frac{\mathsf{P}[L=l \mid \mathsf{HW}(Z)=h]}{\mathsf{P}[L=l]} \,\mathrm{d}l = \\
&\underbrace{-\sum_{h=0}^{n} \mathsf{P}[\mathsf{HW}(Z)=h] \frac{(\bar{\mu}_h - \bar{\mu}_{\mathrm{tot}})^2}{2\sigma^2 + 2\bar{\sigma}^2_{\mathrm{tot}}}}_{\text{Term1}} + \underbrace{\sum_{h=0}^{n} \frac{\mathsf{P}[\mathsf{HW}(Z)=h]}{2\ln 2} \ln \frac{1 + \bar{\sigma}^2_h/\sigma^2}{1 + \bar{\sigma}^2_{\mathrm{tot}}/\sigma^2}}_{\text{Term2}}.
\end{aligned}
$$

This result is directly deduced from the formula of the Kullback-Leibler divergence for two independent normal laws. We can now derive the expression of the two

terms. First of all:

$$\text{Term1} = \frac{-1/2}{\sigma^2 + \bar{\sigma}_{\text{tot}}^2} \sum_{h=0}^{n} \mathsf{P}[\mathsf{HW}(Z) = h] \left(\bar{\mu}_h - \bar{\mu}_{\text{tot}}\right)^2 \propto \frac{1}{\sigma^2} \ , \tag{7.2}$$

because in the large noise hypothesis, $\sigma \gg \bar{\sigma}_{\text{tot}}$.

In order to evaluate the Term2, we expand a Taylor' series at order two on variable $\varepsilon = \sigma_h/\sigma \ll 1$, using $\ln(1+\varepsilon) = \varepsilon - \frac{1}{2}\varepsilon^2 + o(\varepsilon^2)$. It yields:

$$\text{Term2} \approx \frac{1}{4 \ln 2 \times \sigma^4} \left(\bar{\sigma}_{\text{tot}}^4 - \sum_{h=0}^{n} \mathsf{P}[\mathsf{HW}(Z) = h] \bar{\sigma}_h^4 \right) \propto \frac{1}{\sigma^4} \ . \tag{7.3}$$

For the sum combining function, the Term1 is null, because all the μ_h are the same. More information leaks with the absolute difference and the product, because of the preponderant Term1, that uses the even distribution of μ_h. Thus, MIM $\propto 1/\sigma^4$ for the sum, whereas MIM $\propto 1/\sigma^2$ for the absolute difference and the centered product combinings. The result is in line with Theorem 4 of Chapter 6. In fact, from Tab. 7.1 we can see that for the sum the $\mathsf{HCI} = 2$ whereas $\mathsf{HCI} = 1$ for the absolute difference and the product combinings. This is also consistent with the exact computation of Eqn. (7.1), notably the observed asymptotical behavior of Fig. 7.1(b).

Nonetheless, we insist that Fig. 3 of [SVCO+10] tends to let us thing the law $\log(\text{MIM}) = f(\log(\sigma))$ was simply shifted in the ordinate axis. However, we show that the slopes actually differ.

7.2.2 Discussion

It can be misleading not to study high values of σ. Indeed, for instance, in the article [SVCO+10], it seems (in Fig. 3, where $\sigma \leq \sqrt{10} \ll n$) that the tendency of leakage is the same for the sum and the absolute difference combining functions. However, we show that the later leaks much more.

Apart from this clarification, it remains that the greatest quantity of sensitive information is leaked when combining the samples with one of these two combining functions (absolute difference or centered product). Now, in hardware implementations, only the sum of the samples (done physically because side channel quantities are additive) is available to the attacker. This is a built-in protection of hardware implementations, since the simultaneous processing of the masked data and the mask prevents the attacker from choosing the most adequate combining function.

7.3 Security Estimation with Attacks

The goal of this section is to decide if the combining function leading to the greatest vulnerabilities identified by the MIM in previous section (*i.e.* CPA with combining of product) also leads to the best attack. Moreover, we will study the relationship between the mutual information and the number of traces to reach a certain success rate for correlation attacks.

7.3.1 Success Rate Results

The success rates of correlation and MIA simulated attacks (computed over 100 independent experiments), using different combining functions, are given in Fig. 7.2(a) and Fig. 7.2(b), without and with noise. The 4-bit output of the first DES S-box is targeted. CPA with the sum combining function is not shown, because the attack is perfectly thwarted by the countermeasure. For the MIA attacks, we use histograms for the PDF estimation with a number of bins computed using the Scott's rule.

From these figures, two important consequences can be emphasized:

1. **In noise free model**: The MIA with sum and absolute difference perform well. About 50 observations are required to achieve 100% success rate. Similarly, MIA with product and using joint distribution outperform the attacks using Pearson's correlation coefficient. Finally, we can see that the CPA when combined with the normalized product is better than the CPA when combined with the absolute difference.

2. **With increasing noise**: The presence of noise affects the different distinguishers in a very different manner. The MIA with the sum and absolute difference work worst. About 450 and 550 observations are required respectively to achieve 100% success rate. By contrast, CPA combined with the normalized product works well. MIA using the joint distribution is the worst efficient attack in our simulation. This is caused by the need of more data to estimate the probability distributions.

7.3.2 Security Analysis of Masking

Our experiments are consistent to those studied in [SVCO+10]. Hence, the study of the evolution of these attacks for high values of noise standard deviation is an interesting direction. It allows giving a hint about the number of measurements that are required to break the implementation if we estimate the value of the SNR. In addition, if an attacker has an acquisition campaign, he can conclude on which attack should be performed to find the secret key.

In Fig. 7.3[1], the number of messages needed to achieve a success rate of 90% is recorded for each attack studied before.

The results in Fig. 7.3 can be analysed depending on the noise level:

1. **Undersampling**: When $\sigma \leq 4$, both distinguishers succeed with few number of messages (lower than 10K). The CPA attack outperforms the MIA, whatever the standard deviation of the noise is. In a noise-free context, MIA with joint distribution needs only 35 messages to reach a success rate of 90% whereas CPA when combined with the normalized product needs 50 messages to achieve the same threshold. But, when the standard deviation of noise

[1] We failed to calculate the number of messages to achieve a success rate of 90% for the MIA attacks using different combining for $\sigma > 2^2$ because it exceeds 10^7 (our computational budget).

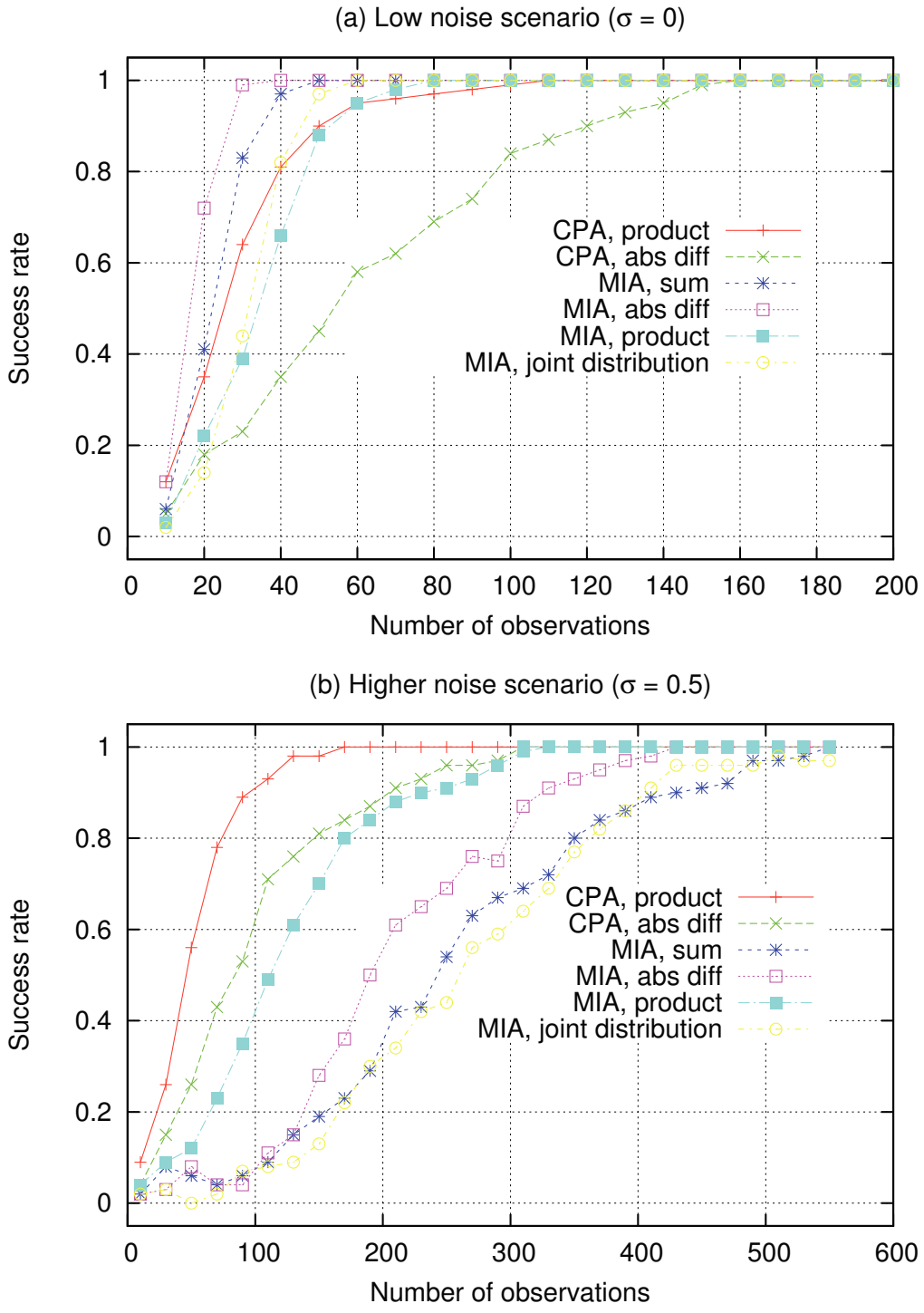


Figure 7.2: Success rate of simulated attacks.

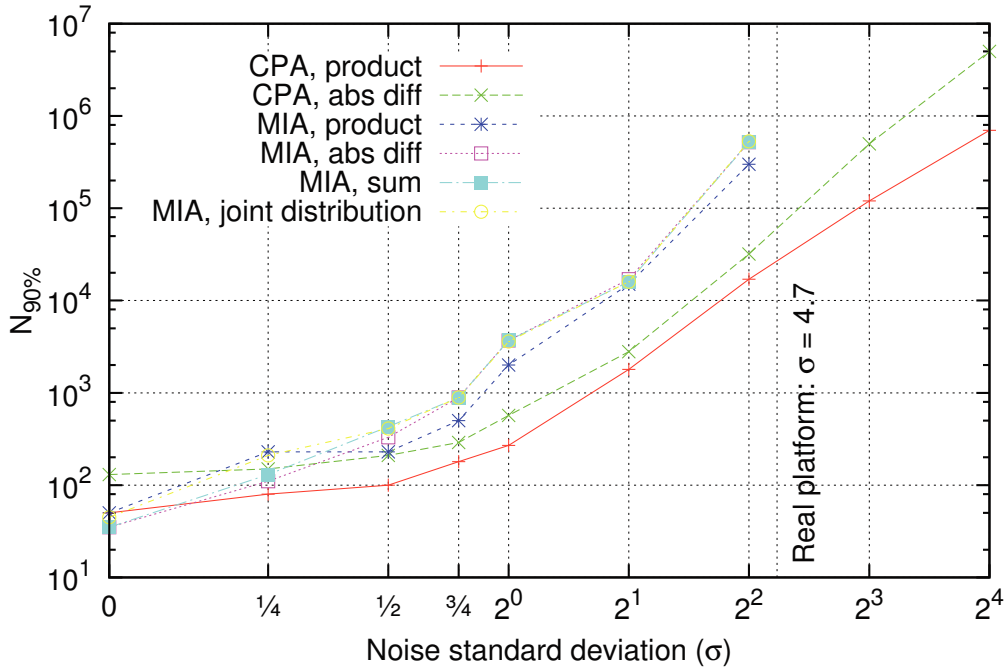


Figure 7.3: Evaluation of $N_{90\%}$, the number of messages to achieve a success rate greater than 90%, according to the noise standard deviation when attacking a Boolean masking scheme.

increases, the CPA combined with the normalized product becomes the most efficient attack. This confirms that Pearson's coefficient is the good tool to measure a linear correlation in software implementations.

2. **Oversampling**: When $\sigma \geq 4$, all the distinguishers need a large number of messages (greater than 10K) to reach 90% success rate. In this case, the curves do not have the same shape for each distinguisher asymptotically. This suggests that the noise differently affects the attacks efficiency. The curves corresponding to MIA using sum and difference absolute combinings and using joint distribution are superimposed. This implies that those attacks share approximately the same efficiency and that none of them is emerging as better candidate than the others. The MIA when combined with the normalized product is better and it requires less observations to achieve 90% of success rate.

   As expected, The CPA attacks are more powerful than the others are (around 30 times more efficient than MIA with joint distribution). Moreover, the CPA combined with the normalized product is more efficient than the CPA combined with the difference absolute (around 5 times more efficient).

In order to exhibit the utility of the previous study, we computed the SNR from our FPGA acquisitions of a masked DES implementation; we obtained $\sigma = 4.7$. From Fig. 7.3, we observe that we can perform the CPA with normalized product combining: it is the attack that requires the less measurements (about 20K traces). For CPA with absolute difference combining, we need about 35K traces. The MIA with joint distribution requires more than 10^6 measurements, which demands more time to be generated and space to be stored. This is probably due to the fact that MIA needs to estimate PDFs on-line. It is thus noisier than CPA for small numbers of traces. Therefore, we will focus on correlations attacks in the sequel.

7.3.3 Analytical Derivation of the Security Level for Correlation Attacks

We showed in Sec. 7.2.1.2 that MIM $\propto 1/\sigma^2$ asymptotically when we focus on the product or the absolute difference combining. From [PRB09], we deduce an approximation of Pearson's correlation coefficient for those combining functions when the bitwidth n is negligible compared to the noise standard deviation σ, and for centered leakages. The correlation in the case of product combining is:

$$\rho_{\mathcal{C}(L_1,L_2)=(L_1-E(L_1))\times(L_2-E(L_2))} \quad \propto \quad \frac{\sqrt{n}}{4\sigma^2} \ , \tag{7.4}$$

and for the absolute difference combining function:

$$\rho_{\mathcal{C}(L_1,L_2)=|L_1-L_2|} \quad \propto \quad \frac{\sqrt{n}}{4\sigma^2\sqrt{2\pi-4}} \ . \tag{7.5}$$

Therefore, we can exhibit the relationship between the MIM and the correlation attack: we have asymptotically $\frac{\text{MIM}}{\text{CPA}} = constant$. We mention that our results differ from that of standard univariate side channel attack (see [MOS11]) where MIM $\propto \frac{1}{2\ln 2}\text{CPA}^2$.

It has been demonstrated that the number of observations required to break a cryptographic implementation by CPA is equal to [Man04]:

$$3 + 8 \times (Z_{1-\alpha})^2 \Bigg/ \left(\ln\left(\frac{1+\rho}{1-\rho}\right)\right)^2 \ , \tag{7.6}$$

where $Z_{1-\alpha}$ is a quantile of a normal distribution for the 2-sided confidence interval with error $1-\alpha$. We can plunge Eqn. (7.4) into Eqn. (7.6). Assuming the number of traces is large and ρ is small, it yields the number of messages to achieve a success rate of 90%, denoted $N_{90\%}$:

$$N_{90\%} \approx 8 \times \left(\frac{Z_{90\%}}{2\rho}\right)^2 \propto 8 \times \left(\frac{Z_{90\%}}{2\frac{\sqrt{n}}{4\sigma^2}}\right)^2 \quad \propto \quad \sigma^4 \ .$$

So we have $\log(N_{90\%}) \propto 2 \cdot \log(\sigma^2)$. Our result coincides with that of Fig. 8 in [SVCO+10] for the slope in the case of first-order masking.

7.3.4 Discussion

In presence of large noise, the most powerful attack (namely CPA with centered product combining, as seen in Sec. 7.3) is based on a combining function that had also been showed to leak the most in Sec. 7.2. In hardware, it thus seems that the MIM is a good indicator of the most relevant attack strategy. So, we can exhibit the unequivocal link between number of messages to achieve a success rate of 90% of correlation attack (*i.e.* $N_{90\%}$) and the mutual information (MIM). These conclusions do not hold for small noises.

The law $N_{90\%} \approx f(\text{MIM})$ conjectured in [SVCO+10, §8.1] was $f : x \to \frac{1}{x}$ for small values of σ. Here, we prove that this law cannot be true over all the noise values. Indeed, $N_{90\%} \approx \frac{1}{\text{MIM}^2}$ when $\sigma \to +\infty$.

7.4 Conclusions

Attacks against masking [SP06] are difficult when the noise level is high, which is typically the case of parallel hardware cryptoprocessors. We showed in this chapter that the ways to catch the most of the leakage (MIM) and to exploit it (CPA) are more relevant for software implementation than hardware, as hardware has more algorithmic noise and is limited to the arithmetic sum of the leakages as combining function. Therefore, for these two important reasons, masking is a countermeasure more efficient in hardware than in software. While not in contradiction with previous results in the field, these investigations reshape the understanding of certain assumptions and extend the results to the case of hardware implementations.

Part IV
New proposed Masking Countermeasures

Chapter 8

First-Order Leakage-Free Masking Countermeasure

In this chapter, we are not concerned with higher-order masking, but devise an optimised Boolean masking scheme when the leakage function is known. Typically, we show that with a first-order masking it is possible to zero the sensible information leaked during the registers update. This countermeasure applies to all devices that leak a function of the distance between consecutive values of internal variables. In particular, we illustrate its practicality on both hardware and software implementations. Moreover, we introduce a framework to evaluate the soundness of the new first-order masking when the leakage slightly deviates from the rules involved to design the countermeasure.

The results presented in this chapter have been published in collaboration with Emmanuel Prouff, Sylvain Guilley and Jean-Luc Danger in the *Cryptographers' Track at RSA conference (CT-RSA 2012)* [MPGD12a].

Contents

8.1 The GLUT Masking Method

In this section we recall the GLUT method described in Chapter 6 Sec. 6.3. Our proposal is to benefit from all seminal assets of this method and to additionally

achieve a better resistance against univariate side channel attacks.

8.1.1 Detailed Description of the GLUT Method

In hardware, GLUT method can be implemented as shown in Fig. 8.1. This figure encompasses the masking scheme already presented in [SRQ06]. For the sake of simplicity, the linear parts, like the expansion (in DES), MixColumns (in AES), *etc,* are not represented. So, without loss of generality, we assume that the S-box S is an (n, n)-function (*i.e.* $S : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$). For instance, using AES, n can be chosen equal to 8 (straightforward tabulation of SubBytes), 4 (with the decomposition of SubBytes in $\mathrm{GF}((2^4)^2)$), or even 2 (using the $\mathrm{GF}(((2^2)^2)^2)$ tower field [SMTM01]). The registers RD and RM contain respectively the masked data $(X \oplus M)$ and the mask (M).

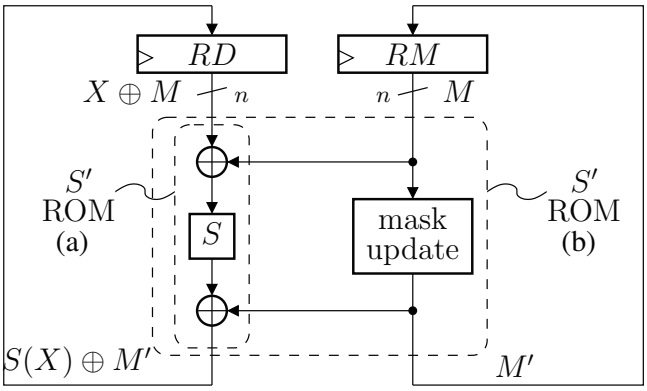


Figure 8.1: First-order hardware masking implementation.

For any (n, n)-function S that must be processed in a secure way, the core principle is to define from S the look-up table representation of a new $(3n, n)$-function S' which is indexed by both the masked data and the masking material. Thanks to this new function, a masked representation $S(X) \oplus M'$ of $X' = S(X)$ is securely derived from $X \oplus M$, M and the output mask M' by accessing the look-up table representing S'. The variables X and X' are the two consecutive values of the sensitive variable. The size of the table can be reduced by defining the output mask as a deterministic function of the input mask. In such a case, the ROM look-up table represents a $(2n, n)$-function S' such that $S'(X \oplus M, M) = S(X) \oplus M'$, where M' is a deterministic function of M (*e.g.* $M' = M \oplus \alpha$ for some constant α).

In the first case, the ROM look-up table has $(3n)$-bit input words: the two shares and the new mask for the remasking, and one n-bit output (*e.g.* option (a) of Fig. 8.1). In the second case, the new masks are derived deterministically from the old ones, and thus the ROM look-up table can have only two inputs (*e.g.* option (b) of Fig. 8.1). The ROM look-up table thus represents a $(2n, 2n)$-function.

8.1.2 Security Analysis of the GLUT Method

During the processing of the scheme depicted in Fig. 8.1, the updating of the registers RD and RM leaks information. We respectively denote by L_{RD} and L_{RM} the variables corresponding to their leakage. We assume that they satisfy:

$$\begin{aligned} L_{RD} &= A(X \oplus M, X' \oplus M') + N_{RD} \ , \\ L_{RM} &= A(M, M') + N_{RM} \ , \end{aligned} \tag{8.1}$$

where A is a deterministic function representing the power consumption during the register updating (*e.g.* A may be the Hamming distance as discussed in [PSDQ05]) and where N_{RD} and N_{RM} are two independent noises. Furthermore, we make the hypothesis that the power consumption related to the simultaneous updating of the registers RD and RM equals $L_{RD} + L_{RM}$ and is denoted by L. We motivate this hypothesis in the following:

Assumption 1. ***(Leakages Adaptivity)*** *During the register update, the global leakage is the sum of the partial leakages.*

In the first part, we will add the following assumption that will be relaxed in the second part of this chapter.

Assumption 2. ***(Adaptivity of Memory States)*** *For any pair (X, Y), we have $A(X, Y) = A(X \oplus Y)$, i.e. A depends only on the distance between its two arguments.*

Under Assumption 1 and 2, the variable L satisfies:

$$L = A(Z \oplus M'') + A(M'') + N_{RD} + N_{RM} \ , \tag{8.2}$$

where Z (respectively M'') denotes $X \oplus X'$ (respectively $M \oplus M'$). Except for very particular definitions of A, the distribution of L (and in particular its variance) depends on the sensitive variable Z. This dependency has already been exploited in several attacks (see *e.g.* [WW04]). In the sequel, we study whether it can be broken by replacing the bit-wise data masking $X \oplus M$ by a new one denoted by $X@M$ and by adding conditions on M and M'.

8.1.3 Towards a New Masking Function

A simple solution, deeply analyzed in this work, is to choose a function @ such that $X \ @ \ M = X \ \oplus \ F(M)$ for some well chosen function F. For such a new masking function, @ is not commutative and M and X do no longer need to have the same dimension n. Only the output size of the function F must be n. In the following, we denote by p the dimension of M and we assume that F is a (p, n)-function, *i.e.* $F : \mathbb{F}_2^p \mapsto \mathbb{F}_2^n$. We will see in Sec. 8.2.1 that p and n must satisfy some conditions for the masking to be sound. In this case, the deterministic part in Eqn. (8.2) can be rewritten:

$$\begin{aligned} A(X@M, X'@M') + A(M, M') &\doteq A(X \oplus X' \oplus F(M) \oplus F(M')) + A(M \oplus M') \\ &= A(Z \oplus F(M) \oplus F(M')) + A(M'') \, . \end{aligned} \tag{8.3}$$

In view of Eqn. (8.3), we deduce the two following sufficient conditions for L to be independent of Z:

1. **[Constant Masks Difference]**: $M \oplus M'$ is constant and
2. **[Difference Uniformity]**: $F(M) \oplus F(M')$ is uniform.

To the two security conditions above, a third one must also be introduced to enable the bit-wise introduction of the key on the internal state X:

3. **[Operations Commutativity]**: For every (X, M, K), we have:

$$X@M \oplus K = (X \oplus K)@M \ .$$

In the following section, we propose a way to specify M, M' and F to satisfy the three sufficient conditions. We structure our study of this new technique in two steps: the first one (*cf.* Sec. 8.2) is performed by assuming that A satisfies Property 2 (*i.e.* $A(X,Y) = A(X \oplus Y)$) and the second one (*cf.* Sec. 8.3) is conducted in an imperfect model where A satisfies $A(X,Y) = P(X,Y)$, with $P(X,Y)$ being a polynomial function in $\mathbb{R}[X_1, \cdots, X_n, Y_1, \cdots, Y_n]$ that does not satisfy Property 2.

8.2 Study in the Idealized Model

8.2.1 Proposed Masking Function

Under Property 2 and as argued in the previous section, we can render the variable L independent of Z. It indeed amounts to fix the condition $M' = M \oplus \alpha$ for some nonzero constant term α and to design a function F *s.t.* the function $Y \mapsto F(Y) \oplus F(Y \oplus \alpha)$ is uniform for this α. The latter function is usually called derivative of F with respect to α, and noted $D_\alpha F$. This notion is for instance defined in Sec. 8.2.2 of [Car10b]. The construction of functions F having such uniform derivatives has been highly investigated in the literature [Car10b, Chp. 4]. We give hereafter two examples of construction of such functions F.

8.2.1.1 First Construction Proposal

We choose $p = n + 1$ and we split $\mathbb{F}_2^{n+1}$ into the direct sum $E \oplus (E \oplus \alpha)$, where E is a n-dimensional vector space and $\alpha \in \mathbb{F}_2^p - E$. One bijective function G from E into $\mathbb{F}_2^n$ is arbitrarily chosen and F is defined such that for every $Y \in \mathbb{F}_2^{n+1}$, we have $F(Y) = G(Y)$ if $Y \in E$ and $F(Y) = 0$ otherwise.

8.2.1.2 Second Construction Proposal

We choose $p = n+n'$ with $n' < n$ and we select one injective function G from $\mathbb{F}_2^{n'}$ into $\mathbb{F}_2^n - \{0\}$. Then, for every $(X, Y) \in \mathbb{F}_2^{n'} \times \mathbb{F}_2^n = \mathbb{F}_2^p$ we define $F(X,Y) = G(X) \cdot Y$, with "$\cdot$" is the field product over $\mathbb{F}_2^n$. The outputs of the (p,n)-function F are uniformly distributed over $\mathbb{F}_2^n$ (since the functions $Y \mapsto G(X) \cdot Y$ are linear and non-zero for

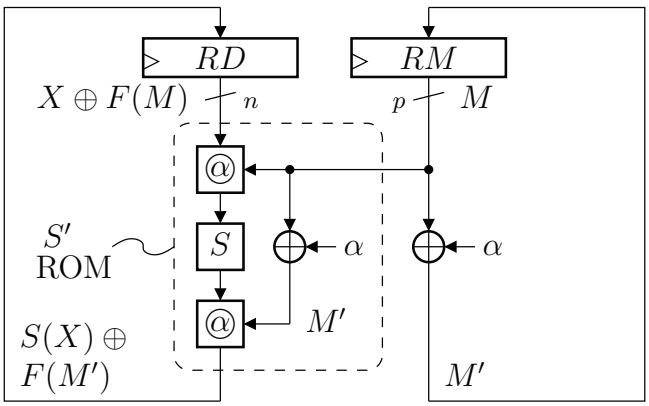


Figure 8.2: Leakage-free masking hardware implementation.

every X). Moreover, for every non-zero element α' in $\mathbb{F}_2^{n'}$, the function $D_\alpha F$ defined with respect to $\alpha = (\alpha', 0) \in \mathbb{F}_2^{n'} \times \mathbb{F}_2^n$ is also balanced. In fact, its outputs are uniformly distributed over $\mathbb{F}_2^n$ when its inputs are uniformly distributed over $\mathbb{F}_2^{n'}$. Indeed, we have $D_\alpha F = (G(X) \oplus G(X + \alpha')) \cdot Y$ and, since the injectivity of G implies that $G(X) \oplus G(X \oplus \alpha')$ is never zero, the functions $Y \mapsto (G(X) \oplus G(X \oplus \alpha')) \cdot Y$ are linear and non-constant for every X.

The two constructions of F satisfy the *difference uniformity* condition defined in Sec. 8.1.3. The mask dimension p for the first construction is only slightly greater than the dimension n of the data to be masked. This makes it more efficient than the second construction. However, the second construction ensures that not only $D_\alpha F$ but also F is balanced. This is not mandatory to ensure the security of the countermeasure in our context where the targeted leakage is assumed to satisfy Property 2, but it can be of interest if one wishes that the data X and X' be masked with a uniform mask $F(M)$ and $F(M')$ respectively.

Figure 8.2 shows a hardware implementation of our countermeasure. The registers RD and RM contain respectively the masked variable $X \oplus F(M)$ and the mask M. The mask update operation is constrained to be a $\oplus$ operation with a constant value α in order to satisfy the first condition. Consequently, every computation in the algorithm is protected with the single pair of masks (M, $M' = M \oplus \alpha$). Nonetheless, the value of M changes at every computation; thus, the injected entropy in one computation is p bits. The look-up table representing the function $S' : (X, Y) \mapsto S(X@Y)@(Y \oplus \alpha) = S(X \oplus F(Y)) \oplus F(Y \oplus \alpha)$ has been precomputed and stored in ROM. The new masked variable $S(X) \oplus F(M')$ is obtained by accessing the ROM table as described in Fig. 8.2. We assume that this addressing is not leaking sensitive information but the leakage comes from the updating of the registers RD and RM following Eqn. (8.2) and (8.3).

8.2.2 Security Evaluation

In our security analysis, we assume that the attacker can query the targeted cryptographic primitive with an arbitrary number of plaintexts and obtain the corresponding physical observations, but cannot choose its queries in function of the previously obtained observations (such a model is called *non-adaptive known plaintext model* in [SMY09]). We also assume that the attacker has access to the power consumption and electromagnetic emanations of the device and applies any univariate SCA attack but is not able to perform a multivariate one.

Regarding the leakage model, we assume that the device leaks a function A of the distance between the processed data and its initial state handled in the register (*i.e.* A satisfies Property 2). This situation is more general than the Hamming distance model, and notably encompasses the imperfect model studied in [VCS09, Sec. 4]. The mutual information $\mathsf{I}[A(Z \oplus F(M) \oplus F(M')) + A(M''); Z] = 0$ since M'' is constant and since $F(M) \oplus F(M')$ is uniformly distributed over $\mathbb{F}_2^n$ and independent of Z. Hence, our construction is *leakage-free* and immune against univariate attacks of all orders. Furthermore, as $A(M'')$ is constant, the mutual information

$$\mathsf{I}[\big(A(Z \oplus F(M) \oplus F(M')), A(M'')\big); Z]$$

is also null, which means that the masking countermeasure is secure against an adversary who observes the leakage in the transition from one state during the registers update and can repeat this as many times as he wants. The adversary recovers the observations of the variable $(L_{RD} + L_{RM})$ and can make all the treatments he wants (*e.g.* computation of mutual information, raise to any power the variable $L_{RD} + L_{RM}, \ldots$).

Recently in the international conference on *Cryptographic Hardware and Embedded Systems* (CHES 2012), Amir Moradi and Oliver Mischke [MM12b] have evaluated the security of our leakage-free countermeasure in a real-world context (implemented on the SASEBO-GII evaluation board) using the so-called *correlation-collision attacks* [MME10]. They demonstrate the high efficiency of our proposal when assuming a Hamming distance leakage model. In fact, the leakage-free countermeasure remains secure against first- and second-order correlation-collision attacks. However, when assuming a S-box input model (*i.e.* targeting the inputs of the S-box rather than the register update) our countermeasure shows vulnerabilities against correlation-collision attacks when using one million of power consumption traces. Nonetheless, we insist that our leakage-free countermeasure was not devised to withstand such an attack. Since the results of [MME10] are not analysed precisely enough, we cannot decide whether the implementation or the CM is responsible for the observed leakage. We study the impact of the leakage model deviations on our countermeasure in Sec. 8.3.

8.2.3 Application to the Software Implementation Case

Our proposal can be applied also in some particular software implementations. In some access memory schemes, the address and the read value are transferred through

the same bus (*e.g.* Von-Neumann architecture). Thus, when accessing a table, the value overwrites the address and a leakage as in Eqn. (8.2) occurs. Such access is obtained with a code such as:

```
mov  dptr, #tab
mov  acc, y
movc acc, @acc+dptr
```

In the code above, `dptr` refers to a data memory pointer and `#tab` to the address of a table stored in data. The variable `y` is assumed to contain the index of the value that must be read in table `tab`. After the third step, the accumulator register `acc` contains the value `tab[y]`. During this processing, the accumulator goes from state `y` to state `tab[y]`. Let us now assume that `tab` refers to the look-up table defined in Sec. 8.2.1 and that `y` refers to the variable $(X@M, M)$. If we associate the most significant bits of the accumulator `acc` to a (sub-)register RD and its least significant bits to a (sub-)register RM then we are in the same context as the analysis conducted in Secs. 8.2.1 and 8.2.2. A first-order DPA attack can be conducted on this register to reveal information about the sensitive data. Taking advantage from our proposal, the memory access is made completely secure under the assumption of Property 2.

8.3 Study in the Imperfect Model

8.3.1 Description

In this section we assume that the hardware has been protected under the assumption that A satisfies Property 2, while the assumption is wrong. Namely, A was assumed to be *s.t.* $A(X, Y) = A(X \oplus Y)$ whereas in reality, it is a polynomial that does not satisfy this property. In fact, the Hamming distance leakage model is in practice an idealization of the reality. Indeed, the assumption that all the bits leak identically, and without interfering, does not hold in real hardware [VCS09]. Also, it has been shown that with specific side channel capturing systems the attacker can distort the measurement. For instance, in [PSQ07], the authors show that with a home-made magnetic coil probing the circuit at a crucial location, the rising edges can be forced to dissipate 17% more than the falling edges. Therefore, we study how the CM is resilient to imperfections of the leakage model.

To do so, we define a general model that depends on random variables. The variability is quantified in units of the side channel dissipation of a bit-flip. The model is affected by small imperfections (due to process variation, or small parasitic cross-coupling) when the variability is about 10%. We also consider the 20% case, that would reflect a distortion of the leakage due to measurements in weird conditions. Eventually, the cases of a 50% and of a 100% deviation indicate that the designer has few or no a prior knowledge about the device leakage's model.

More precisely, we assume that the leakage model satisfies $A(X, Y) = P(X, Y)$, with $P(X, Y)$ being a multivariate polynomial function in $\mathbb{R}[X_1, \cdots, X_n, Y_1, \cdots, Y_n]$

of degree less or equal to $\tau \in [\![1, 2n]\!]$, where X_i and Y_i denote the i^{th} Boolean coordinate of X and Y respectively. We recall that a polynomial of degree τ in $\mathbb{R}[X_1, \cdots, X_n, Y_1, \cdots, Y_n]$ takes the following form:

$$P(X_1, \cdots, X_n, Y_1, \cdots, Y_n) \doteq \sum_{\substack{(u,v)\in\mathbb{F}_2^n\times\mathbb{F}_2^n,\\ \mathsf{HW}(u)+\mathsf{HW}(v)\leq\tau}} C_{(u,v)} \cdot \prod_{i=1}^{n} X_i^{u_i} Y_i^{v_i} \ , \tag{8.4}$$

where the $C_{(u,v)}$ are real coefficients. For example, it is shown in [PRB09, Eqn. (3)] that $P(X_1, \cdots, X_n, Y_1, \cdots, Y_n)$ is equal to $\mathsf{HW}(X \oplus Y)$ when the coefficients $C_{(u,v)} \doteq c_{(u,v)}$ satisfy:

$$c_{(u,v)} = \begin{cases} +1 & \text{if } \mathsf{HW}(u) + \mathsf{HW}(v) = 1\,, \\ -2 & \text{if } \mathsf{HW}(u) = 1 \text{ and } v = u\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{8.5}$$

In the following experiments, we study experimentally the amount of information that the leakage L defined in Eqn. (8.2) leaks on Z when $\tau = \{2, 3\}$ and when the coefficients $C_{(u,v)}$ deviate randomly from those of Eqn. (8.5). More precisely, $C_{(u,v)}$ are respectively drawn at random from this law:

$$C_{(u,v)} \sim c_{(u,v)} + \mathcal{U}(\left[-\tfrac{\delta}{2}, +\tfrac{\delta}{2}\right]) \ . \tag{8.6}$$

The randomness lays in the uniform law $\mathcal{U}(\left[-\frac{\delta}{2}, +\frac{\delta}{2}\right])$, that we parametrize by the deviation $\delta \in \{0.1, 0.2, 0.5, 1.0\}$. The computed mutual information is $\mathsf{I}[L; Z]$, where $L = P(X \oplus F(M), X' \oplus F(M \oplus \alpha)) + P(M, M \oplus \alpha) + N$ and $N = N_{RD} + N_{RM}$.

8.3.2 Simulation Parameters

We assume that F has been designed thanks to the first construction presented in Sec. 8.2.1. Hence, it is a function from $\mathbb{F}_2^{n+1}$ into $\mathbb{F}_2^n$. The mask M and the constant α are of dimension $n + 1$, whereas X is n-bit long.

The mutual information $\mathsf{I}[L; Z]$ is computed for:

- a Gaussian noise N of standard deviation σ varying in $]0, 5]$,
- $n = 3$ bits (to speed up the computations),
- $E = \{0\} \times \mathbb{F}_2^n \subset \mathbb{F}_2^{n+1}$ and the constant α is equal to the binary word 1000, and
- $F(x_3x_2x_1x_0) = 0$ if $x_3 = 1$ or $x_2x_1x_0$ otherwise.

For each experiment, we also compute the mutual information for the straightforward CM of the state-of-the-art (implementation of [SRQ06] represented in Fig. 8.1). We also give the mutual information of this CM if the model is exactly the Hamming distance, and indicate the corresponding leakage without any countermeasure. We recall that, still with a perfect model, the mutual information for our countermeasure with Z is null, whatever sigma is.

8.3.3 Simulation Results

The results are shown in Tab. 8.1. It appears that the degree τ has minor influence on the leakage. The major factor is the deviation from the Hamming distance model. As expected, for low deviations (much smaller than 1, *e.g.* 10% or 20%), the leakage-free countermeasure of Fig. 8.2 definitely outperforms the CM of Fig. 8.1. However, in the presence of deviations close to the unity, the state-of-the-art CM remains the best. In this case, the proposed countermeasure still leaks less than an unprotected design. Nonetheless, we insist that this situation is unlikely, as the deviations from the assumed Hamming distance model is of the order of one bit flip. This means that the designer has a very poor knowledge of the technology as he applies the countermeasure without checking the assumption (Property 2).

Eventually, it is noteworthy that state-of-the-art CM is even slightly improved by the imperfection of the leakage function A. This reflects the fact that the random variable $\mathsf{HW}(Z \oplus M'') + \mathsf{HW}(M'')$ do carry a lot of information on Z, and the noise help reduce the dependency (and thus favors the defender). Also, both CMs are equivalent for an intermediate deviation of 50%. As this value is already quite large, we can conclude that our countermeasure is relevant even if the assumptions on the hardware leakage are extremely approximate.

8.4 Conclusions

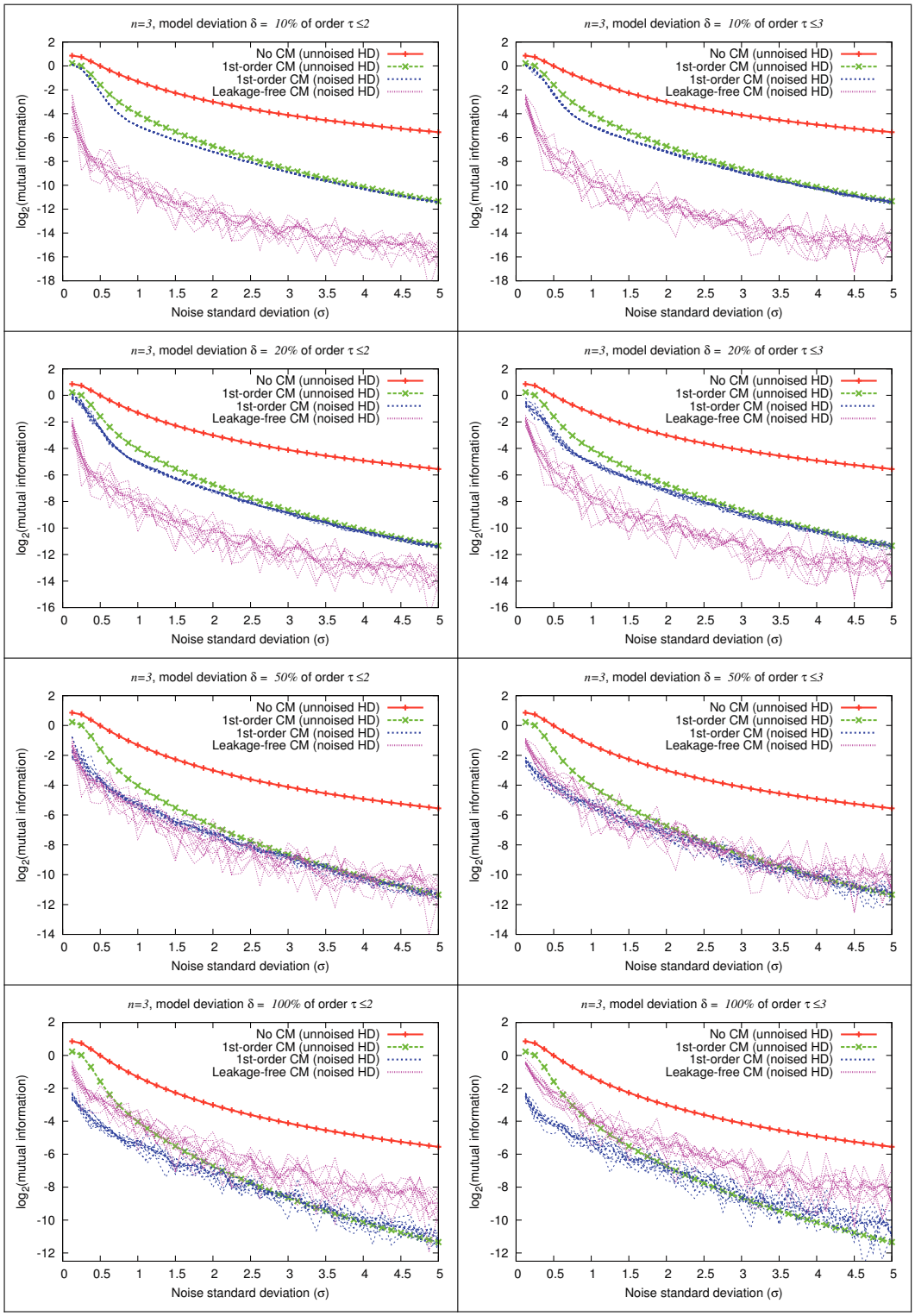
In this chapter, we presented a new masking scheme for hardware S-box implementations. We argued that our proposal is a leakage-free countermeasure under some realistic assumptions about the device architecture. The solution has been evaluated within an information-theoretic study, proving its security against univariate SCA attacks under the Hamming distance assumption. When the leakage function deviates slightly from this assumption (by a few tens of percent), our solution still achieves excellent results. However, if the model is very noisy (the model deviates from the Hamming distance by $\approx 50\%$), then our countermeasure remains all the same as good as state-of-the-art countermeasure.

It has been underlined (in the second construction) that some functions F have a balanced derivative in more than one direction $\alpha \neq 0$. As a perspective, we mention that this feature can be taken advantage of to increase the security of the countermeasure. Indeed, in the perfect model, the leakage remains null. However, using many "αs" certainly helps to counter model imperfections, thus reducing the leakage in this case.

Also, we underline that the proposed countermeasure can be adapted to the hypothetical case where the perfect model is not the Hamming distance $A(X, Y) = \mathsf{HW}(X \oplus Y)$, but is asymmetrical in rising and falling edges (*e.g.* $A(X, Y) = \mathsf{HW}(X \cdot \neg Y)$). Such leakages can be found in near-field electromagnetic measurements (refer to: [PSQ07] or [SGD+09, Fig. 4, left]).

In the next chapter, we propose an enhancement version of the leakage-free countermeasure.

Table 8.1: Leakage comparison of one state-of-the-art CM and our proposed CM in the imperfect HD leakage model.



Chapter 9

Register Leakage Masking Using Gray Counter

In this chapter, we propose a new way to apply masking to secure hardware implementations of block ciphers. The new countermeasure is highly inspired from the leakage-free masking scheme presented in the previous chapter. The main advantage of the new proposed solution is that not only the masked variables and the masking material can be manipulated simultaneously without leaking sensitive information in the Hamming distance model, but also they have the same bitwidth. This should reduce the memory requirements. Moreover, we show that the leaking information stays very limited when the deterministic part of the real leakage slightly deviates from the Hamming distance. Finally, we apply our method to protect an AES hardware implementation and we show that the performances are suitable for practical implementations.

The results presented in this chapter have been published in collaboration with Emmanuel Prouff, Sylvain Guilley and Jean-Luc Danger in the international symposium on *Hardware-Oriented Security and Trust (HOST 2012)* [MPGD12b].

Contents

9.1 Core Principle, Existing Works and Novelties

In this section, we recall the leakage of the first-order masking scheme related to the simultaneous updating of the register RD and RM (*i.e.* register of the masked

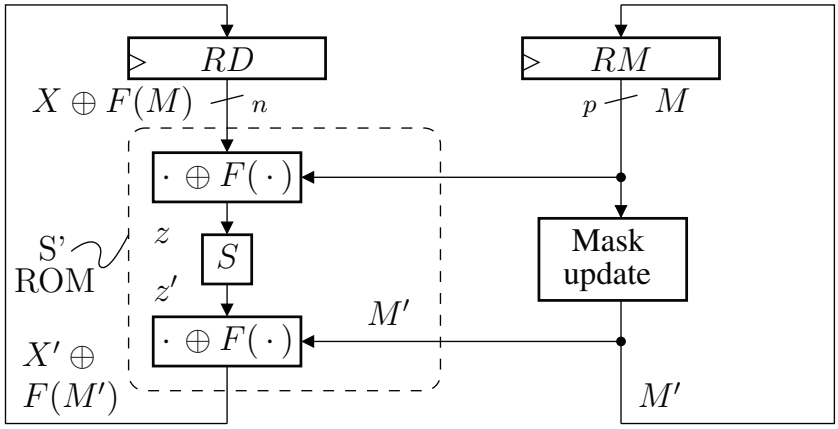


Figure 9.1: Leakage-free masking implementation.

data and the register of the mask) depicted in Fig. 8.1 of Chapter 8. The leakage L can be expressed as:

$$L = A(X \oplus M, X' \oplus M') + A(M, M') + N_{RD} + N_{RM} \ , \tag{9.1}$$

where A is a deterministic function representing the power consumption during the register updating and where N_{RD} and N_{RM} are two independent noises. Furthermore, when A satisfies $A(X, Y) = A(X \oplus Y)$ for any pair (X, Y), then the leakage can be rewritten as:

$$L = A(Z \oplus M'') + A(M'') + N_{RD} + N_{RM} \ , \tag{9.2}$$

where Z (respectively M'') denotes $X \oplus X'$ (respectively $M \oplus M'$). Since the leakage variable L is statistically dependent on Z, any leakage on it is potentially exploitable by an univariate SCA involving its higher-order moments (and in particular its variance [MDFG09]).

In order to balance the distributions of the leakage L, we suggest in the leakage-free countermeasure (Chapter 8) to involve a function @ such that $X \ @ \ M = X \ \oplus \ F(M)$ for some well chosen function F from $\mathbb{F}_2^p$ into $\mathbb{F}_2^n$ with $p \geq n$ (see Fig. 9.1). For such a masking, the deterministic part in Eqn. (9.2) can be rewritten:

$$A(Z \oplus F(M) \oplus F(M')) + A(M'') \, . \tag{9.3}$$

Based on Eqn. (9.3), we deduce two sufficient conditions for the leakage to be independent of Z:

1. **[Constant Mask Difference]**: $M \oplus M'$ is constant,
2. **[Difference Uniformity]**: $Y = F(M) \oplus F(M')$ is uniform.

To fulfill the two conditions above, we fix the condition $M' = M \oplus \alpha$ for some nonzero constant term α and design a function F *s.t.* $Y \mapsto F(Y) \oplus F(Y \oplus \alpha)$ is uniform for this α.

Despite its advantages (perfect first-order security in the Hamming distance model, efficiency, simplicity, *etc.*), the leakage-free countermeasure has two drawbacks. First, all the computations during the algorithm processing are protected with the single pair of masks (M, $M' = M \oplus \alpha$). This property has no impact on the first-order security but potentially weakens the implementation resistance with respect to higher-order SCA (and especially the second-order ones). A second issue with the leakage-free countermeasure is that the input and output dimensions p and n of a function F satisfying the two sufficient conditions cannot be equal. Essentially because for every permutation F on $\mathbb{F}_2^n$ and every constant $\alpha \in \mathbb{F}_2^n$, the function $x \mapsto F(x) \oplus F(x \oplus \alpha)$ is not bijective. This implies that the mask bitwidth p is strictly greater than the bitwidth n of the data to be masked, which induces a time/memory overhead compared to classical first-order Boolean maskings.

In the following section, we present an alternative to the leakage-free CM which does not suffer from the two drawbacks cited above. More precisely, thanks to the so-called *Gray code*, we will show that it is possible to define a masking scheme which shares almost all the good properties of the leakage-free solution with the additional asset that the mask and the masked data have the same size ($p = n$). This makes the new countermeasure an optimal one in terms of randomness consumption per datum to mask. Moreover, we will show that this strategy enables more variability of the mask values used during the overall algorithm processing which strengthens the countermeasure resistance against higher-order SCA. For such a purpose, we however have to relax the first condition which is replaced by the following new constraint:

1. **[Constant Mask Difference Weight]**: $\mathsf{HW}(M \oplus M')$ is constant.

If the device is leaking in the Hamming distance model, we will argue that this replacement has no impact on the implementation security w.r.t. first-order SCA.

9.2 A New masking Function

We structure our study of the new proposal in two steps as in the previous chapter: the first one is performed by assuming that A satisfies the property $A(X, Y) = A(X \oplus Y)$ for any pair (X, Y) and the second one is conducted in an imperfect model where A deviates from the Hamming distance model.

9.2.1 Introduction of the New Proposal

In order to ensure that $\mathsf{HW}(M \oplus M')$ is constant between two consecutive iterations of the algorithm loop in Fig. 9.1, we propose to define M' as the successor of M in the Gray counter. Indeed, as a property of this code, we have $\mathsf{HW}(M \oplus M') = 1$. In the following, we denote by $M^{(i)}$ the value of M at the i^{th} application of an

n-bit Gray code (*i.e.* for the i^{th} iteration of the loop in Fig. 9.1). For example, we have $M^{(0)} = M$, $M^{(1)} = M'$, $M^{(2)} = (M')'$, *etc.* As a Gray code is a bijective (n, n)-function, we have $M^{(2^n)} = M^{(0)}$. Moreover, all the $M^{(i)}$, $0 \leq i < 2^n$, are different. With this specification of the relationship between two consecutive mask values, we ensure that the scheme satisfies the first *Constant Mask Difference Weight* condition. In order to finalize the countermeasure specification, it remains to define a function F such that the *Difference Uniformity* is satisfied when M and M' are two consecutive states in a Gray code. Actually, we will see that there is a huge amount of possible choices for F.

Let us first denote by i the integer value corresponding to the binary representation of an element in $\mathbb{F}_2^n$ (by construction i ranges over all integer values in $[\![0, 2^{n-1}]\!]$). For a function F to satisfy the second condition of Sec. 9.1 (*Difference Uniformity*), there must exist a permutation G on $[\![0, 2^{n-1}]\!]$ such that F is a solution of the following system S_G:

$$S_G : \begin{cases} F(M^{(0)}) \oplus F(M^{(1)}) &= G(0)\,, \\ F(M^{(1)}) \oplus F(M^{(2)}) &= G(1)\,, \\ \cdots \quad \vdots & \quad \cdots \\ F(M^{(2^n-1)}) \oplus F(M^{(0)}) &= G(2^n-1)\,. \end{cases}$$

Proposition 1. *For every permutation G on $[\![0, 2^n - 1]\!]$, there exists 2^n function F that satisfy S_G.*

Proof. As the values $M^{(i)}$ fill $\mathbb{F}_2^n$, if we choose one value of $F(M^{(0)})$ (amongst the 2^n possibilities), then S_G has one unique solution, since $\forall i > 0$, $F(M^{(i+1)}) = F(M^{(i)}) \oplus G(i)$. □

In view of this proposition, there exists a huge number of functions to select a candidate F satisfying the *Difference Uniformity* condition. A construction strategy to define one of them may simply consist in randomly generating a permutation G and a value for $F(M^{(0)})$. Then, the function F is designed by solving the system S_G. This construction of F is very efficient (quadratic in 2^n for small n) and must be done only once for an implementation (the same function F can be used for all the executions of the implementation).

9.2.2 Analysis of the New Proposal in the Idealized Model

In our security analysis, we assume that the attacker has access to the power consumption of the device and applies HO-SCA attacks. Regarding the leakage model, we assume that the device leaks the Hamming distance between two consecutive values handled in the register. Since our proposed masking scheme satisfies that $F(M) \oplus F(M')$ is uniformly distributed over $\mathbb{F}_2^n$ and independent of Z, the mutual information $\mathsf{I}[A(Z \oplus F(M) \oplus F(M')) + A(M''); Z]$ is zero. Hence, our construction is leakage-free and immune against univariate SCA attacks.

9.2.3 Study in the Imperfect Model

In this section, we study how the countermeasure is resilient to imperfections of the leakage model. Namely, we quantify the amount of information leaked when the leakage function A is a multivariate polynomial $P(X_1, \cdots, X_n, Y_1, \ldots, Y_n)$ that does not satisfy the property $A(X, Y) = A(X \oplus Y)$ as detailed in Chapter 8 Sec. 8.3. In the following experiments, we compute the mutual information between L and Z when the degree of the multivariate polynomial of Eqn. (8.4) is equal to 2 (*i.e.* $\tau = 2$) and when its coefficients $C_{(u,v)}$ are drawn at random from the law defined in Eqn. (8.6). Three bit variables are considered in our simulation. The results are shown in Tab. 9.1. They represent the leakage for:

- an unprotected device (in red color),
- a straightforward masking, as in Fig. 8.1 (in green color),
- our CMs (in purple color), with an imperfect model (otherwise the leakage is null) of order $\tau = 2$.

It appears that the leakage-free CM and the one using the Gray counter have comparable values. This was indeed expected, as both use the same amount of masking entropy (after the application of F), namely n bits. They remain unambiguously better than the straightforward first-order masking when the deviation $\delta < 0.5$. Now, this order of magnitude is already quite large. A deviation $\delta \geq 0.5$ is more likely to be due to the CM designer assuming a wrong leakage model than the hardware behaving strangely. In addition, the designer is able to reproduce *in silico* those characterizations, and can thus notice that the model was indeed 50% or more erroneous. Consequently, the characterizations presented in Tab. 9.1 confirm that a large diminution of leakage can be obtained with this CM.

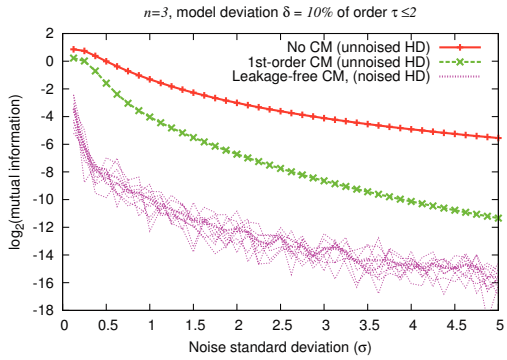
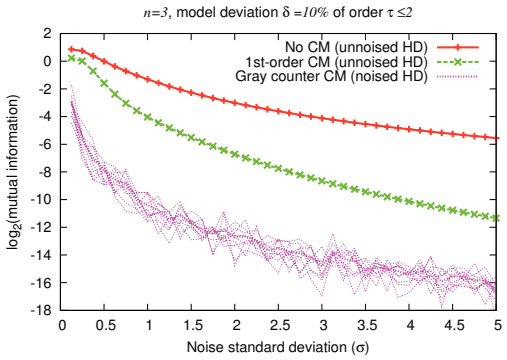
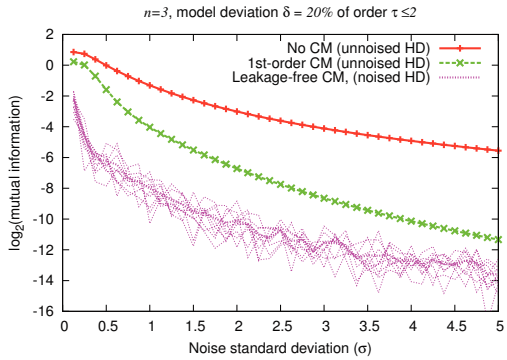
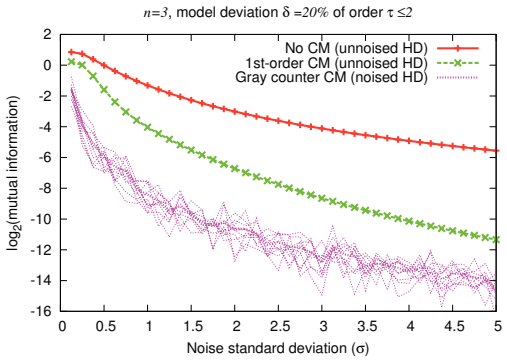
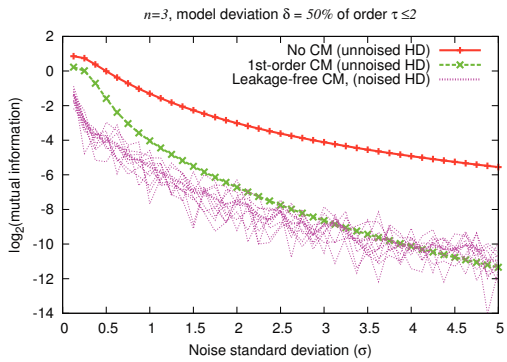
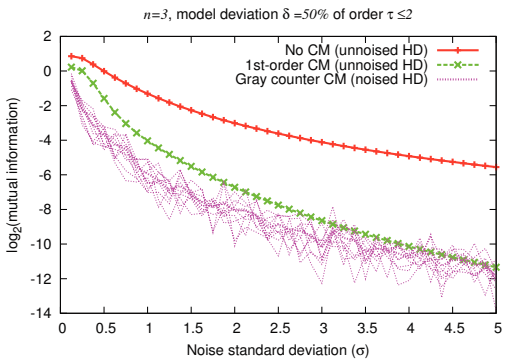
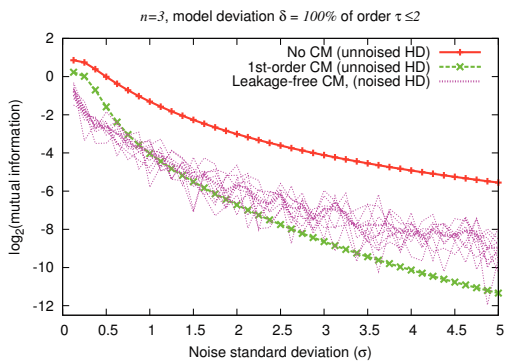
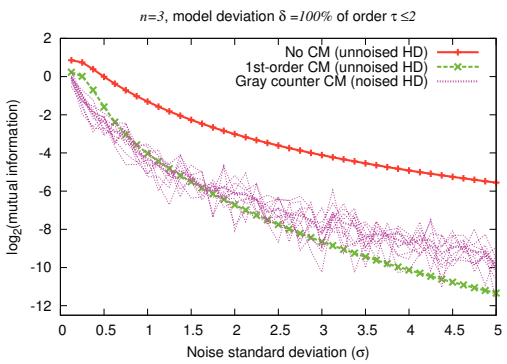
9.3 Hardware Implementation of the Countermeasure

9.3.1 New Masking Architecture

In this section, we apply our proposal to the AES block cipher. The registers RD and RM contain respectively the masked variable $X \oplus F(M)$ and the mask M. We assume that the leakage comes mainly from the activity of these registers, following Eqn. (9.2) and (9.3). The mask M is considered Gray encoded, thus the mask update to M' is an incrementation such as $\mathsf{HW}(M \oplus M') = 1$ which can be pre-computed in ROM or synthesized in logic gates.

A first implementation would be to use the GLUT architecture as shown in Fig. 9.1 with a look-up table representing the function $S' : (X, Y, Y') \mapsto S(X \oplus F(Y)) \oplus F(Y')$ being pre-computed and stored in ROM. However, this would involve huge ROM look-up tables (*e.g.* of size 2^{24} for $n = 8$). In our case, M' can be deduced from M (it is the successor of M in the Gray code): thus, the table input size can

Table 9.1: Leakage comparison of the leakage-free CM (*left column*) and of the one using the Gray counter (*right column*) in the imperfect HD leakage model.

Leakage-free CM	Gray counter CM
	
	
	
	

be reduced from $3n$ to $2n$ bits. However, the ROM remains quite large: it contains 2^{16} bytes for $n = 8$.

In a second implementation, we eliminate the ROM with $2n$ inputs by resorting to a more simple structure called *Universal S-box Masking* (USM) presented in [MDFG09]. In this architecture, the input and output XOR with respectively $F(M)$ and $F(M')$ of the GLUT S' give place to ROMs surrounding the S-box S. Each table uses external encodings with bijections B_1 and B_2 (*e.g.* a simple XOR operation with a constant value) in order to avoid nets with unmasked variables, for instance:

$$\underbrace{S(X \oplus F(Y)) \oplus F(Y')}_{\text{ROM}} = \underbrace{F(Y') \oplus B_2^{-1}}_{\text{ROM \#3}} \circ \underbrace{B_2 \circ S \circ B_1^{-1}}_{\text{ROM \#2}} \circ \underbrace{B_1(X \oplus F(Y))}_{\text{ROM \#1}} .$$

Figure 9.2 illustrates the mask path of AES with the USM implementation taking advantage of the Gray code encoding with the use of three n inputs ROMs. The linear part (ShiftRows and MixColumns) inserted before the third ROM allows to remove the linear computation of the mask. In order to reduce the memory size, B_1 and B_2 are split in two 4-bit bijections. So, memories #1 and #3 could be implemented in two 256×4 ROMs as shown in Fig. 9.2. Hence, the total ROM complexity is 3×2^8 bytes for the masked variable path and 2^8 bytes for the mask path: thus 2^{10} bytes.

If the bijections B_1 and B_2 are public, an attacker can target the variables between the tables. Thus, in order to ensure an optimal security and remove all possible leakage, we propose hereafter three solutions to avoid this issue.

9.3.1.1 Time-Security Trade-off

The first idea is to pause the encryption every once in awhile, in order to compute a new sets of ROM using different bijections. If we denote by η the amount of measurements for which the implementation is DPA-resistant, then we perform the encryption η times before changing the bijections used. This strategy looks like the so-called *leakage-resilient* countermeasure [Koc05, GSD+11] (the key is regularly updated to avoid passive and active attacks).

9.3.1.2 Surface-Security Trade-off

The second idea is to use RAMs, rather than ROMs. Then, the bijections should be refreshed regularly. For instance, the simultaneous read-write facility of the dual-port Block RAM (BRAM) in FPGA can be used. While a given set of data is read from one port of the BRAM (say Port A), the next data using new bijections are computed and written to other port (Port B). The read and write ports are switched every η encryptions. Thus, the data which was written earlier to port B, would now be read. The port A would now be used to write the next data values. A control bit is used to toggle the operation of switching the ports.

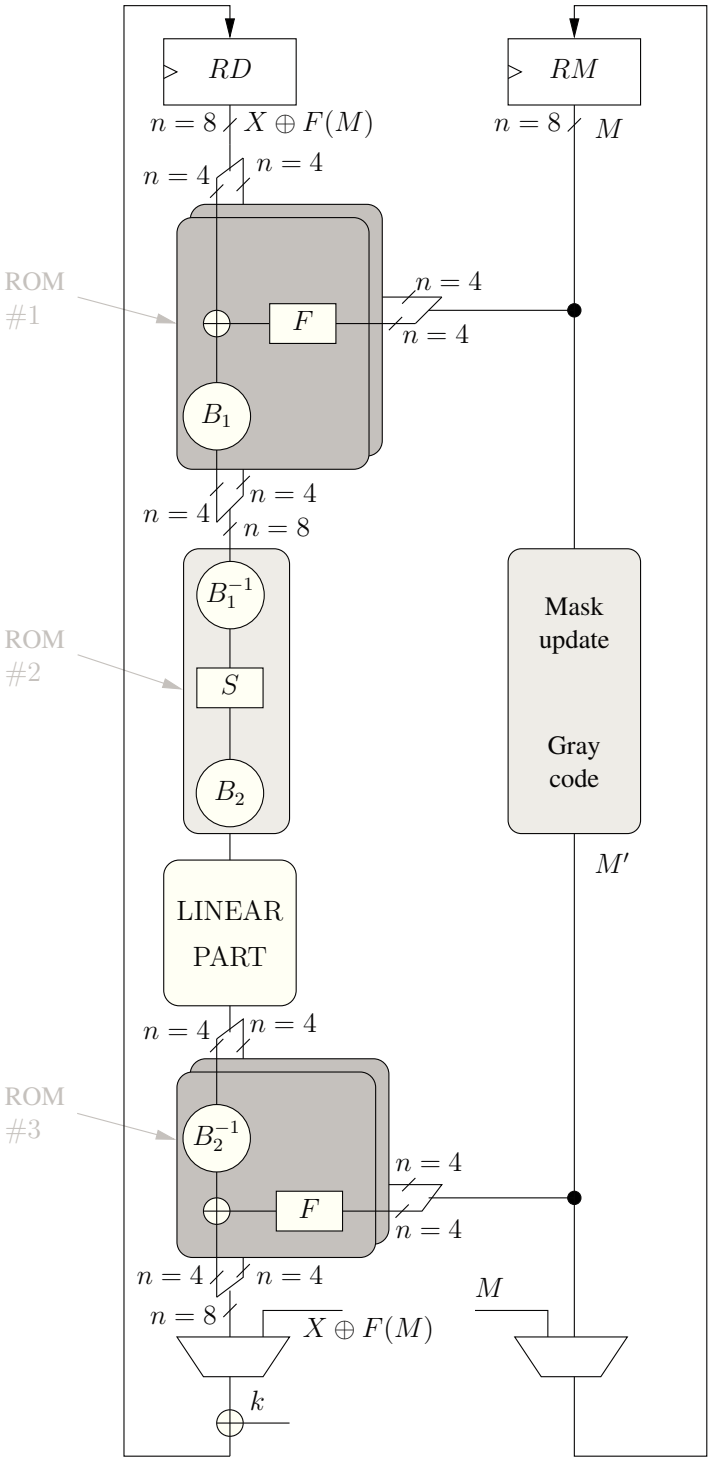


Figure 9.2: The USM hardware implementation.

Eventually the area consumption of this solution would be roughly twice the regular one (for memory blocks), for the same performances, but the implementation security is enhanced.

9.3.1.3 Partial Reconfiguration

Another way to update the RAMs is to partially re-configure the FPGA. The idea is to modify a portion of the bitstream, *i.e.* the one containing the RAM using different bijections. This way, the countermeasure size and performances would be almost unchanged, except for the reconfiguration times, during which the computations must be paused.

9.3.2 Complexity and Throughput Results

The proposed implementation has been tested in a StratixII FPGA of the SASEBO-B evaluation board. It has been compared with an unprotected AES and with a straightforward first-order masked AES (see Fig. 8.1 of Chapter 8). The table 9.2 summarizes the memory complexity for each implementation and the estimated frequency.

Table 9.2: Implementation results for reference and protected AES

	Unprotected	**First-Order masked AES**	**Our Proposal**	**Difference vs Unprot.**
Number of ALUTs	783	1287	951	+21%
Number of M4K ROM Blocs	20	32	80	× 4
Frequency (MHz)	133	90	103	-29%

These results show that the proposed method on hardware implementations has little impact on complexity and speed compared with the reference implementation. As we can see, the overheads in terms of logical cells, ROM blocks and clock frequency are all within reasonable ranges, even for real-life applications where several IPs are included in the same FPGA.

9.3.3 Attack Experiments

The security evaluation of this countermeasure was made in a real-life context. First, the power consumption is measured by acquiring the magnetic field radiated by the FPGA's decoupling capacitors. This non-intrusive methodology yields high quality measurements. Second, we applied several side channel distinguishers (*e.g.* first- and second-order CPA, MIA) to the leakage measurements in order to check them. For each scenario, we acquired a set of 100K power consumption traces using random masks and plaintexts. Finally, we performed the first-order success rate as in [SMY09].

For our proposed CM, the attacks performed worse. The success rates stay under 10% even when using all the 100K measurements. We conclude that the experiments on a real circuit shows the evident benefit of our CM against univariate side channel attacks.

9.4 Conclusions

In this chapter, we presented a new masking scheme for hardware S-box implementations which aims at blanking the leakage under the Hamming distance assumption. This method consists in using the Gray counter for the mask update and modifying the mask value by applying some specific functions F. When the leakage function deviates slightly from the model assumption, our solution still achieves excellent results. Practical implementations showed that the performances decrease in terms of complexity and speed are very limited, which is particularly true for the USM implementation which does not require large memories.

Chapter 10

First-Order Leakage Squeezing Countermeasure

In this chapter, we propose a new masking countermeasure called *leakage squeezing*. It consists in manipulating the mask through a bijection F, aimed at reducing the dependency between the shares' leakage. In particular, we explore the functions F that thwart HO-CPA of maximal order d. We mathematically demonstrate that optimal choices for F relate to optimal binary codes (in the sense of communication theory). First, we exhibit optimal linear F functions. Second, we note that for bitwidth values of n for which non-linear codes exist with better parameters than linear ones. These results are exemplified in the case $n = 8$, where the optimal F can be identified: it is derived from the optimal rate $1/2$ binary code of size $2n$, namely the Nordstrom-Robinson $(16, 256, 6)$ code. This example provides explicitly with the optimal protection that limits to one mask of byte-oriented algorithms such as the AES block cipher. It protects against all HO-CPA attacks of order $d \leq 5$. Eventually, the countermeasure is shown to be resilient to imperfect leakage models.

The results presented in this chapter have been published in collaboration with Sylvain Guilley and Jean-Luc Danger in the international *Workshop in Information Security Theory and Practice (WISTP 2011)* [MGD11b] and in collaboration with Claude Carlet, Sylvain Guilley and Jean-Luc Danger in the international conference on Cryptology *AfricaCrypt 2012* [MCGD12].

Contents

10.1 Studied Implementation and its Leakage

The two shares manipulated in a Boolean first-order countermeasure are $(X \oplus M, M)$. In the "leakage squeezing" CM we propose in this chapter, a bijection F is applied on the mask share in order to break the too strong link between the shares. Thus, the new shares are now $(X \oplus M, F(M))$. The schematic of this scheme is illustrated in Fig. 10.1. This figure highlights two registers, able to hold each one n-bit word. The left register hosts the masked data, $X \oplus M$, whereas the register on the right holds $F(M)$, *i.e.* the mask M passed through the bijection F. In this chapter, we are also concerned with the leakage from those two registers only. Indeed, they are undoubtedly the resource that leaks the most. Also, the rest of the logic can be advantageously hidden in tables, thereby limiting their side channel leakage [SVKH10]. It is referred to as "tabulated round logic" in Fig. 10.1.

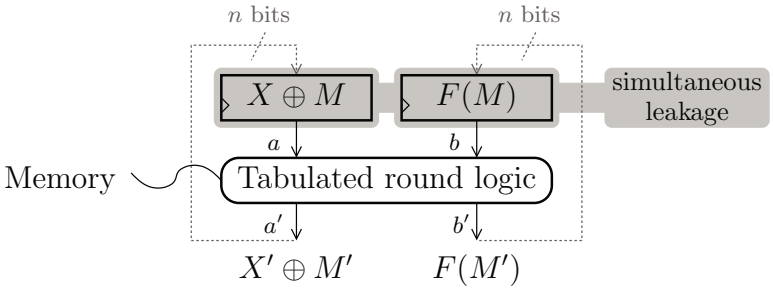


Figure 10.1: Setup of the first-order masking countermeasure with bijection F.

The computation of the bijection F shall not leak. Actually, F can be merged into memories, hence being totally dissolved. Therefore, the two shares $(X \oplus M, F(M))$ remain manipulated concomitantly only once, namely at the clock rising edge. For the sake of illustration, we provide with a typical functionality of this combinational logic hidden in memory. If we denote by C the round function and by R the mask refresh function, then the table implements:

- $a' = C(a \oplus F^{-1}(b)) \oplus R(F^{-1}(b))$ and

- $b' = F(R(F^{-1}(b)))$.

The detail of the tabulated round logic is represented in Fig. 10.2.

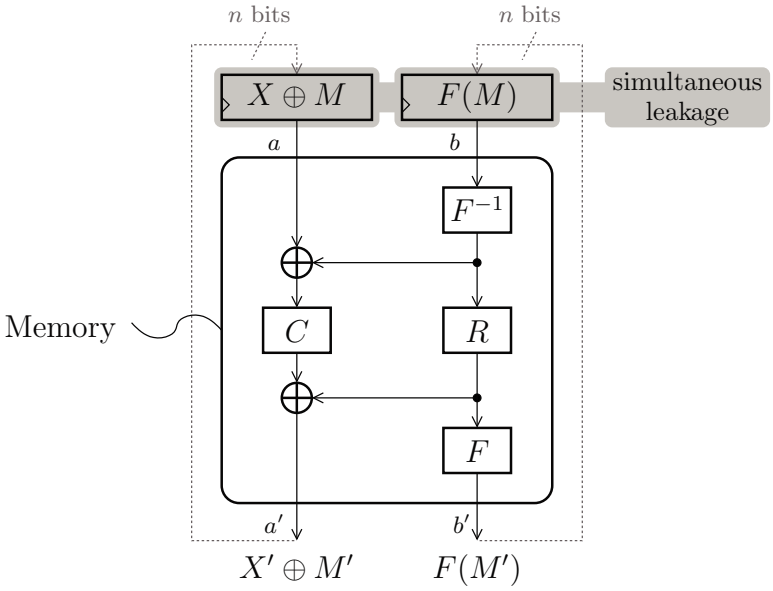


Figure 10.2: Detail of the function implemented in the tabulated round logic shown in Fig. 10.1.

In the context of a side channel attack against a block cipher, either the first round or the last round is targeted. Thus either the input X (plaintext) or the output X' (ciphertext) is known by the attacker. We assume that the device leaks in the Hamming distance model. Therefore, the sensitive variable to protect is $Z = X \oplus X'$. The leakage of the studied hardware (Fig. 10.1) is thus:

$$\begin{aligned} L &= \mathsf{HD}(X \oplus M, X' \oplus M') + \mathsf{HD}(F(M), F(M')) \\ &= \mathsf{HW}(Z \oplus M \oplus M') + \mathsf{HW}(F(M) \oplus F(M')) \ . \end{aligned} \tag{10.1}$$

10.2 Optimal Function in d^{th}-Order CPA

10.2.1 Optimal Function $f_{\mathbf{opt}}$

We recall the optimal function defined in [PRB09] that optimize the CPA attack $f_{\mathrm{opt}}(z) = \mathbb{E}[L - \mathbb{E}[L] \mid Z = z]$. If $z \mapsto f_{\mathrm{opt}}(z)$ is constant (*i.e.* $f_{\mathrm{opt}}(Z)$ is deterministic), then the authors in [PRB09] show that the correlation coefficient of the attack is null, which means that the attack fails.

This result can be applied on the studied leakage function of Eqn. (10.1), without F (*i.e.* with F equal to the identity function Id). The leakage function therefore simplifies in $\mathsf{HW}(Z \oplus M'') + \mathsf{HW}(M'')$, where $M'' = M \oplus M'$ is a uniformly distributed random variable in $\mathbb{F}_2^n$.

- In a first-order attack, the attacker uses $f_{\text{opt}}(Z) = \mathbb{E}[\mathsf{HW}(Z \oplus M'') + \mathsf{HW}(M'') - n \mid Z] = 0$, which is deterministic,
- whereas in a second-order attack, the attacker uses $f_{\text{opt}}(Z) = \mathbb{E}[\big(\mathsf{HW}(Z \oplus M'') + \mathsf{HW}(M'') - n\big)^2 \mid Z] = n - \mathsf{HW}(Z)$, which depends on Z. This result is easily obtained by developing the square. The only non-trivial term in this computation is $\mathbb{E}[\mathsf{HW}(z \oplus M'') \times \mathsf{HW}(M'')]$, which is proved to be equal to $\frac{n^2+n}{4} - \frac{1}{2}\mathsf{HW}(z)$ in [PRB09, Eqn. (19)].

In summary, without F, a first-order attack is thwarted, but a second-order attack will succeed. In the sequel, when mentioning HO-CPA attack, we implicitly mean the univariate correlation attack that uses a higher-order moment of the traces instead of the raw traces. Nonetheless, as explained in [WW04], this second-order attack requires more traces than a first-order attack on an unprotected version that do not use any mask. Indeed, the noise is squared and thus its effect is exacerbated. More generally, the higher the order d of a HO-CPA attack, the greater the impact of the noise. Thus, attacks are still possible for small d, but get more and more difficult when d increases. Therefore, our objective is to improve the masking CM so that the HO-CPA fails for orders $[\![1, d]\!]$, with d being as high as possible. This translates in terms of $f_{\text{opt}}(Z)$ by having $\mathbb{E}[\big(\mathsf{HW}(Z \oplus M \oplus M') + \mathsf{HW}(F(M) \oplus F(M')) - n\big)^d \mid Z]$ deterministic (*i.e.* independent of the sensitive variable Z) for the highest possible values of the integer d. Thus, when developing the sum raised at the power d, we are led to study terms of this form:

$$\begin{aligned} \mathsf{Term}[p,q](f_{\text{opt}})(z) &\doteq \mathbb{E}[\mathsf{HW}^p(z \oplus M \oplus M') \times \mathsf{HW}^q(F(M) \oplus F(M'))] \\ &= \mathbb{E}[\mathsf{HW}^p(z \oplus M'') \times \mathsf{HW}^q(F(M) \oplus F(M \oplus M''))] \quad (10.2) \end{aligned}$$

where p and q are two positive integers. If either p or q is null, then trivially, $\mathsf{Term}[p,q](f_{\text{opt}})$ is constant. We are thus interested more specifically in p and q values that are strictly positive. We note that in order to resist d^{th}-order HO-CPA, $\mathsf{Term}[p,q](f_{\text{opt}})(z)$ must not depend on z for all p and q that satisfy $p + q \leq d$.

Incidentally, the same condition would hold if the two shares were:

- $F_0(X \oplus M)$ and $F_1(M)$, where F_0 and F_1 are two bijections, with F_0 linear, instead of merely
- $X \oplus M$ and $F(M)$, as in Fig. 10.2.

This new setting is more general, since by choosing $F_0 = \mathsf{Id}$ (linear bijection) and $F_1 = F$ (arbitrary bijection), it comes down to that of Fig. 10.2. The generalization of Eqn. (10.2) is:

$$\begin{aligned} & \mathbb{E}[\mathsf{HW}^p(F_0(z \oplus M \oplus M')) \times \mathsf{HW}^q(F_1(M) \oplus F_1(M'))] \quad \text{(because } F_0 \text{ is linear)} \\ = \; & \mathbb{E}[\mathsf{HW}^p(F_0(z) \oplus F_0(M) \oplus F_0(M')) \times \mathsf{HW}^q(F_1(M) \oplus F_1(M'))] \quad (\textit{idem}) \\ = \; & \mathbb{E}[\mathsf{HW}^p(\tilde{z} \oplus \tilde{M} \oplus \tilde{M'}) \times \mathsf{HW}^q(F_1(F_0^{-1}(\tilde{M})) \oplus F_1(F_0^{-1}(\tilde{M'})))] \;, \quad (10.3) \end{aligned}$$

where $\tilde{z} \doteq F_0(z)$, $\tilde{M} \doteq F_0(M)$ and $\tilde{M''} \doteq F_0(M'')$. The random variable $\tilde{M}$ (resp. $\tilde{M''}$) is uniformly distributed because M (resp. M'') is also uniformly distributed and F_0 (resp. F_1) is a bijection. Thus, the more general setting is secure if Eqn. (10.3) does not depend on $\tilde{z}$ for all $p+q \leq d$, which is equivalent to having the setting of Fig. 10.2 secure with $F = F_1 \circ F_0^{-1}$. For this reason, we simply reason in the sequel with only one bijection applied to the mask, *i.e.* the masked sensitive data being manipulated plain.

10.2.2 Sequential Leakage

If the shares $X \oplus M$ and $F(M)$ are manipulated at different dates (*i.e.* not simultaneously as in Fig. 10.2), then the attacker could typically attempt to combine their leakage. The paper [PRB09] precisely covers this topic: it proves that the best combination tool is the centered product. Thus, the attacker's strategy remains in line with that discussed on parallel leakage: terms such as $\mathsf{Term}[p,q](f_{\text{opt}})(z)$ are checked for dependence in z. So, the results discussed here also apply to software implementations that handle the shares sequentially.

10.2.3 Condition on F for the Resistance Against Second-Order CPA

To resist second-order CPA, the term in Eqn. (10.2) must be constant for $p+q \leq 2$. As just mentioned, the cases $(p,q) = (2,0)$ and $(0,2)$ are trivial. This subsection thus focuses on the case where $p = q = 1$.

The term $F(M) \oplus F(M \oplus M'')$ is the value at M of the derivative of F in the direction M'', noted $D_{M''}F(M)$. It can be observed that Eqn. (10.2) also writes as a convolution product: $\mathsf{Term}[p,q](f_{\text{opt}})(z) = \frac{1}{2^n}\left(\mathsf{HW} \otimes \mathbb{E}[\mathsf{HW}(D_{(.)}F(M))]\right)(z)$, where $\otimes$ denotes the convolution operation. In this expression, $\mathbb{E}[\mathsf{HW}(D_{(.)}F(M))]$ designates the function:

$$\begin{array}{rccl} \mathbb{E}[\mathsf{HW}(D_{(.)}F(M))]: & \mathbb{F}_2^n & \to & \mathbb{Z} \\ & m'' & \mapsto & \mathbb{E}[\mathsf{HW}(D_{m''}F(M))] = \frac{1}{2^n}\sum_m \mathsf{HW}(D_{m''}F(m)) \ . \end{array}$$

The Fourier transform of a function $f : \mathbb{F}_2^n \to \mathbb{Z}$ is defined as $\widehat{f} : \mathbb{F}_2^n \to \mathbb{Z}, z \mapsto \sum_{y \in \mathbb{F}_2^n} f(y)(-1)^{y \cdot z}$. An appealing property of this Fourier transform is that it turns a convolution into a product. So, we have:

$$\begin{array}{lll} f_{\text{opt}}(z) = \text{cst} & \iff & \widehat{f_{\text{opt}}}(a) \propto \delta(a) \quad \begin{array}{l} // \text{ where } \propto \text{ means "is proportional to",} \\ // \ a \in \mathbb{F}_2^n \text{ and } \delta(\,\cdot\,) \text{ is the Kronecker symbol.} \end{array} \\ & \iff & \widehat{\mathsf{HW}}(a) \times \mathbb{E}[\widehat{\mathsf{HW} \circ D_{(.)}F(M)}](a) = \left(n \times 2^{n-1}\right)^2 \times \delta(a) \\ & \iff & \forall a \neq 0, \widehat{\mathsf{HW}}(a) = 0 \ \text{ or } \ \mathbb{E}[\widehat{\mathsf{HW} \circ D_{(.)}F(M)}](a) = 0\,. \end{array} \tag{10.4}$$

To prove the second line, we note that on the one hand:

$$\widehat{\mathsf{HW}}(0) = \sum_z \mathsf{HW}(z) \cdot (-1)^{0 \cdot z} = \frac{n}{2} 2^n \ ,$$

and on the other hand:

$$\begin{aligned}
&\mathbb{E}[\widehat{\mathsf{HW} \circ D_{(\cdot)}F}(M)](0)\\
=\ &\textstyle\sum_z \mathbb{E}[\mathsf{HW}(D_zF(M))(-1)^{0\cdot z}]\\
=\ &\mathbb{E}[\textstyle\sum_z \mathsf{HW}(F(M)\oplus F(M\oplus z))]\\
=\ &\mathbb{E}[\textstyle\sum_{z'} \mathsf{HW}(z')] \quad // \text{ Because } \forall m, z \mapsto F(m)\oplus F(m\oplus z) \text{ is bijective}\\
=\ &\mathbb{E}[\tfrac{n}{2}2^n] = \tfrac{n}{2}2^n \ .
\end{aligned}$$

Now, if we denote by e_i the lines of the identity matrix I_n of size $n \times n$,

$$\begin{aligned}
\widehat{\mathsf{HW}}(a) &= \sum_z \frac{1}{2}\sum_{i=1}^{n}\left(1-(-1)^{z_i}\right)(-1)^{a\cdot z}\\
&= n\cdot 2^{n-1}\delta(a) - \tfrac{1}{2}\textstyle\sum_z\sum_{i=1}^{n}(-1)^{(a\oplus e_i)\cdot z}\\
&= \begin{cases} n\cdot 2^{n-1} & \text{if } a=0,\\ -2^{n-1} & \text{if } \exists i\in[\![1,n]\!], \text{ such that } a=e_i,\\ 0 & \text{otherwise.}\end{cases}
\end{aligned} \tag{10.5}$$

Thus, the problem comes down to finding a function F such that: $\mathbb{E}[\widehat{\mathsf{HW} \circ D_{(\cdot)}F}(M)](a) = 0$ for all $a = e_i$. This condition rewrites:

$$\forall a = e_i, \quad \sum_{z,m} \mathsf{HW}(F(m)\oplus F(m\oplus z))(-1)^{a\cdot z} = 0 \ . \tag{10.6}$$

Let $a \neq 0$. Then:

$$\begin{aligned}
&\textstyle\sum_{z,m} \mathsf{HW}(F(m)\oplus F(m\oplus z))(-1)^{a\cdot z}\\
=\ &\textstyle\sum_{z,m}\frac{1}{2}\sum_{i=1}^{n}\left(1-(-1)^{F_i(m)\oplus F_i(m\oplus z)}\right)(-1)^{a\cdot z}\\
=\ &\cancel{n2^{2n-1}\delta(a)} - \tfrac{1}{2}\textstyle\sum_{i=1}^{n}\sum_{z,m}(-1)^{F_i(m)\oplus F_i(m\oplus z)\oplus a\cdot z}\\
=\ &-\tfrac{1}{2}\textstyle\sum_{i=1}^{n}\sum_m(-1)^{F_i(m)}\sum_z(-1)^{a\cdot z\oplus F_i(m\oplus z)}\\
=\ &-\tfrac{1}{2}\textstyle\sum_{i=1}^{n}\sum_m(-1)^{F_i(m)}\sum_z(-1)^{a\cdot(z\oplus m)\oplus F_i(z)} \quad // \ z \leftarrow z\oplus m\\
=\ &-\tfrac{1}{2}\textstyle\sum_{i=1}^{n}\sum_m(-1)^{a\cdot m\oplus F_i(m)}\sum_z(-1)^{a\cdot z\oplus F_i(z)}\\
=\ &-\tfrac{1}{2}\textstyle\sum_{i=1}^{n}\left(\sum_m(-1)^{a\cdot m\oplus F_i(m)}\right)^2\\
=\ &-\tfrac{1}{2}\textstyle\sum_{i=1}^{n}\left(\widehat{(-1)^{F_i}}(a)\right)^2 \ .
\end{aligned}$$

Thus, this quantity is null if and only if $\forall i \in [\![1,n]\!]$, $\widehat{(-1)^{F_i}}(a) = 0$. Generalizing the Fourier transform on vectorial Boolean functions (by applying the transformation component-wise), and using the notation f_χ for the sign function of f (also component-wise), then Eqn. (10.6) is equivalent to: $\forall a = e_i, \widehat{F_\chi}(a) = 0$. The Fourier transform of a sign function is also known as the Walsh-Hadamard transform[1]. Now, as F is balanced (since bijective), this equality also holds for $a = 0$. By definition,

[1]Both notions are linked through the relationship $\forall a, \widehat{F_\chi}(a) = 2^n\delta(a) - 2\widehat{F}(a)$.

a Boolean function g is d-resilient if its Walsh-Hadamard transform $\widehat{g_\chi}(a)$ is null for all a such as $\mathsf{HW}(a) \leq d$. Thus every coordinate of F is 1-resilient. Constructions for such functions exist, as explained in [Car10a, Sec. 8.7].

In the next subsection, we use P-resilient functions F: according to the definition, they are balanced when up to P input bits are fixed.

10.2.4 Condition on F for the Resistance Against d^{th}-Order CPA

A generalization of the previous result for arbitrary $p, q \in \mathbb{N}^\star \doteq \mathbb{N}\backslash\{0\}$ is presented in this section. We have the following theorem:

Theorem 5. *Let P and Q be two positive integers, and F a bijection of $\mathbb{F}_2^n$. Eqn. (10.2) is constant for all $p \in [\![0, P]\!]$ and $q \in [\![0, Q]\!]$ if and only if:*

$$\forall a, b \in \mathbb{F}_2^n, 0 < \mathsf{HW}(a) \leq P, 0 \leq \mathsf{HW}(b) \leq Q, \quad \widehat{(b \cdot F)_\chi}(a) = 0 \ . \tag{10.7}$$

Before proving Theorem 5, let us introduce the following intermediate result.

10.2.4.1 First Intermediate Result for the Proof of Theorem 5

Theorem 6. $\forall a \in \mathbb{F}_2^n, \forall p \in \mathbb{N}, \quad \widehat{\mathsf{HW}^p}(a) = 0 \iff \mathsf{HW}(a) > p.$

Let us define the function $H(n, p, h) \doteq \sum_{z \in \mathbb{F}_2^n} \mathsf{HW}^p(z)(-1)^{z \cdot \oplus_{i=1}^h e_i}$, for $n \in \mathbb{N}^\star$, $p \in \mathbb{N}$ and $h \in [\![0, n]\!]$. It is tabulated for $n = 4$ in Tab. 10.1. The value $H(n, n, n)$, indicated by dagger sign (*i.e.* "†") in the table, is equal to $(-1)^n n!$.

Table 10.1: Some values of $H(n = 4, p, h)$.

		h				
		0	1	2	3	4
p	0	16	0	0	0	0
	1	32	−8	0	0	0
	2	80	−32	8	0	0
	3	224	−116	48	−12	0
	4	680	−416	224	−96	$24^\dagger$
	⋮	> 0	< 0	> 0	< 0	> 0

As the order of the bits of the dummy variable z is indifferent in the term $\sum_z \mathsf{HW}^p(z)(-1)^{a \cdot z}$, we have $\widehat{\mathsf{HW}^p}(a) = H(n, p, \mathsf{HW}(a))$.

Lemma 3.

$$H(n, p, n) \quad \begin{cases} = 0 & \text{if} \quad p < n, \\ > 0 & \text{if} \quad p \geq n \text{ and } n \text{ is even}, \\ < 0 & \text{if} \quad p \geq n \text{ and } n \text{ is odd}. \end{cases}$$

Proof.

$$\begin{aligned} H(n,p,n) &= \sum_z \mathsf{HW}^p(z)(-1)^{z\cdot\oplus_{i=1}^n e_i} = \sum_z \mathsf{HW}^p(z)(-1)^{\mathsf{HW}(z)} \\ &= \sum_{j=0}^{n} \binom{n}{j} j^p (-1)^j = (-1)^n \sum_{j=0}^{n} \binom{n}{j} j^p (-1)^{n-j} = (-1)^n n! \left\{ {p \atop n} \right\} , \end{aligned}$$

where $\left\{ {p \atop n} \right\}$ is a Stirling number of the second kind [Ed.09]. More precisely, it is the number of ways of partitioning a set of p elements into n nonempty sets. Consequently, $\left\{ {p \atop n} \right\} = 0$ if $n > p$, because otherwise at least one set would be empty. Also, $\left\{ {p \atop n} \right\} > 0$ if $n \leq p$. Now, the sign of $H(n,p,n)$ depends on the parity of n if $n \leq p$. It is positive (resp. negative) if n is even (resp. odd). □

Lemma 4.

$$H(n,p,h) \quad \begin{cases} = 0 & \text{if} \quad p < h, \\ > 0 & \text{if} \quad p \geq h \text{ and } h \text{ is even}, \\ < 0 & \text{if} \quad p \geq h \text{ and } h \text{ is odd}. \end{cases}$$

Proof. This lemma has already been proved in Lemma 3 if $h = n$. Thus, we assume in the remainder of this proof that $h < n$. For $z \in \mathbb{F}_2^n$, we note $z = (z_L, z_H)$, where $z_L \in \mathbb{F}_2^h$ and $z_H \in \mathbb{F}_2^{n-h}$.

$$\begin{aligned} H(n,p,h) &= \sum_{(z_L,z_H)} \mathsf{HW}^p((z_L,0) \oplus (0,z_H))(-1)^{\left(z_L\cdot\oplus_{i=1}^h e_i\right)\oplus(z_H\cdot 0)} \\ &= \sum_{(z_L,z_H)} (\mathsf{HW}(z_L) + \mathsf{HW}(z_H))^p (-1)^{z_L\cdot\oplus_{i=1}^h e_i} \\ &= \sum_{(z_L,z_H)} \sum_{j=0}^{p} \binom{p}{j} \times \mathsf{HW}^j(z_L) \times \mathsf{HW}^{p-j}(z_H)(-1)^{z_L\cdot\oplus_{i=1}^h e_i} \\ &= \sum_{j=0}^{p} \binom{p}{j} \sum_{z_L} \mathsf{HW}^j(z_L)(-1)^{z_L\cdot\oplus_{i=1}^h e_i} \times \sum_{z_H} \mathsf{HW}^{p-j}(z_H) \\ &= \sum_{j=0}^{p} \binom{p}{j} \times H(h,j,h) \times H(n-h,p-j,0) \ . \end{aligned} \tag{10.8}$$

Now, given Lemma 3, $\forall j < h$, $H(h,j,h) = 0$. Thus, if $p < h$, then all the terms $H(h,j,h)$ involved in Eqn. (10.8) are null, since $j \in [\![0,p]\!]$ is strictly inferior to h. Besides, for all $j \in [\![0,p]\!]$, $\binom{p}{j}$ and $H(n-h,p-j,0)$ are strictly positive. If $p \geq h$, the terms $H(h,j,h)$ for $j \leq p$ are:

- either all strictly positive if h is even, or
- all strictly negative if h is odd.

Hence, so is the sum in Eqn. (10.8) □

We give hereafter the proof of Theorem 6.

Proof. As already noticed, $\widehat{\mathsf{HW}^p}(a) = H(n, p, \mathsf{HW}(a))$. According to Lemma 4, this quantity is null if and only if $p < \mathsf{HW}(a)$. □

10.2.4.2 Second Intermediate Result for the Proof of Theorem 5

For every $X \in \mathbb{F}_2^n$, we have:

$$\begin{aligned}
\left(\sum_{i=1}^{n}(-1)^{X\cdot e_i}\right)^j &= \sum_{i_1,\cdots,i_j\in[\![1,n]\!]^j}\prod_{l=1}^{j}(-1)^{X\cdot e_{i_l}} \\
&= \sum_{i_1,\cdots,i_j\in[\![1,n]\!]^j}(-1)^{X\cdot\oplus_{l=1}^{j}e_{i_l}} \quad \left\{\begin{array}{l}\text{Under this form,}\\ \text{some terms appear}\\ \text{multiple times.}\end{array}\right. \\
&= \sum_{\substack{(k_1,\cdots,k_n)\in\mathbb{N}^n,\\ k_1+\cdots+k_n=j}}\binom{j}{k_1,\cdots,k_n}(-1)^{X\cdot\left(\oplus_{i=1}^{n}k_ie_i\right)}\ , \qquad (10.9)
\end{aligned}$$

where each vector k_ie_i in $\bigoplus_{i=1}^{n}k_ie_i$ is either e_i if k_i is odd or 0 otherwise. In the Eqn. (10.9), the term $\binom{j}{k_1,\cdots,k_n}$ is a multinomial coefficient. Then:

$$\begin{aligned}
&\sum_{z,m}\mathsf{HW}^q(F(m)\oplus F(m\oplus z))(-1)^{a\cdot z} \\
=\ &\frac{1}{2^q}\sum_{z,m}\left(n-\sum_{i=1}^{n}(-1)^{F_i(m)\oplus F_i(m\oplus z)}\right)^q(-1)^{a\cdot z} \\
=\ &\frac{1}{2^q}\sum_{z,m}\sum_{j=0}^{q}\binom{q}{j}n^{q-j}(-1)^j\left(\sum_{i=1}^{n}(-1)^{F_i(m)\oplus F_i(m\oplus z)}\right)^j(-1)^{a\cdot z} \quad \text{// See Eqn. (10.9)} \\
=\ &\frac{1}{2^q}\sum_{j=0}^{q}\binom{q}{j}n^{q-j}(-1)^j\sum_{k_1+\cdots+k_n=j}\binom{j}{k_1,\cdots,k_n}\sum_{z,m}(-1)^{(F(m)\oplus F(m\oplus z))\cdot\left(\oplus_{i=1}^{n}k_ie_i\right)}(-1)^{a\cdot z} \\
=\ &\frac{1}{2^q}\sum_{j=0}^{q}\binom{q}{j}n^{q-j}(-1)^j\sum_{k_1+\cdots+k_n=j}\binom{j}{k_1,\cdots,k_n}\left(\widehat{(\oplus_{i=1}^{n}k_ie_i)\cdot F}\right)_\chi(a)\Big)^2\ . \qquad (10.10)
\end{aligned}$$

10.2.4.3 The Complete Proof of Theorem 5

Based on the previous intermediate results, we give hereafter the complete proof of Theorem 5.

Proof. As requested by Theorem 5, we introduce P and Q, two positive integers, and F, a bijection of $\mathbb{F}_2^n$. With a reasoning close to that of Eqn. (10.4) for the case

$p = q = 1$, we get:

$$
\begin{aligned}
& \forall p \in [\![0, P]\!], \forall q \in [\![0, Q]\!], \text{the function } f_{\text{opt}}, \text{ defined in Eqn. (10.2), is constant} \\
\iff\ & \forall p \in [\![0, P]\!], \forall q \in [\![0, Q]\!], \forall a \in \mathbb{F}_2^{n\star}, \widehat{\mathsf{HW}^p}(a) = 0 \text{ or } \mathbb{E}[\widehat{\mathsf{HW}^q \circ D_{(.)}F(M)}](a) = 0 \\
\iff\ & \forall p \in [\![0, P]\!], \forall q \in [\![0, Q]\!], \forall a \in \mathbb{F}_2^{n\star}, \begin{cases} \text{either } \mathsf{HW}(a) > p \text{ (See Theorem 6)} \\ \text{or Eqn. (10.10) is zero} \end{cases} \\
\iff\ & \forall p \in [\![0, P]\!], \forall q \in [\![0, Q]\!], \forall a \in \mathbb{F}_2^{n\star}, \mathsf{HW}(a) \leq p \implies \text{Eqn. (10.10) is zero} \\
\iff\ & \begin{cases} \forall p \in [\![0, P]\!], \\ \forall q \in [\![0, Q]\!], \\ \forall a \in \mathbb{F}_2^{n\star}, \\ \quad \mathsf{HW}(a) \leq p \end{cases} \implies \begin{cases} \underline{q = 1}: & \forall b, \mathsf{HW}(b) \leq 1 \implies \widehat{(b \cdot F)_\chi}(a) = 0, \\ \underline{q = 2}: & \forall b, \mathsf{HW}(b) \leq 2 \implies \widehat{(b \cdot F)_\chi}(a) = 0, \\ \quad \vdots & \\ \underline{q = Q}: & \forall b, \mathsf{HW}(b) \leq Q \implies \widehat{(b \cdot F)_\chi}(a) = 0. \end{cases}
\end{aligned}
\tag{10.11}
$$

We provide with an explanation for the last part of Eqn. (10.11). The terms of Eqn. (10.10) corresponding to a given j are a sum of squares (weighted by quantities of the same sign). Thus, if those terms for $j < q$ are null, then the ones for $j = q$ must also be null, because the complete sum (of squares) is null by hypothesis. □

Remark 3. *The condition expressed in Eqn. (10.7) of Theorem 5 can be reformulated as follows. Every restriction of the bijective (n, n)-function F to Q components is an (n, Q)-function that is P-resilient.*

10.3 Existence of Bijections Meeting Eqn. (10.7)

In this section, we find bijections that meet Eqn. (10.7).

10.3.1 Three Conditions on Optimal Bijections F

10.3.1.1 Condition in Terms of Walsh-Hadamard Transform

The condition expressed in Eqn. (10.7) for Theorem 5 rewrites:

$$
\begin{aligned}
& \forall b \in \mathbb{F}_2^{n\star} \doteq \mathbb{F}_2^n \backslash \{0\} \text{ and } \forall a \in \mathbb{F}_2^n\,, \\
& \text{if } \mathsf{HW}(a) \leq d - \mathsf{HW}(b) \text{ then } \widehat{(b \cdot F)_\chi}(a) = 0\,.
\end{aligned}
\tag{10.12}
$$

10.3.1.2 Condition in Terms of Correlation-Immunity

Given any (n, n)-function F, let $C \doteq \{(x, F(x)); x \in \mathbb{F}_2^n\}$ be the graph of F. The indicator 1_C of C is the Boolean function:

$$
1_C : \xi \in \mathbb{F}_2^{2n} \mapsto \begin{cases} 1 & \text{if } \xi \in C\,, \\ 0 & \text{otherwise}\,. \end{cases}
$$

The condition on F given in Eqn. (10.12) is satisfied if and only if the indicator of the graph C of F is d^{th}-order correlation-immune (see definition in [CCCS91]);

this result comes from the characterization of correlation-immune functions by their Fourier transform available in [XM88].

By the definition of correlation-immune functions, we also have this characterization on F. For all subset I of $\{1, \cdots, 2n\}$ of cardinality $|I|$ at most d, and for all $a \in \mathbb{F}_2^I$, there are $\frac{|C|}{2^{|I|}}$ codewords of C whose coordinates of indices $i \in I$ coincide with those of a. This means that, by fixing some coordinates of x and some coordinates of $F(x)$, it is impossible to bias $C = \{(x, F(x)); x \in \mathbb{F}_2^n\}$ if the number of fixed coordinates does not exceed d.

10.3.1.3 Condition in Terms of Code

Given any (n, n)-function F, the weight enumerator $W_C(X, Y)$ and distance enumerator $D_C(X, Y)$ of the code C are:

- $W_C(X, Y) \doteq \sum_{x \in \mathbb{F}_2^n} X^{2n - \mathsf{HW}(x, F(x))} Y^{\mathsf{HW}(x, F(x))}$ and
- $D_C(X, Y) \doteq \frac{1}{|C|} \sum_{x,y \in \mathbb{F}_2^n} X^{2n - \mathsf{HW}(x \oplus y, F(x) \oplus F(y))} Y^{\mathsf{HW}(x \oplus y, F(x) \oplus F(y))}$.

We have $W_C(X+Y, X-Y) = \sum_{a,b \in \mathbb{F}_2^n} \left(\sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F(x) + a \cdot x} \right) X^{2n - \mathsf{HW}(a,b)} Y^{\mathsf{HW}(a,b)}$ and $D_C(X+Y, X-Y) = \frac{1}{|C|} \sum_{a,b \in \mathbb{F}_2^n} \left(\sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F(x) \oplus a \cdot x} \right)^2 X^{2n - \mathsf{HW}(a,b)} Y^{\mathsf{HW}(a,b)}$. Hence $d + 1$ is exactly the minimum value of the nonzero exponents of Y with nonzero coefficients in $D_C(X+Y, X-Y)$, called the dual distance of C in the sense of Delsarte [Del73, MS77].

In summary, our goal can also be restated as follows: we seek to find a bijection F such as the code C equal to the graph of F has the largest possible dual distance.

10.3.2 Optimal Linear Bijections

The bijection F can be chosen linear. All linear (n, n)-functions write $F(x) = (x \cdot v_1, \cdots, x \cdot v_n)$, where v_i are elements of $\mathbb{F}_2^n$. F is bijective if and only if $(v_1, \cdots, v_n)$ is a basis of $\mathbb{F}_2^n$. We have:

$$\begin{aligned}
\widehat{(b \cdot F)_\chi}(a) = 0 \quad &\iff \quad \textstyle\sum_x (-1)^{b \cdot F(x) \oplus x \cdot a} = 0 \\
&\iff \quad \textstyle\sum_x (-1)^{\oplus_{i=1}^n b_i (x \cdot v_i) \oplus x \cdot a} = 0 \\
&\iff \quad \textstyle\sum_x (-1)^{x \cdot \oplus_{i=1}^n (b_i v_i) \oplus x \cdot a} = 0 \\
&\iff \quad \textstyle\sum_x (-1)^{x \cdot \left(\oplus_{i=1}^n (b_i v_i) \oplus a \right)} = 0 \\
&\iff \quad \textstyle\bigoplus_{i=1}^n b_i v_i \neq a \,.
\end{aligned}$$

As this is true for all a such that $\mathsf{HW}(a) \leq d - \mathsf{HW}(b)$, we have the necessary and sufficient condition:

$$\forall b \neq 0, \quad \mathsf{HW}\left(\textstyle\bigoplus_{i=1}^n b_i v_i\right) > d - \mathsf{HW}(b) \ . \tag{10.13}$$

We notice that the set of ordered pairs $C' \doteq \{(b, \bigoplus_{i=1}^n b_i v_i) \,; b \in \mathbb{F}_2^n\}$ forms a vector subspace of $\mathbb{F}_2^{2n}$. Therefore, it defines a $[2n, n, \delta]$ binary linear code, where δ is

Table 10.2: Minimal distance of some binary optimal linear rate 1/2 codes.

S-boxes of algorithm [iso]	**DES, MISTY1, CAST-128, HIGHT**	**n/a**	**n/a**	**n/a**	**AES, Camellia, SEED**
Value of $2n$	8	10	12	14	16
Value of $\delta_{\max}(n)$	4	4	4	4	5

its minimum (direct) distance. Because of Eqn. (10.13), the necessary and sufficient condition becomes merely $\delta > d$.

The codes C and C' have rate 1/2; and F being bijective, each of these codes admits the two information sets $[\![1, n]\!]$ and $[\![n+1, 2n]\!]$. We recall that a set of indices I is called an information set of a code if every possible tuple occurs in exactly one codeword within the specified coordinates $x_i;\ i \in I$. More generally, a rate 1/2 code which admits two complementary information sets is called a *Complementary Information Set* code, or CIS code in short. These codes are studied in [CGKS12]. From any such linear CIS code, it is possible to deduce a linear bijection F. Indeed, by permuting the coordinates, these two information sets can be respectively available at coordinates of indices $[\![1, n]\!]$ and $[\![n+1, 2n]\!]$ in the codewords. The $[2n, n, \delta]$ binary linear code can thus be spawned by a generator matrix $(A \quad B)$, where A and B are two $n \times n$ invertible matrices. A left-hand side multiplication by the inverse of A turns the generic generator matrix into the systematic representation of the code, namely $(I_n \quad G)$, where $G \doteq A^{-1} \times B$. This corresponds to a code $\{(x, F(x)); x \in \mathbb{F}_2^n\}$ where F is bijective because G is invertible (in the general case of 1/2 rate codes, the systematic representation also writes as $(I_n \quad G)$, but G in not necessarily invertible). It also corresponds to a code $C' = \{(b, \bigoplus_{i=1}^{n} b_i v_i; b \in \mathbb{F}_2^n\}$ giving a bijection F.

Now, $[2n, n, \delta]$ binary linear codes have been well studied. From the condition $\delta > d$, we deduce that the best achievable d using a linear bijection F is $\delta_{\max}(n) - 1$. Consequently, $d \leq n$, and this bound is met if and only if C is *Maximum Distance Separable* (MDS), which is equivalent to saying that F is a multipermutation [Vau94]. However, binary MDS codes exist only if the code dimension is equal to 0, 1, the code length or the code length minus 1. Thus, binary MDS codes not exist if $n > 1$, hence the bound $d \leq n - 1$.

The greatest minimal distance $\delta_{\max}(n)$ of rate 1/2 binary linear codes is known (refer for instance to [GO04]); corresponding codes are called "optimal". For some practical values of n, they are recalled in Tab. 10.2.

In particular, this result proves that with linear F, it is possible to protect:

- DES against all HO-CPA of order $d \leq 3$, and
- AES against all HO-CPA of order $d \leq 4$.

Indeed, there exist optimal linear codes that also enjoy the CIS property; as mentioned in [CGKS12], this notion is not trivial, since for instance there exists a

$[34, 17, 8]$ code which is not CIS. Now, for $n = 4$, the matrix $G = \overline{I_4}$ is invertible since $\overline{I_4}^{-1} = \overline{I_4}$.

The optimal linear function in the case $n = 8$ is generated by the non-identity half of the systematic matrix of $[16, 8, 5]$ code. This matrix is[2]:

$$\begin{pmatrix} 1&1&1&0&1&0&1&1 & 1&0&0&0&0&0&0&0 \\ 0&1&1&1&0&1&0&1 & 1&1&0&0&0&0&0&0 \\ 0&0&1&1&1&0&1&0 & 1&1&1&0&0&0&0&0 \\ 0&0&0&1&1&1&0&1 & 0&1&1&1&0&0&0&0 \\ 0&0&0&0&1&1&1&0 & 1&0&1&1&1&0&0&0 \\ 0&0&0&0&0&1&1&1 & 0&1&0&1&1&1&0&0 \\ 0&0&0&0&0&0&1&1 & 1&0&1&0&1&1&1&0 \\ 0&0&0&0&0&0&0&1 & 1&1&0&1&0&1&1&1 \end{pmatrix} \begin{matrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \end{matrix} . \tag{10.14}$$

It is already in row echelon form. Therefore, it can be turned into systematic form with a Gauss-Jordan elimination. It involves the following linear operations on the rows:

$$\begin{pmatrix} 1&1&1&0&1&0&1&1 & 1&0&0&0&0&0&0&0 \\ 0&1&1&1&0&1&0&1 & 1&1&0&0&0&0&0&0 \\ 0&0&1&1&1&0&1&0 & 1&1&1&0&0&0&0&0 \\ 0&0&0&1&1&1&0&1 & 0&1&1&1&0&0&0&0 \\ 0&0&0&0&1&1&1&0 & 1&0&1&1&1&0&0&0 \\ 0&0&0&0&0&1&1&1 & 0&1&0&1&1&1&0&0 \\ 0&0&0&0&0&0&1&1 & 1&0&1&0&1&1&1&0 \\ 0&0&0&0&0&0&0&1 & 1&1&0&1&0&1&1&1 \end{pmatrix} \begin{matrix} L'_1 \leftarrow L_1 \oplus L_2 \oplus L_4 \oplus L_7 \\ L'_2 \leftarrow L_2 \oplus L_3 \oplus L_5 \oplus L_8 \\ L'_3 \leftarrow L_3 \oplus L_4 \oplus L_6 \\ L'_4 \leftarrow L_4 \oplus L_5 \oplus L_7 \\ L'_5 \leftarrow L_5 \oplus L_6 \oplus L_8 \\ L'_6 \leftarrow L_6 \oplus L_7 \\ L'_7 \leftarrow L_7 \oplus L_8 \\ L'_8 \leftarrow L_8 \end{matrix} ,$$

which yields, after execution:

$$\begin{pmatrix} 1&0&0&0&0&0&0&0 & 1&0&0&1&1&1&1&0 \\ 0&1&0&0&0&0&0&0 & 0&1&0&0&1&1&1&1 \\ 0&0&1&0&0&0&0&0 & 1&1&0&0&1&1&0&0 \\ 0&0&0&1&0&0&0&0 & 0&1&1&0&0&1&1&0 \\ 0&0&0&0&1&0&0&0 & 0&0&1&1&0&0&1&1 \\ 0&0&0&0&0&1&0&0 & 1&1&1&1&0&0&1&0 \\ 0&0&0&0&0&0&1&0 & 0&1&1&1&1&0&0&1 \\ 0&0&0&0&0&0&0&1 & 1&1&0&1&0&1&1&1 \end{pmatrix} \begin{matrix} L'_1 = \texttt{0x80} \parallel \texttt{0x9e} \\ L'_2 = \texttt{0x40} \parallel \texttt{0x4f} \\ L'_3 = \texttt{0x20} \parallel \texttt{0xcc} \\ L'_4 = \texttt{0x10} \parallel \texttt{0x66} \\ L'_5 = \texttt{0x08} \parallel \texttt{0x33} \\ L'_6 = \texttt{0x04} \parallel \texttt{0xf2} \\ L'_7 = \texttt{0x02} \parallel \texttt{0x79} \\ L'_8 = \texttt{0x01} \parallel \texttt{0xd7} \end{matrix} ,$$

that has the expected form $(I_8 \quad G4)$. The bijection $F4 : x \mapsto G4 \times x$ is the optimal linear one for $n = 8$.

For $n = 4$, the bound $d \leq n - 1$ is met, but not for $n = 8$, since the best $d = 4$ is at distance 3 from $n - 1 = 7$.

We note that C' is a permuted code of the dual $C^\perp$ of C, obtained by swapping the leftmost half of the codewords (*i.e.* b) with the rightmost half (*i.e.* $\bigoplus_{i=1}^{n} b_i v_i$).

[2]This code is a subcode of the BCH $[17, 9, 5]$ code. For more details, please refer to: http://www.math.colostate.edu/~betten/research/codes/BOUNDS/sub_16_8_5-7_2.code.

Indeed, C and C' have the same dimension n hence $C^\perp$ and C' have the same dimension n, and the scalar product between $(\bigoplus_{i=1}^n b_i v_i, b)$ and $(x, F(x))$ is equal to $(\bigoplus_{i=1}^n b_i v_i) \cdot x \oplus b \cdot F(x) = b \cdot F(x) \oplus b \cdot F(x) = 0$. Thus, finding the largest dual distance of C is equivalent to finding the largest direct distance of C'.

Incidentally, when no bijection is used (*i.e.* like for the genuine masking [WW04]), F is the identity (hence a linear function) and C' is the repetition code. This code is autodual ($C'^\perp = C'$) and furthermore $C = C'$ (because the generating matrix $(I_n \quad I_n)$ is invariant under left-right halves exchange). Its minimal distance is $\delta = 2$, and thus the maximal resistance order is $d = 2 - 1 = 1$, as expected.

Now, the minimal order d for leakage squeezing with CIS codes (linear or not) is $\delta = 2$. Indeed the distance between two different codewords $(x, F(x))$ and $(y, F(y))$ is $\mathsf{HW}(x \oplus y) + \mathsf{HW}(F(x) \oplus F(y)) \geq 1+1 = 2$ because the code has two complementary information sets. As a consequence, the minimal order d for CIS codes is $\delta_{\min} \geq 2$. As it is equal to 2 for the identity, the protection order is exactly at least 1. This worst case for the security is thus attained when the leakage squeezing is not used, which positively motivates for its usage.

10.3.3 Optimal Non-Linear Bijections

Under some circumstances, a non-linear bijection F allows to reach better performances. There is no non-linear code for $n = 4$ that has a better dual distance than linear codes of the same length and size, but there are some for $n = 8$. A non-linear optimal code for $n = 8$ is the Nordstrom-Robinson $(16, 256, 6)$ code (that is also CIS, as discussed in details in [CGKS12, Example III.4]). With these parameters, this code coincides with Preparata and Kerdock codes [Sno73] and has same minimum distance and dual distance. Some codewords, as obtained from Golay code in standard form [FST92], are listed in Tab. 10.3.

Table 10.3: Some codewords of the Nordstrom-Robinson $(16, 256, 6)$ code.

Bit index	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Codeword $x = 0$	0	0	0	0	0	0	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Codeword $x = 1$	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Codeword $x = 2$	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Codeword $x = 3$	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	1	1	1	1	1	1	1	1
Codeword $x = 4$	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
Codeword $x = 5$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Codeword $x = 6$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Codeword $x = 7$	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Codeword $x = 8$	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Codeword $x = 254$	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	1
Codeword $x = 255$	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1

It happens that the code cannot be trivially split into two halves that each fill

exactly $\mathbb{F}_2^n$. Indeed, if the codewords are partitioned with bits $[\![15, 8]\!]$ on the one hand, and bits $[\![7, 0]\!]$ on the other,

- then $(11111111)_2$ — also noted `0xff` in the sequel — is present (at least) twice in the first half (from the high byte of codewords $x = 3$ and $x = 7$),
- and $(00000000)_2$ — also noted `0x00` in the sequel — is present (at least) twice in the second half (from the low byte of codewords $x = 0$ and $x = 7$).

We tested all the $\binom{16}{8}$ partitionings. For 2760 of them, the code can be cut in two bijections F_{high} and F_{low} of $\mathbb{F}_2^8$. This means that if we note $x \in \mathbb{F}_2^8$ the codewords index in Tab. 10.3, the Nordstrom-Robinson $(16, 256, 6)$ code writes as $F_{\text{high}}(x) \,||\, F_{\text{low}}(x)$. The codewords can be reordered according to the first column, so that the code rewrites $x \,||\, F_{\text{low}}(F_{\text{high}}^{-1}(x))$ [CGKS12]. So the bijection F can be chosen equal to $F = F_{\text{low}} \circ F_{\text{high}}^{-1}$. For example, when F_{high} consists in bits $[\![15, 9]\!] \cup \{7\}$ of the code (and F_{low} in bits $\{8\} \cup [\![6, 0]\!]$), F takes the values tabulated in Tab. 10.4.

Thus byte-oriented cryptographic implementations can be protected with this code against all HO-CPA of order $d \leq 5$.

10.3.4 Cost Optimality of the Leakage Squeezing

The leakage squeezing CM can be generalized to any injective (n, m)-function F, where $m \geq n$. The corresponding hardware architecture is depicted in Fig. 10.3. For instance, the first-order leakage-free CM presented in Chapter 8 also uses a mask size greater than that of the sensitive variable to protect.

In terms of codes, relaxing F from a bijection to an injection means that codes of rates smaller than $1/2$ are also eligible. For linear codes (*i.e.* linear F functions), this is tantamount to saying that there exists an information set I such that the restriction of the code to the complement of I is of same dimension as the code.

Unfortunately, this strategy does not bring any improvement. Indeed, codes $\{(x, F(x)); x \in \mathbb{F}_2^n\} \subseteq \mathbb{F}_2^{n+m}$ can have a greater direct distance when m increases (for instance by padding the code with new columns), but in the meantime their dual distance decreases. Consequently the cost of the CM increases with m, while the security of the masking scheme decreases.

The best situation is thus to have m minimal, *i.e.* $m = n$. The leakage squeezing CM initially presented (in Fig. 10.2) is thus optimal.

10.4 Security and Leakage Evaluations of the Optimal Linear and Non-Linear Bijections

As argued in [SMY09], the robustness evaluation of a CM encompasses two dimensions: its resistance to specific attacks, and its amount of leakage irrespective of any attack strategy. Indeed, a CM could resist some attacks, but still be vulnerable to others. For instance, in our study, we have focused on HO-CPA, but we have disregarded other attacks, such as MIA attack [BGP+11] or attacks based on generic

Table 10.4: Truth table for the found non-linear bijection.

$$\left\{F(x); x \in \mathbb{F}_2^8\right\} =$$

```
{ 0x00, 0xb3, 0xe5, 0x6a, 0x2f, 0xc6, 0x5c, 0x89,
  0x79, 0xac, 0x36, 0xdf, 0x9a, 0x15, 0x43, 0xf0,
  0xcb, 0x1e, 0xb8, 0x51, 0x72, 0xfd, 0x97, 0x24,
  0xd4, 0x67, 0x0d, 0x82, 0xa1, 0x48, 0xee, 0x3b,
  0x9d, 0x74, 0xd2, 0x07, 0xe8, 0x5b, 0x31, 0xbe,
  0x4e, 0xc1, 0xab, 0x18, 0xf7, 0x22, 0x84, 0x6d,
  0xa6, 0x29, 0x7f, 0xcc, 0x45, 0x90, 0x0a, 0xe3,
  0x13, 0xfa, 0x60, 0xb5, 0x3c, 0x8f, 0xd9, 0x56,
  0x57, 0xd8, 0x8e, 0x3d, 0xb4, 0x61, 0xfb, 0x12,
  0xe2, 0x0b, 0x91, 0x44, 0xcd, 0x7e, 0x28, 0xa7,
  0x6c, 0x85, 0x23, 0xf6, 0x19, 0xaa, 0xc0, 0x4f,
  0xbf, 0x30, 0x5a, 0xe9, 0x06, 0xd3, 0x75, 0x9c,
  0x3a, 0xef, 0x49, 0xa0, 0x83, 0x0c, 0x66, 0xd5,
  0x25, 0x96, 0xfc, 0x73, 0x50, 0xb9, 0x1f, 0xca,
  0xf1, 0x42, 0x14, 0x9b, 0xde, 0x37, 0xad, 0x78,
  0x88, 0x5d, 0xc7, 0x2e, 0x6b, 0xe4, 0xb2, 0x01,
  0xfe, 0x4d, 0x1b, 0x94, 0xd1, 0x38, 0xa2, 0x77,
  0x87, 0x52, 0xc8, 0x21, 0x64, 0xeb, 0xbd, 0x0e,
  0x35, 0xe0, 0x46, 0xaf, 0x8c, 0x03, 0x69, 0xda,
  0x2a, 0x99, 0xf3, 0x7c, 0x5f, 0xb6, 0x10, 0xc5,
  0x63, 0x8a, 0x2c, 0xf9, 0x16, 0xa5, 0xcf, 0x40,
  0xb0, 0x3f, 0x55, 0xe6, 0x09, 0xdc, 0x7a, 0x93,
  0x58, 0xd7, 0x81, 0x32, 0xbb, 0x6e, 0xf4, 0x1d,
  0xed, 0x04, 0x9e, 0x4b, 0xc2, 0x71, 0x27, 0xa8,
  0xa9, 0x26, 0x70, 0xc3, 0x4a, 0x9f, 0x05, 0xec,
  0x1c, 0xf5, 0x6f, 0xba, 0x33, 0x80, 0xd6, 0x59,
  0x92, 0x7b, 0xdd, 0x08, 0xe7, 0x54, 0x3e, 0xb1,
  0x41, 0xce, 0xa4, 0x17, 0xf8, 0x2d, 0x8b, 0x62,
  0xc4, 0x11, 0xb7, 0x5e, 0x7d, 0xf2, 0x98, 0x2b,
  0xdb, 0x68, 0x02, 0x8d, 0xae, 0x47, 0xe1, 0x34,
  0x0f, 0xbc, 0xea, 0x65, 0x20, 0xc9, 0x53, 0x86,
  0x76, 0xa3, 0x39, 0xd0, 0x95, 0x1a, 0x4c, 0xff }.
```

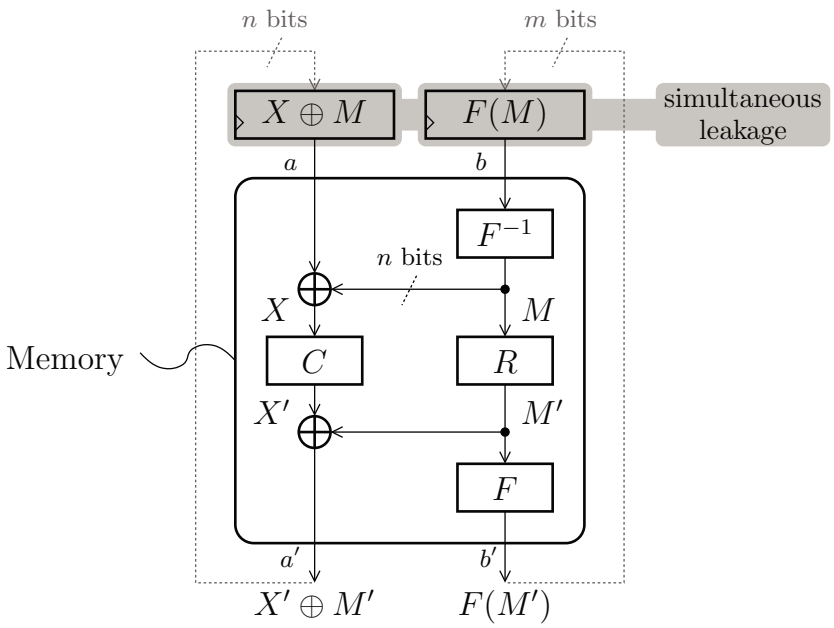


Figure 10.3: Generalization of the leakage squeezing countermeasure to any injective function $n \to m$.

side channel distinguishers [VCS11]. Therefore, in addition to a security evaluation conducted in Sec. 10.4.1, we will also estimate the leakage of the CM in Sec. 10.4.2.

10.4.1 Verification of the Security for $n = 8$

In this section, we illustrate the efficiency of the identified bijection from an HO-CPA point of view. We focus more specifically on the $n = 8$ bit case, because of its applicability to AES. We compute the values of $f_{\text{opt}}(z)$ for the centered leakage raised at power $1 \leq d \leq 6$ for four linear bijections (noted $F1$, $F2$, $F3$ and $F4$) and the non-linear bijection given in Sec. 10.3.3 (noted $F5$). The linear functions are defined from their matrix:

- $G1$ is the identity I_8, *i.e.* the Boolean masking function without F;
- $G2$ is a matrix that allows second-order resistance and is found without method;
- $G3$ is the circulant matrix involved in the AES block cipher;
- $G4$ is non-systematic half of the $[16, 8, 5]$ code matrix (see Eqn. (10.14)).

The $G2$, $G3$ and $G4$ matrices are:

$$G2 = \begin{pmatrix} 0&0&0&0&0&1&1&1\\ 0&0&0&1&1&0&1&1\\ 1&0&1&0&0&0&1&1\\ 0&0&1&1&1&0&0&0\\ 1&1&1&0&0&0&0&0\\ 1&1&1&1&1&1&1&1\\ 1&0&1&0&1&1&0&0\\ 0&1&0&1&0&1&1&0 \end{pmatrix}, \quad G3 = \begin{pmatrix} 1&0&0&0&1&1&1&1\\ 1&1&0&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&1&0&0&0&1\\ 1&1&1&1&1&0&0&0\\ 0&1&1&1&1&1&0&0\\ 0&0&1&1&1&1&1&0\\ 0&0&0&1&1&1&1&1 \end{pmatrix},$$

$$G4 = \begin{pmatrix} 1&0&0&1&1&1&1&0\\ 0&1&0&0&1&1&1&1\\ 1&1&0&0&1&1&0&0\\ 0&1&1&0&0&1&1&0\\ 0&0&1&1&0&0&1&1\\ 1&1&1&1&0&0&1&0\\ 0&1&1&1&1&0&0&1\\ 1&1&0&1&0&1&1&1 \end{pmatrix}.$$

It can be checked that they are invertible. Namely, their inverses are:

$$G2^{-1} = \begin{pmatrix} 0&1&1&1&0&0&0&0\\ 1&1&1&0&0&1&0&0\\ 1&0&0&1&1&1&0&0\\ 0&0&1&0&1&1&1&0\\ 1&0&1&0&0&0&1&0\\ 0&1&0&0&1&1&0&0\\ 1&0&0&0&0&1&1&1\\ 0&1&0&0&1&0&1&1 \end{pmatrix}, \quad G3^{-1} = \begin{pmatrix} 0&0&1&0&0&1&0&1\\ 1&0&0&1&0&0&1&0\\ 0&1&0&0&1&0&0&1\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&0&1&0\\ 0&0&1&0&1&0&0&1\\ 1&0&0&1&0&1&0&0\\ 0&1&0&0&1&0&1&0 \end{pmatrix},$$

$$G4^{-1} = \begin{pmatrix} 1&1&1&0&1&0&1&1\\ 1&0&0&1&1&1&1&0\\ 0&1&0&0&1&1&1&1\\ 1&1&0&0&1&1&0&0\\ 0&1&1&0&0&1&1&0\\ 0&0&1&1&0&0&1&1\\ 1&1&1&1&0&0&1&0\\ 0&1&1&1&1&0&0&1 \end{pmatrix}.$$

Table 10.5 reports some values of the optimal functions. The lines represented in gray are those for which the $f_{\text{opt}}(z)$ are the same for all the values of the sensitive variable $z \in \mathbb{F}_2^n$. For the sake of clarity, we represent only $n+1$ values of z, *i.e.* one per value of $\mathsf{HW}(z)$. But we are aware that unlike in the case where $F = \mathsf{Id}$, the optimal functions are not invariant in the bits reordering of z. If the line d is represented in gray, then a d^{th}-order HO-CPA cannot succeed. The last column shows the optimal correlation coefficient $\rho_{\text{opt}} = \sqrt{\frac{\mathsf{Var}[\mathbb{E}[(L-\mathbb{E}[L])|Z]]}{\mathsf{Var}[(L-\mathbb{E}[L])]}}$ that an attacker can expect [PRB09, Eqn. (15)].

It can be seen that the first nonzero ρ_{opt} approximately decreases with the CM strength: it is about 25% for $F1$, about 4% for $F2$ and $F3$, and about 2% for $F4$ and $F5$. The table shows that amongst the linear functions, $F4 : x \mapsto G4 \times x$ is indeed the best, since it protects against HO-CPA of orders 1, 2, 3 and 4. It can also be seen that the non-linear function $F5$ further protects against 5^{th}-order HO-CPA, as announced in Sec. 10.3.3.

Table 10.5: Computation of $f_{\text{opt}}(z)$ for centered traces raised at several powers d, and optimal correlation coefficient ρ_{opt}.

	$f_{\text{opt}}(z)$									ρ_{opt}
z	0x00	0x01	0x03	0x07	0x0f	0x1f	0x3f	0x7f	0xff	
Bijection $F = F1$ (reference $F1 : x \mapsto I_8 \times x = x$)										
$d = 1$	0	0	0	0	0	0	0	0	0	0.000000
$d = 2$	8	7	6	5	4	3	2	1	0	0.258199
$d = 3$	0	0	0	0	0	0	0	0	0	0.000000
$d = 4$	176	133	96	65	40	21	8	1	0	0.235341
$d = 5$	0	0	0	0	0	0	0	0	0	0.000000
$d = 6$	5888	3787	2256	1205	544	183	32	1	0	0.197908
Bijection $F = F2$ (linear $F2 : x \mapsto G2 \times x$)										
$d = 1$	0	0	0	0	0	0	0	0	0	0.000000
$d = 2$	4	4	4	4	4	4	4	4	4	0.000000
$d = 3$	−1.5	−1.5	−1.5	−1.5	0	0	0	0	1.5	0.036509
$d = 4$	49	49	49	49	49	46	49	46	46	0.015548
$d = 5$	−120	−75	−37.5	−30	7.5	22.5	15	22.5	67.5	0.051072
$d = 6$	1399	1061	949	971.5	971.5	821.5	971.5	821.5	979	0.027247
Bijection $F = F3$ (linear $F3 : x \mapsto G3 \times x$)										
$d = 1$	0	0	0	0	0	0	0	0	0	0.000000
$d = 2$	4	4	4	4	4	4	4	4	4	0.000000
$d = 3$	0	0	0	0	0	0	0	0	0	0.000000
$d = 4$	70	61	52	43	40	37	40	43	46	0.043976
$d = 5$	0	0	0	0	0	0	0	0	0	0.000000
$d = 6$	2584	1684	1144	694	544	484	544	694	664	0.067175
Bijection $F = F4$ (linear $F4 : x \mapsto G4 \times x$)										
$d = 1$	0	0	0	0	0	0	0	0	0	0.000000
$d = 2$	4	4	4	4	4	4	4	4	4	0.000000
$d = 3$	0	0	0	0	0	0	0	0	0	0.000000
$d = 4$	46	46	46	46	46	46	46	46	46	0.000000
$d = 5$	−90	−37.5	−15	15	7.5	−22.5	7.5	7.5	0	0.023231
$d = 6$	1339	956.5	799	799	866.5	821.5	776.5	821.5	844	0.016173
Bijection $F = F5$ (non-linear F tabulated in Sec. 10.3.3)										
$d = 1$	0	0	0	0	0	0	0	0	0	0.000000
$d = 2$	4	4	4	4	4	4	4	4	4	0.000000
$d = 3$	0	0	0	0	0	0	0	0	0	0.000000
$d = 4$	46	46	46	46	46	46	46	46	46	0.000000
$d = 5$	0	0	0	0	0	0	0	0	0	0.000000
$d = 6$	2104	1159	844	799	664	799	844	1159	844	0.023258

10.4.2 Verification of the Leakage of the Identified Bijections

As a complement to the security analysis carried out in Sec. 10.4.1, the leakage of the CM using the bijections $F1$, $F2$, $F3$, $F4$ and $F5$ is computed. It consists in the mutual information metric (MIM), defined as $\mathsf{I}[\mathsf{HW}(Z \oplus M'') + \mathsf{HW}(F(M) \oplus F(M \oplus M'')) - n + N; Z]$. The random variable N is an additive noise, that follows a normal law of variance σ^2. The result of the MIM computation is shown in Fig. 10.4. In the ordinate, the smaller the MIM, the more secure the CM. Now, there are at least two ways to interpret the abscissa:

1. **In terms of attacker budget:** an attacker who is able to develop advanced denoising filters and who can buy accurate side channel probes will be placed in the low noise areas (*i.e.* at the left-hand side of the graph).

2. **In terms of defender budget:** the designer can integrate more logic to increase the algorithmic noise, or he can even add artificial noise sources [GM11]; however, the more noise the designer wishes to inject in a view to obscure the leakage (*i.e.* at the right-hand side of the graph), the more area and power are required.

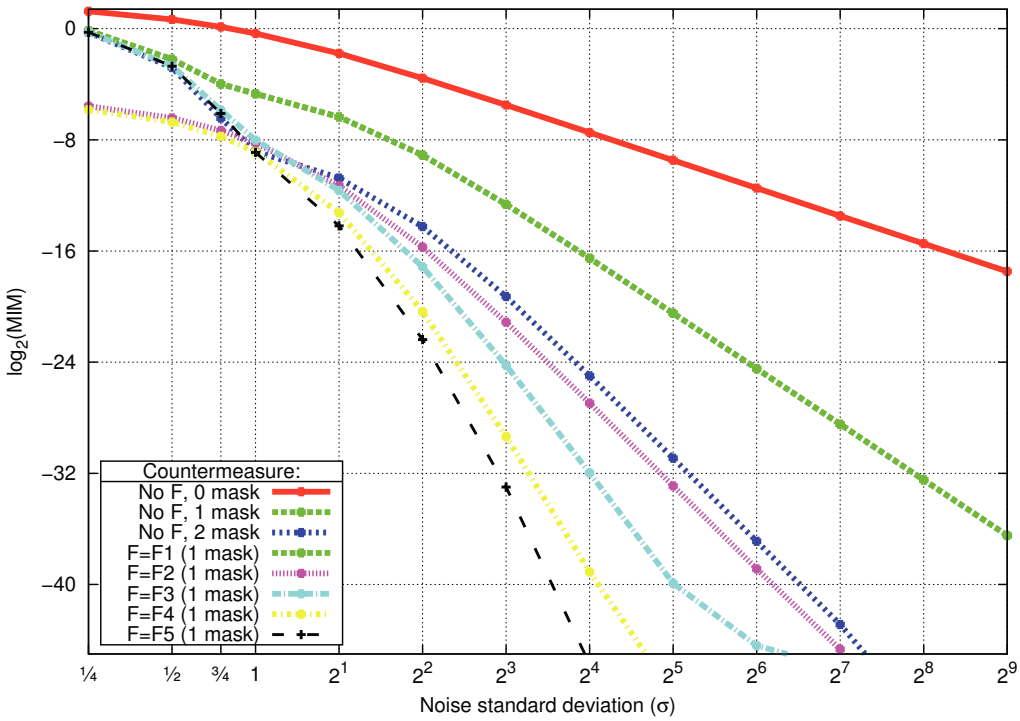


Figure 10.4: Mutual information of the leakage with the sensitive variable Z for $n = 8$ bits.

It appears that the leakage agrees with the strength of the CM against HO-CPA: the greater the order of resistance against HO-CPA, the smaller the mutual

information, at least for a reasonably large noise. This simulated characterization validates (in the particular scheme of Fig. 10.2) the relevance of choosing F based on a HO-CPA criterion.

Furthermore, Fig. 10.4 represents the leakage of a similar CM, where more than two shares would be used. More precisely, the shares would be the triple $(x \oplus m_1 \oplus m_2, m_1, m_2)$, where the masks m_i are not transformed by bijections. This CM is obviously more costly than our proposal of keeping one single mask, but passed through F. We notice that all the proposed bijections (suboptimal $F2$ and $F3$, optimal linear $F4$ and optimal non-linear $F5$) perform better, in that they leak less irrespective of σ.

10.4.3 Results in Imperfect Models

Masking schemes randomize more or less properly the leakage. In the straightforward example studied in this chapter (Eqn. (10.1) with $F = \mathsf{Id}$), when the sensitive variable z has all its bits equal to '1' (*i.e.* $Z = \texttt{0xff}$), then the mask has no effect whatsoever on the leakage. Indeed, this is due to a well-known property of the Hamming weight function: $\forall M'' \in \mathbb{F}_2^n, \mathsf{HW}(\texttt{0xff} \oplus M'') + \mathsf{HW}(M'') = \mathsf{HW}(\overline{M''}) + \mathsf{HW}(M'') = n$. To avoid this situation, the proposed CM based on the bijection F consists in tuning the leakage, so that the masks indeed dispatch randomly the leakage for most (if not all as for the leakage-free countermeasure) values of the sensitive data. The working factor of its improvement is the introduction of a specially crafted Boolean function F aiming at weakening the link between the data to protect and the leakage function.

This technique has been shown to be very effective in the previous sections when the analysis assumed a perfect leakage model. But the Hamming distance leakage model is in practice an idealization of the reality. Therefore, we study the impact of the leakage model imperfections on the leakage squeezing countermeasure. As stated in Chapter 8 Sec. 8.3, we assume that the leakage model is written as a multivariate polynomial in $\mathbb{R}[X_1, \cdots, X_n, X'_1, \cdots, X'_n]$ of degree less or equal to $\tau \in [\![1, 2n]\!]$. It takes the following form:

$$L \doteq P(X_1, \cdots, X_n, X'_1, \cdots, X'_n) = \sum_{\substack{(u,v) \in \mathbb{F}_2^n \times \mathbb{F}_2^n, \\ \mathsf{HW}(u)+\mathsf{HW}(v) \leq \tau}} C_{(u,v)} \cdot \prod_{i=1}^{n} X_i^{u_i} X'^{v_i}_i \ , \tag{10.15}$$

where the $C_{(u,v)}$ are real coefficients. This leakage formulation is similar to that of the higher-order stochastic model [SLP05]. In the following experiments, we compute the mutual information between $L = \mathsf{HW}(Z \oplus M \oplus M') + \mathsf{HW}(F(M) \oplus F(M')) + N$ and Z when $\tau \leq 3$ and when the coefficients $C_{(u,v)}$ deviate randomly from those of Eqn. (8.5) in Chapter 8 Sec. 8.3 or are completely random (*i.e.* deviate from a "Null" model). More precisely, the coefficients $C_{(u,v)}$ are respectively drawn at random from one of these laws:

$$\begin{array}{rcll} C^{\mathrm{HD}}_{(u,v)} & \sim & c^{\mathrm{HD}}_{(u,v)} & + \mathcal{U}([-\frac{\delta}{2}, +\frac{\delta}{2}]) \ , \\ C^{\mathrm{Null}}_{(u,v)} & \sim & 0 & + \mathcal{U}([-\frac{\delta}{2}, +\frac{\delta}{2}]) \ . \end{array} \tag{10.16}$$

Four bit variables (case useful for DES) are considered. The study is conducted on three bijections:

$F1'$: the identity (Id), that acts as a reference,

$F2'$: one bijection that cancels the first-order leakage but not the second-order,

$F3'$: another that cancels both first- and second-order leakage.

They are linear, *i.e.* write $Fi'(x) = Gi' \times x$, where the generating matrices Gi' are given below:

$$G1' = I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad G2' = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad G3' = \overline{I_4} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}. \tag{10.17}$$

In this section, we use bijections Fi' from $\mathbb{F}_2^4$ to $\mathbb{F}_2^4$, noted with a prime, to mark the difference with the bijections $Fi : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ that were studied in Sec. 10.4.1 and 10.4.2.

The results are plotted in Tab. 10.6, 10.7 & 10.8 for the randomized HD model and in Tab. 10.9, 10.10 & 10.11 for the randomized "Null" model.

The curves are represented for a noise standard deviation σ in the interval $[0, 5]$. We insist that the comparison between the different curves shall only be done when the noise is larger than the leakage of one sensitive bit, *i.e.* for $\sigma \geq 1$. This recommendation also applies to the interpretation of Fig. 10.4. Indeed, too low a noise is not realistic in practice.

In Tab. 10.6, 10.7 & 10.8, it can be seen that despite the HD model degradation, the leakage of the CM:

- remains ordered ($F3'$ leaks less than $F2'$, and $F2'$ in turn leaks less than $F1'$),
- and remains low, irrespective of δ.

The average leakage is unchanged, and the leakage values are simply getting slightly scattered. The reason for this resilience comes from the rationale of the CM: the masked value and the mask are decorrelated as much as possible. The dispatching is guided by a randomized pigeon-hole of the values in the image of the leakage function. The CM thus looses efficiency only in the case where two different values of leakage become similar due to the imperfection. This can happen for some variables, but it is very unlikely that it occurs coherently for all variables at the same time. Rather, given the way the imperfect model is built (Eqn. (10.16)), it is almost as likely that two classes get nearer or further away. This explains why, in average, the leakage is not affected: the model noise acts as a random walk, that has an impact on the variance but not on the average. Of course, some samples (with a degraded model) will be weaker than the others (because the variance of the MIA increases with the variance[3] $\delta^2/12$ of the model).

[3] The variance of a uniform law of amplitude δ is indeed equal to $\mathsf{Var}\left(\mathcal{U}([-\delta/2, +\delta/2])\right) = \frac{1}{\delta}\int_{-\delta/2}^{+\delta/2}(u-0)^2\,\mathrm{d}u = \left[\frac{u^3}{3\delta}\right]_{u=-\delta/2}^{u=+\delta/2} = \frac{\delta^2}{12}$.

Table 10.6: Leakage comparison of the leakage squeezing CM (*left column*) and the leakage-free CM (*right column*) in the imperfect HD leakage model ($\tau = 1$).

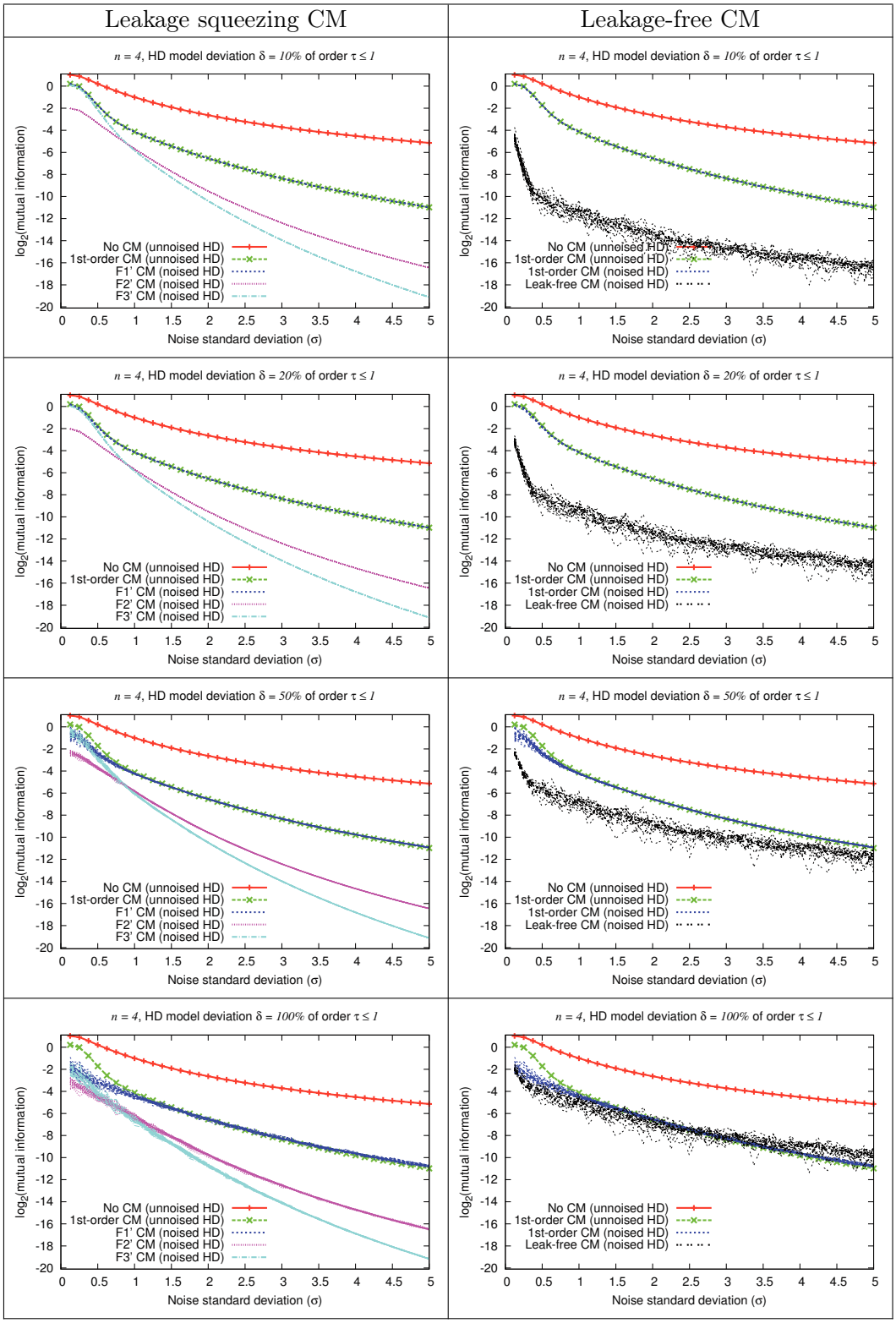


Table 10.7: Leakage comparison of the leakage squeezing CM (*left column*) and the leakage-free CM (*right column*) in the imperfect HD leakage model ($\tau = 2$).

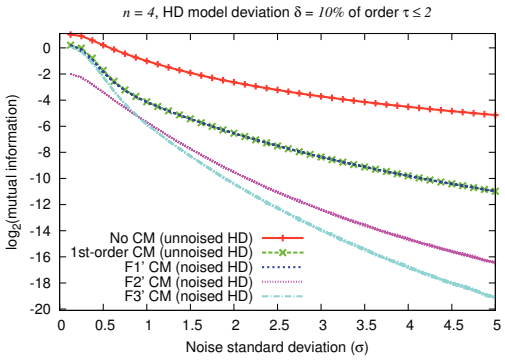
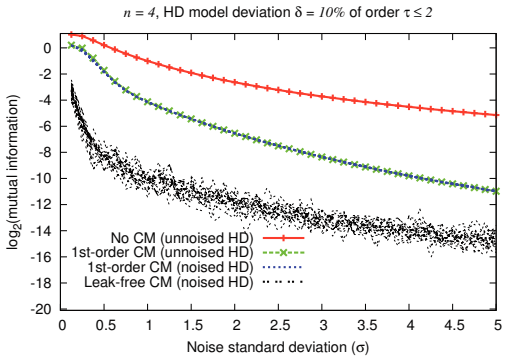
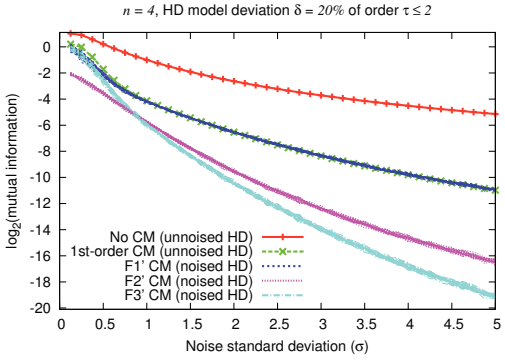
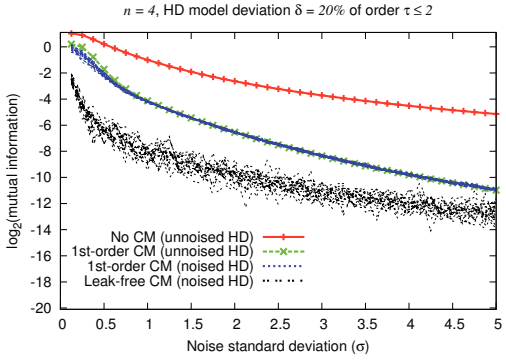
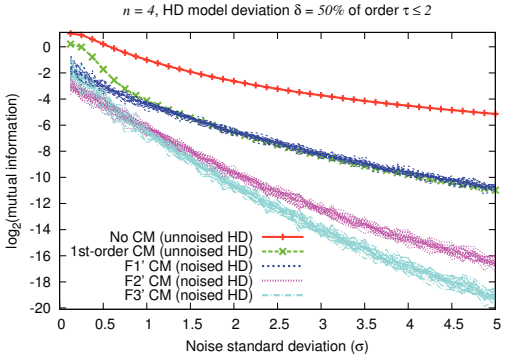
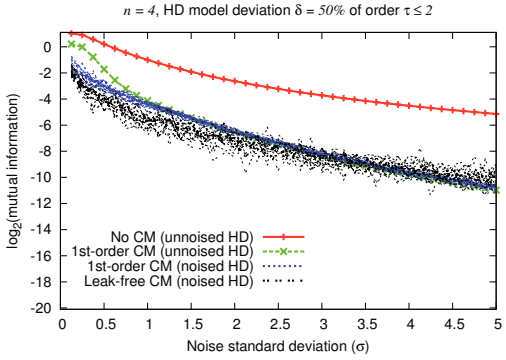
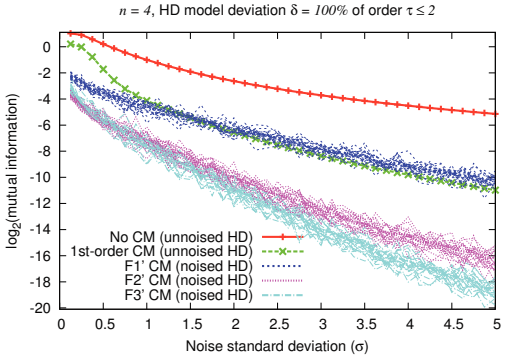
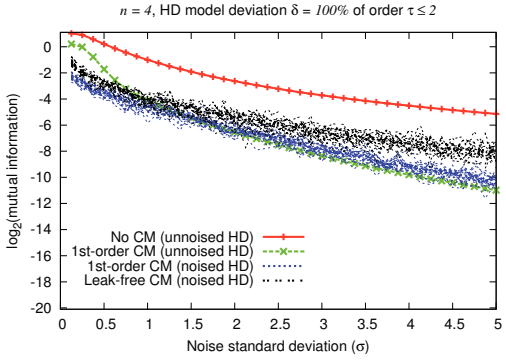
Leakage squeezing CM	Leakage-free CM
	
	
	
	

Table 10.8: Leakage comparison of the leakage squeezing CM (*left column*) and the leakage-free CM (*right column*) in the imperfect HD leakage model ($\tau = 3$).

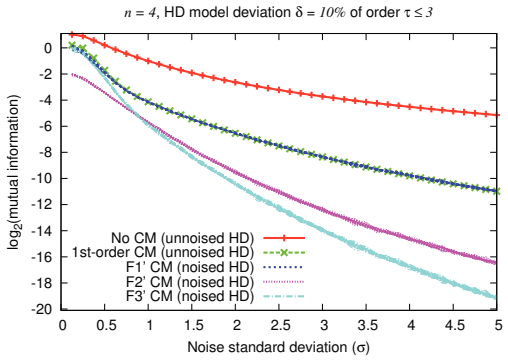
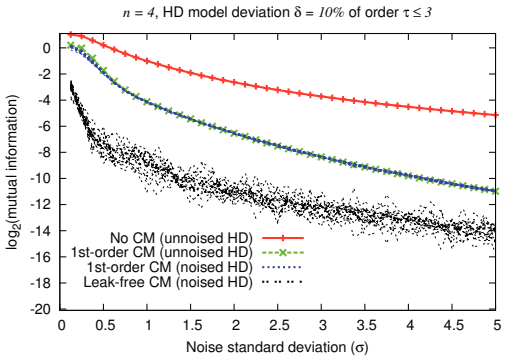
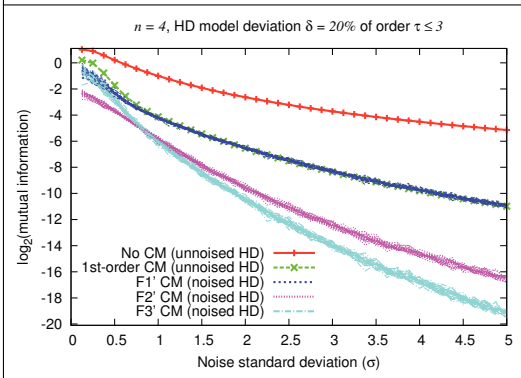
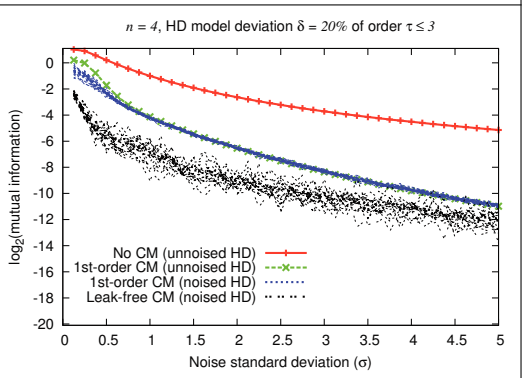
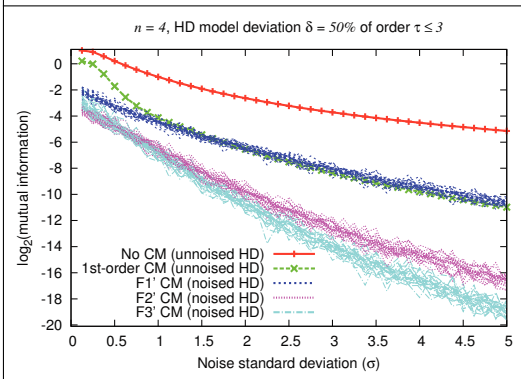
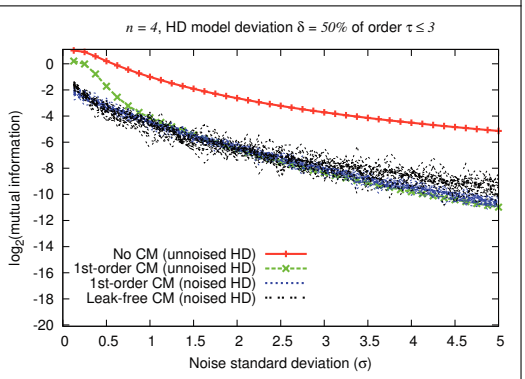
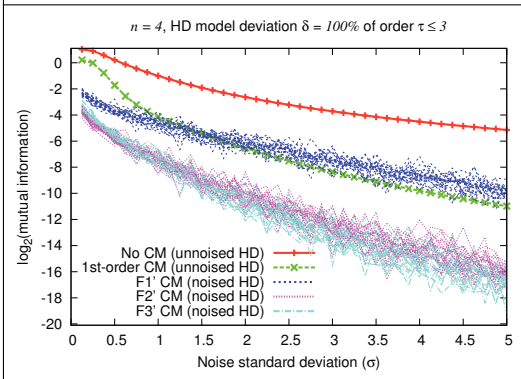
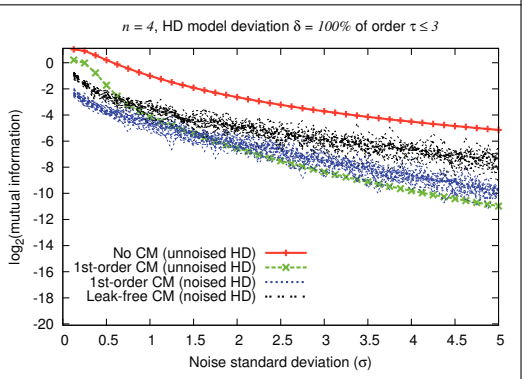
Leakage squeezing CM	Leakage-free CM
	
	
	
	

Table 10.9: Leakage comparison of the leakage squeezing CM (*left column*) and the leakage-free CM (*right column*) in the imperfect "NULL" leakage model ($\tau = 1$).

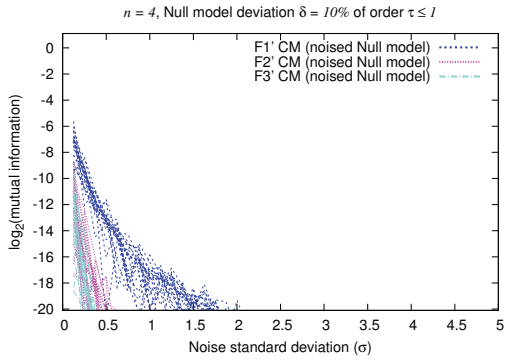
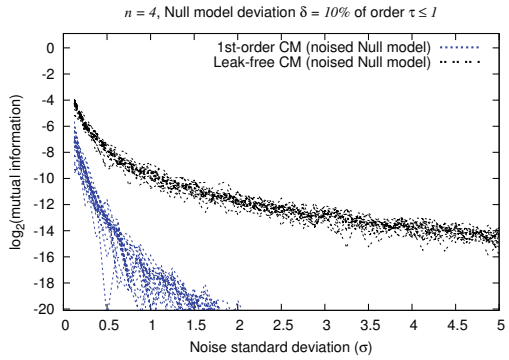
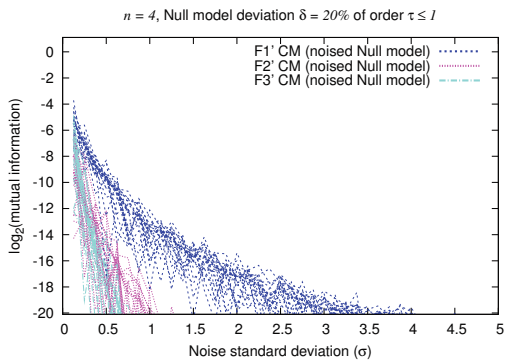
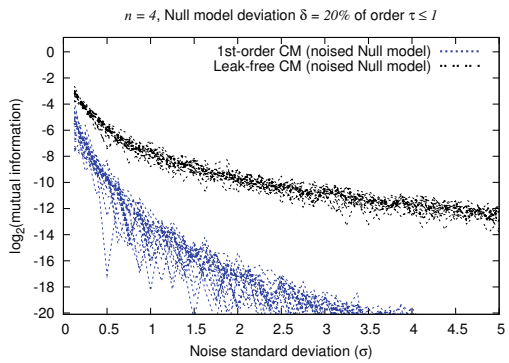
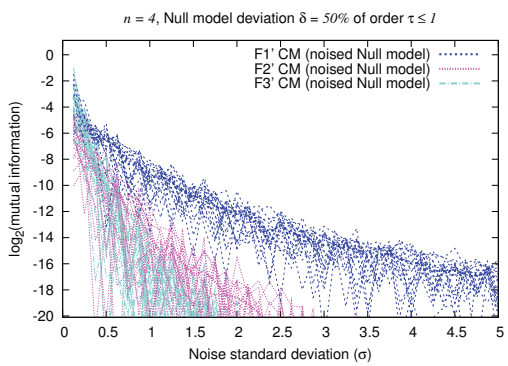
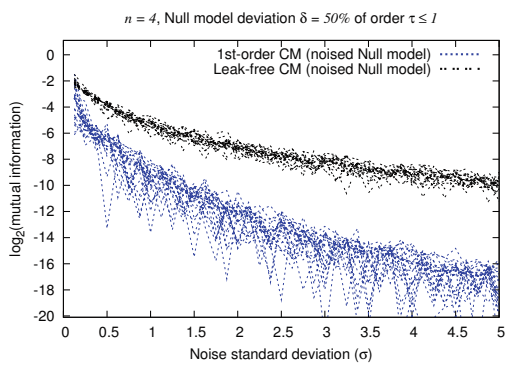
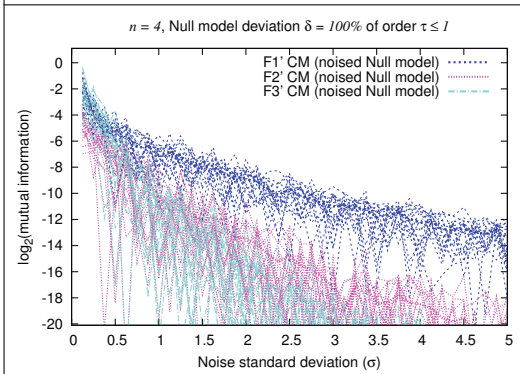
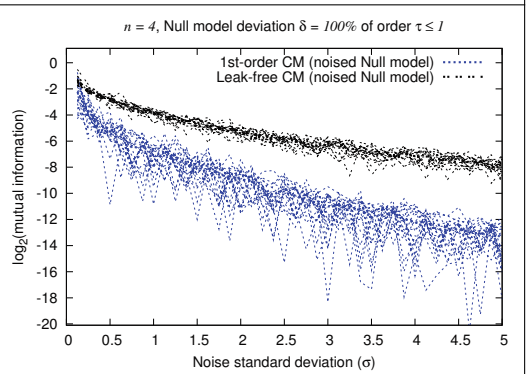
Leakage squeezing CM	Leakage-free CM
	
	
	
	

Table 10.10: Leakage comparison of the leakage squeezing CM (*left column*) and the leakage-free CM (*right column*) in the imperfect "NULL" leakage model ($\tau = 2$).

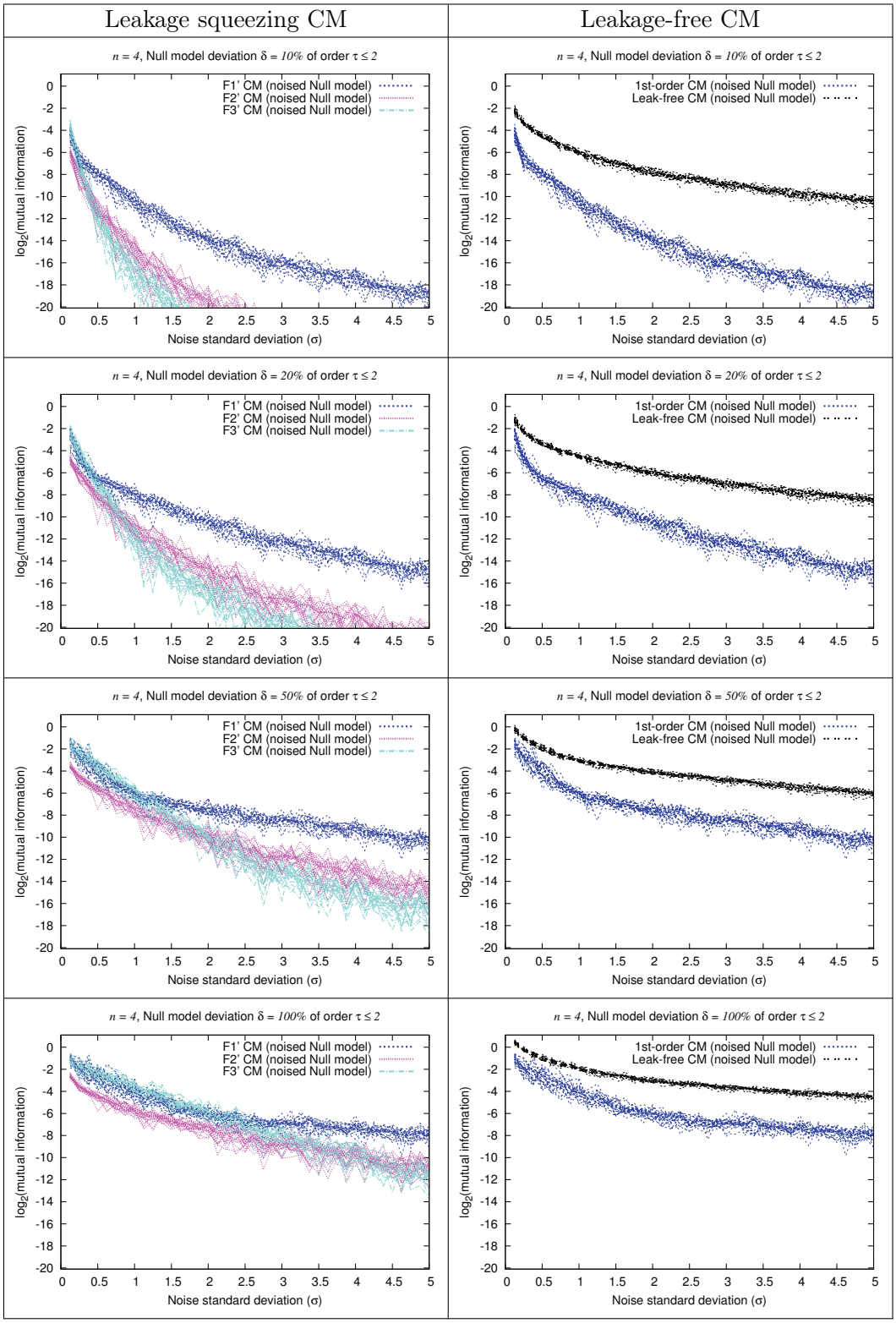
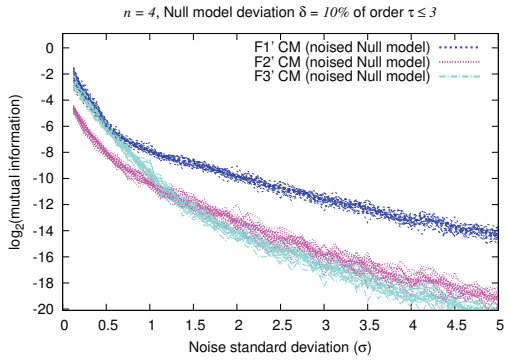
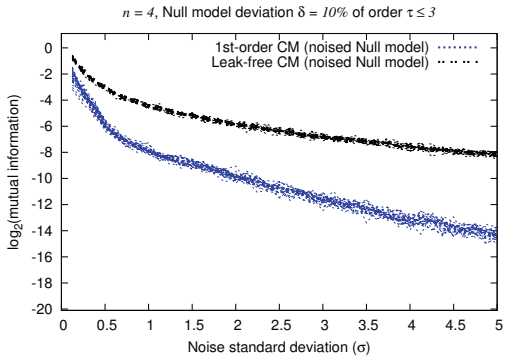
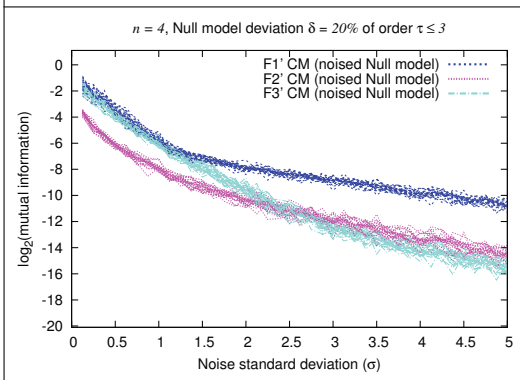
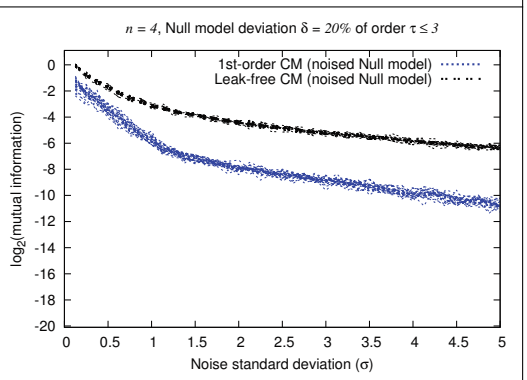
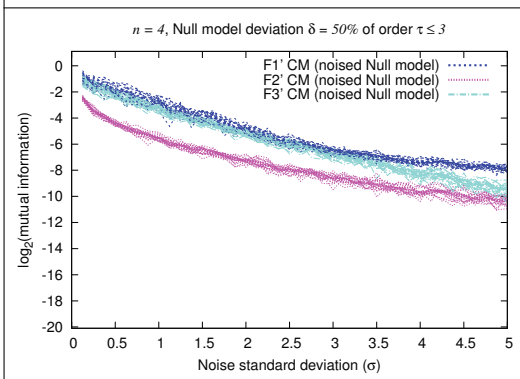
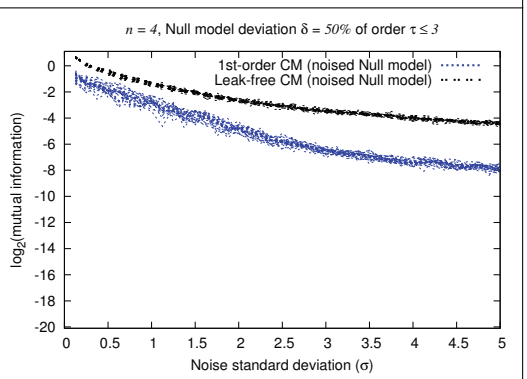
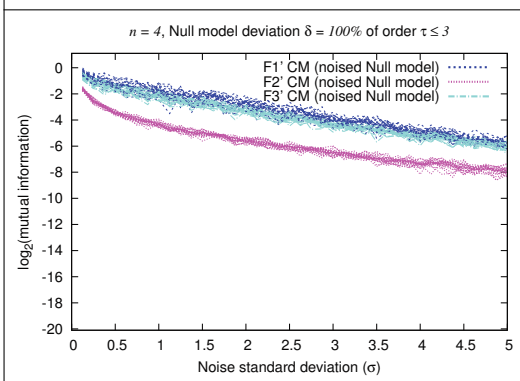
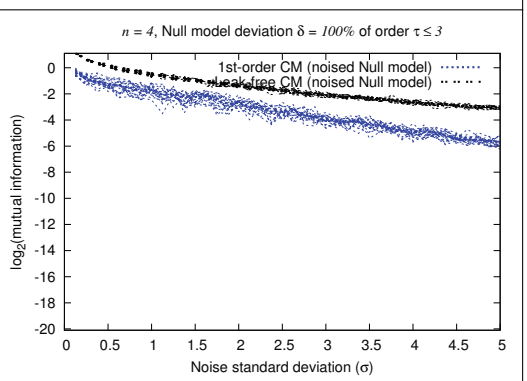


Table 10.11: Leakage comparison of the leakage squeezing CM (*left column*) and the leakage-free CM (*right column*) in the imperfect "NULL" leakage model ($\tau = 3$).

Leakage squeezing CM	Leakage-free CM
	
	
	
	

It is interesting to contrast the leakage squeezing with the first-order leakage-free CM presented in Chapter 8. This CM aims at leaking no information when the HD leakage model is perfect. A study for model imperfection has also been conducted (see right column of Tab. 10.6, 10.7 & 10.8). It appears that this CM is much less robust to deviations from the ideal model. Indeed, the working factor of the CM is to have one share leak nothing. But as soon as there is some imperfection, the very principle of the CM is violated, and it starts to function badly. Concretely the leaked information increases with the model variance, up to a point where the CM is less efficient than the straightforward first-order Boolean masking (starting from $\delta > 50\%$).

For the sake of comparison, we also computed the same curves when the unnoised model is a constant one (called "Null" model in Eqn. (10.16)). The simulation results are shown in Tab. 10.9, 10.10 & 10.11. The reference leakage (when $\delta = 0$) is null; consequently only the noisy curves are shown. It is noticeable that despite this "Null" leakage model is random, the different CMs have clearly distinguishable efficiencies. This had already been noticed by Doget *et al.* in [DPRS11]. In particular, it appears that our CM (the leakage squeezing) continues to work ($F3$ leaks less than $F2$, that leaks less than $F1$), at least for large enough noise standard deviations σ. On the contrary, the leakage-free CM is not resilient to this random model: it leaks more than the straightforward masking (*i.e.* with $F1$).

Eventually, the impact of the leakage degree τ can be studied. Results are computed for τ in $\{1, 2, 3\}$. In all the cases, τ does not impact the general conclusions. Regarding the deviation from the HD model, the greater the multivariate degree τ, the more possible deviations from the genuine ideal model. Indeed, the number of random terms in Eqn. (10.15) is increasing with τ (and is equal to $\sum_{t=0}^{\tau} \binom{2n}{t}$). This explains the greatest variability in the mutual information results. In the meantime, the argumentation for the robustness of the CM against the model deviation still holds, which explains why the average leakage is unchanged. In the "Null" model, the greater τ, the less singularities in the leakage. This explains why the mutual information curves get smoother despite the additional noise. But with higher τ, the leaking sources are higher (because the more non-zero terms in the polynomial), which explains why the leaked mutual information increases in average with τ.

10.5 FPGA Implementation of the Leakage Squeezing Countermeasure

In this section, we apply the principle of leakage squeezing to the masked DES algorithm. Two implementations are proposed: a GLUT based architecture and a simpler structure using the USM architecture.

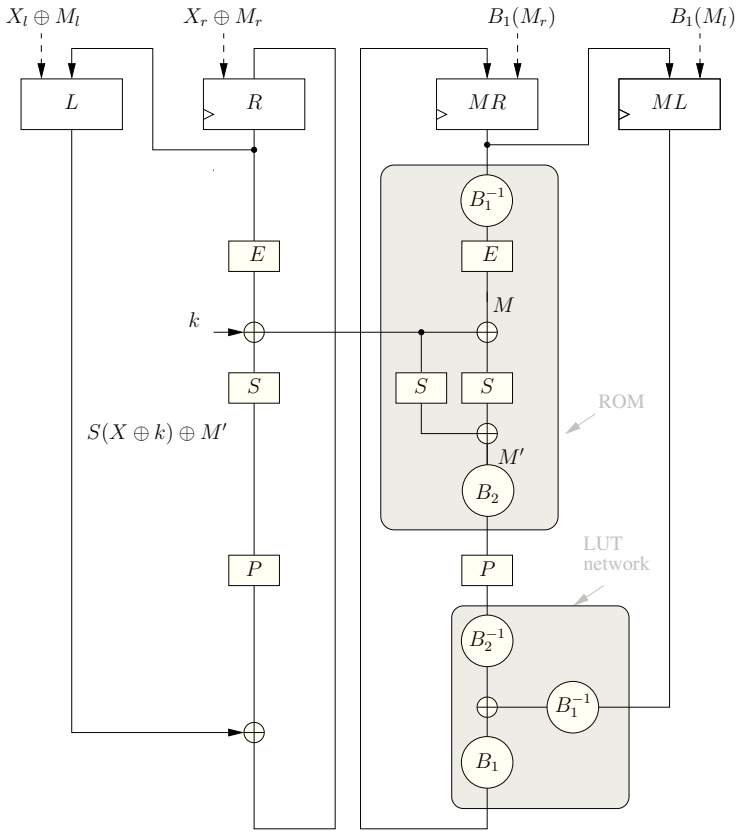


Figure 10.5: Leakage squeezing of DES with a masked ROM implementation.

10.5.1 The GLUT Hardware Implementation

For DES we can use eight different bijections[4], denoted B_1, one for each S-box. To further protect the new mask M', we compose the DES parts by using external encodings with bijection B_2, for instance:

$$\underbrace{B_1^{-1} \circ E \circ S \circ P \circ \texttt{XOR}(L) \circ B_1}_{\text{ROM}} = \underbrace{B_1^{-1} \circ E \circ S \circ B_2}_{\text{ROM}} \circ P \circ \underbrace{B_2^{-1} \circ \texttt{XOR}(L) \circ B_1}_{\text{LUT network}}, \tag{10.18}$$

where B_1 and B_2 are 4-bit bijections, E, S, P and $\texttt{XOR}(L)$ are respectively the expansion, S-Box, permutation and left part recombination of the DES algorithm. This principle of internal encodings has already been proposed by Chow *et al.* in [CEJvO02] in the context of *white box cryptography*. This protection method has already been attacked for the DES [WMGP07] and for the AES [BGEC04]. However, these attacks should not apply for the mask path as it is random and consequently no values can be imposed at the table inputs.

The general GLUT hardware implementation is given in Fig. 10.5. We choose

[4]The same bijection can be reused eight times without compromising the security.

the bijection B_1 to be equal to $F3'$ studied in Sec. 10.4.3. The intermediate data (*e.g.* S-box outputs) have been protected by the same strategy, so as to provide a seamless "squeezing" throughout the combinational logic.

The bijection B_2 is constrained to be a XOR operation with a constant, as the permutation P on 32 bits causes the ROM output bits to be split for the next round. The implementation of the mixing L with the left part can be done by a LUT network in FPGAs rather than a ROM in order to reduce the complexity. If we compare this implementation to the one proposed in [SRQ06] and described in Fig. 3.2 of Chapter 3, we have the same ROM complexity which is of eight 2^8 words of four bits.

10.5.2 The USM Hardware Implementation

The GLUT implementation can be replaced by a more simple structure which is the USM masking scheme. Figure 10.6 illustrates the mask path of DES with USM implementation taking advantage of the leakage squeezing method. It is made up of four stages which can be protected by using bijections B_1, B_2, B_3 and B_4. All the bijection are on four bits except B_2 which is on six bits.

Every stage can be implemented by a set of LUT networks or a ROM. The bijection B_4 is constrained to be a XOR operation with a constant, as the permutation P on 32 bits causes the output bits to split. In order to enhance the implementation security, the bijections should be refreshed regularly by using RAMs rather than ROMs or by using partial reconfiguration as detailed in Chapter 9 Sec. 9.3.1.

10.5.3 Complexity and Throughput Results

The proposed implementations have been tested in a StratixII FPGA which is based on Adaptative LUT Module (ALM) cell. They have been compared with unprotected DES, masked GLUT and masked USM implementations of DES without any leakage squeezing. Table 10.12 summarizes the memories needed for each implementation and the estimated throughput.

Table 10.12: Complexity and speed results. "l. s." denotes the "leakage squeezing" countermeasure.

Implementation	**ALMs**	**Block memory [bit]**	**M4Ks**	**Throughput [Mbit/s]**
Unprotected DES (*reference*)	276	0	0	929.4
DES masked USM	447	0	0	689.1
DES masked ROM	366	131072	32	398.4
DES masked ROM with l. s.	408	131072	32	320.8
DES masked USM with l. s.	488	0	0	582.8

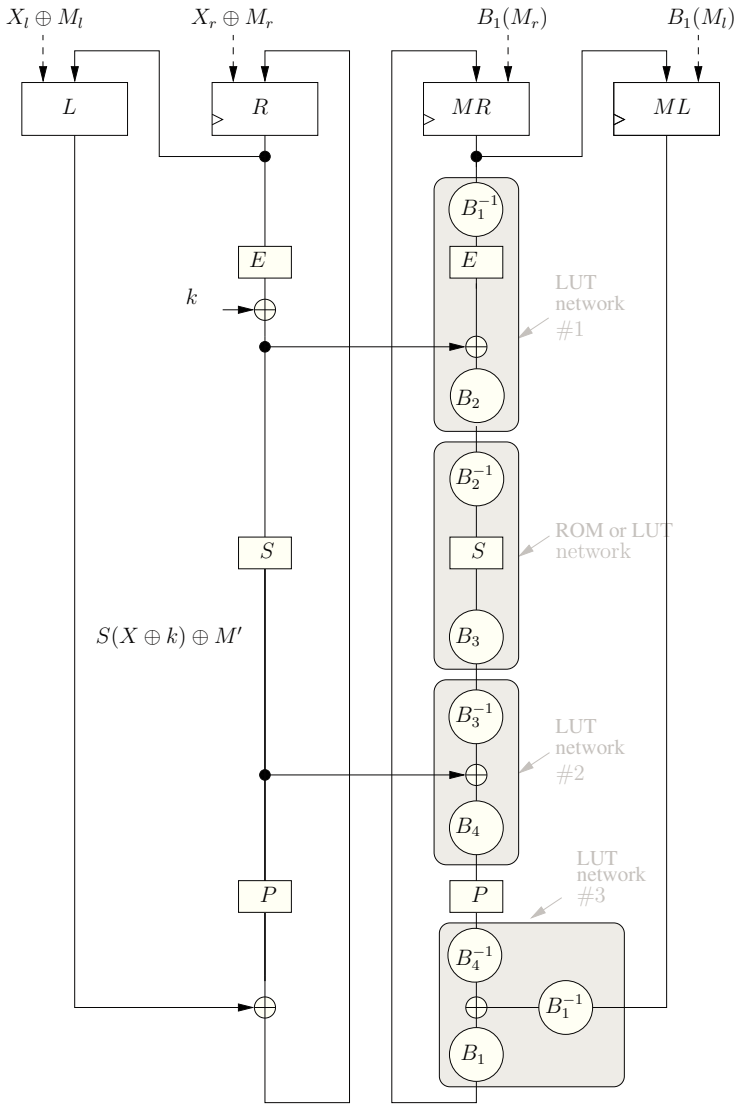


Figure 10.6: Leakage squeezing of DES with a masked USM implementation.

These results show that the leakage squeezing method on hardware implementations has little impact on complexity and speed compared with the straightforward first-order masking CM. Moreover, the USM implementation is particularly efficient as it avoids the use of large ROMs while keeping a high throughput.

In order to validate our implementations, we conduct in the next section a security evaluation of the proposed schemes.

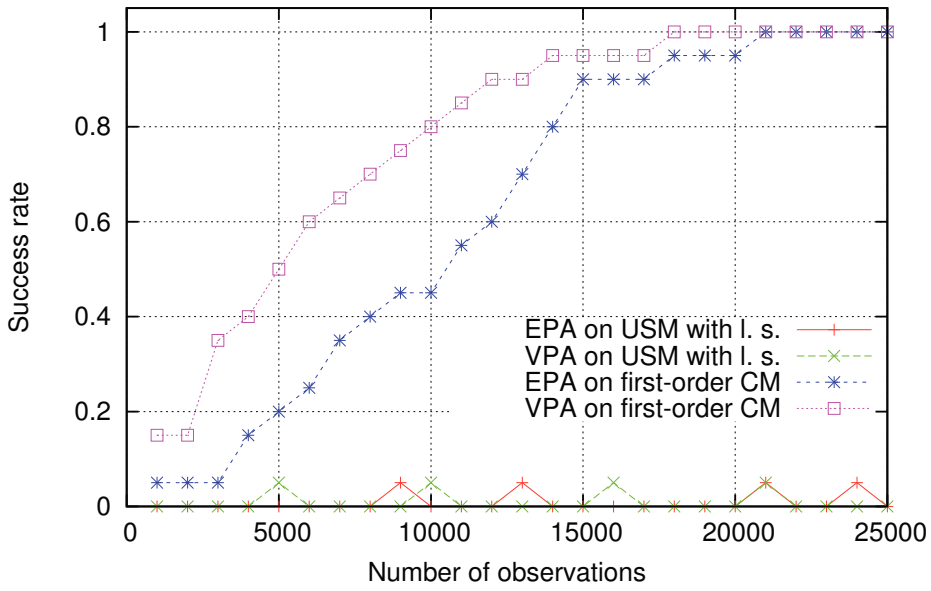


Figure 10.7: First-order success rate of 3 distinguishers.

10.5.4 Evaluation of the Implementations against Second-Order Attacks

The security evaluation of our leakage squeezing CM was made on a real implementation. We applied two side channel distinguishers to the leakage measurements: VPA and EPA attacks. For each scenario, we acquired a set of 25,000 power consumption traces using random masks and plaintexts and we performed the first-order success rate. We showed in Fig. 10.7 our experimental results also for these attacks on the first-order hardware masking implementation used here for comparison purposes with our hardware solution based on the leakage squeezing technique.

We can see that the attacks perform well when applied to the straightforward masking implementation. For the EPA, the success rates stay well above 50% even when using 11,000 measurements, but eventually reaches success rate of 95% using 18,000 traces. For our proposed countermeasure, the attacks perform worse. The success rates stay under 10% even when using 25,000 measurements. We conclude that the experiments on a real circuit shows the evidence of benefit of our countermeasure.

10.6 Conclusions

In this chapter, we showed that when we know that the device leaks in Hamming distance, the highest order d of a successful attack can be increased significantly

thanks to the "leakage squeezing". Its principle is to store $F(m)$ (the image of m by a bijection F) instead of m in the mask register. Typically, when the data to protect are bytes, the state-of-the-art implementations with one mask could be attacked with HO-CPA of order $d = 2$. We demonstrated how to find optimal linear F, that protects against HO-CPA of orders 1, 2, 3 and 4. We also proved that optimal non-linear functions F protect against HO-CPA of orders 1, 2, 3, 4 and 5. This security increase also translates into a leakage reduction. An information-theoretic study reveals that the mutual information between the leakage and the sensitive variable is lower than the same metric computed on a similar CM without F but that uses two masks (instead of one). Two implementations have been proposed and evaluated in a real-life context. They provide a great robustness against higher-order attacks as none of the subkeys have been guessed using 25K traces. Moreover, the performances decrease in terms of complexity and speed are very limited, which is particularly true for the USM implementation which does not require large memories.

In the next chapter, we extend the notion of leakage squeezing to the second-order context.

Chapter 11

Leakage Squeezing of Order Two

In this chapter, we study second-order leakage squeezing, *i.e.* leakage squeezing with two independent random masks. It is proved that, compared to first-order masking, second-order masking at least increments (by one unit) the resistance against higher-order attacks, such as HO-CPA attacks. Now, better improvements over first-order masking countermeasure are possible by relevant constructions of squeezing bijections. Specifically, optimal leakage squeezing with one mask resists HO-CPA of orders up to 5. In this chapter, with two masks, we provide resistance against HO-CPA not only of order $5 + 1 = 6$, but also of order 7.

The results presented in this chapter have been published in collaboration with Claude Carlet, Sylvain Guilley and Jean-Luc Danger in the international conference on Cryptology *IndoCrypt 2012* [CDGM12].

Contents

11.1 Reminder on First-Order Leakage Squeezing

11.1.1 Leakage Squeezing in the Hamming Distance Model

The principle of first-order leakage squeezing is sketched in Fig. 11.1. As opposed to straightforward first-order masking, the two registers contain $X \oplus M$ and $F(M)$,

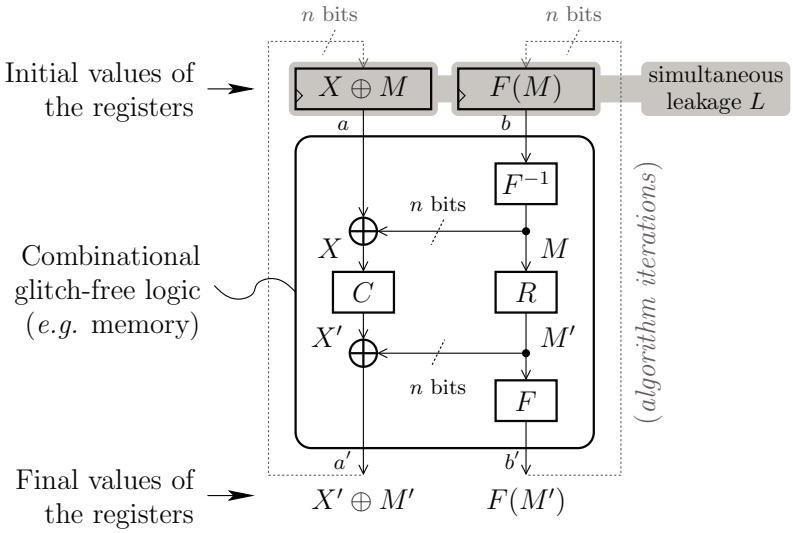


Figure 11.1: Setup of the first-order masking countermeasure with bijection F.

where the function F is a bijection. In Fig. 11.1, the computational logic is concealed in memory tables (to ensure a glitch-free computation). The scheme presented in Fig. 11.1 allows computing (X', M') from (X, M) in one clock cycle: $X' = C(X)$ and $M' = R(M)$, where C and R are respectively the round function and the mask refresh function.

In a hardware setup, the shares leak in the Hamming distance model. The leakage is thus equal to $L = \mathsf{HW}((X \oplus M) \oplus (X' \oplus M')) + \mathsf{HW}(F(M) \oplus F(M'))$, that can be rewritten as:

$$\begin{aligned} L &= \mathsf{HW}(Z \oplus M'') + \mathsf{HW}(F(M) \oplus F(M \oplus M'')) \\ &= \mathsf{HW}(Z \oplus M'') + \mathsf{HW}(D_{M''}F(M)) \ , \end{aligned}$$

where $Z = X \oplus X'$ and $M'' = M \oplus M'$. In the rest of this section, we recapitulate the key steps described extensively in last chapter to find the first-order optimal leakage squeezing. The section 11.2 will conduct step-by-step an accurate and self-contained analysis of the two-mask case.

It is shown in Chapter 10 that this leakage function is unexploitable by a d^{th}-order correlation power analysis if all the terms $\mathbb{E}[\mathsf{HW}(Z \oplus M'')^p \times \mathsf{HW}(D_{M''}F(M))^q \mid Z = z]$, whatever p, q such as $p + q \leq d$ do not depend on z. This equation can be simplified as Theorem 7 below, that was proved in Chapter 10 Sec. 10.2.4.

Theorem 7. $\forall a \in \mathbb{F}_2^n, \forall p \in \mathbb{N}, \quad \widehat{\mathsf{HW}^p}(a) = 0 \iff \mathsf{HW}(a) > p \ .$

So the condition for the leakage squeezing to reach order d is simply to have: for all $a \in \mathbb{F}_2^{n*}$ and for all p such that $\mathsf{HW}(a) \leq p$ and for all q such as $q \leq d - p$, $\mathbb{E}[\widehat{\mathsf{HW}^q \circ D_{(\cdot)}F(M)}](a) = 0$.

This condition is also equivalent to:

$$\forall p, \forall (a,b) \text{ such that } \mathsf{HW}(a) \leq p \text{ and } \mathsf{HW}(b) \leq d-q, \text{ we have } \widehat{(b \cdot F)}_{\chi}(a) = 0 \ .$$

As shown in Chapter 10 Sec. 10.3, this condition can be related to "complementary information set" codes (CIS codes [CGKS12]). It is equivalent that the indicator of the graph $\{(x, F(x)); x \in \mathbb{F}_2^n\}$ of F is d^{th}-order correlation immune.

11.1.2 Leakage Squeezing in the Hamming Weight Model

If the device leaks in Hamming weight, then the relations are still valid if we replace the derivative of F by F itself. It is also worthwhile mentioning that if F is linear, the two problems are the same, because $D_m F(x) = F(x \oplus m) \oplus F(x) = F(x \oplus m \oplus x) = F(m)$, irrespective of x. This property is important, as a recent scholar work has shown empirically that on FPGAs, both Hamming distance and Hamming weight leakage models should be envisioned [MM12b].

11.2 Second-Order Leakage Squeezing

11.2.1 Goal

In this section, an improvement of the leakage squeezing where two masks are used is studied. More precisely,

- the masked data ($X \oplus M_1 \oplus M_2$, also noted $X \oplus M$, where $M \doteq M_1 \oplus M_2$) is processed as is, *i.e.* through a bijection that is the identity (denoted by Id),
- the first mask (M_1) is processed through bijection F_1 and
- the second mask (M_2) is processed through bijection F_2.

This second-order masking scheme is illustrated in Fig. 11.2. With respect to the first-order masking scheme (see Fig. 11.1), the processing of the masked sensitive data is unchanged, and only the masks processing differs: each mask can be seeded independently and evolves from a different diversification function (noted R_1 and R_2). The leakage function is thus:

$$L = \mathsf{HW}((X \oplus M_1 \oplus M_2) \oplus (X' \oplus M_1' \oplus M_2')) + \mathsf{HW}(F_1(M_1) \oplus F_1(M_1')) + \mathsf{HW}(F_2(M_2) \oplus F_2(M_2')) \ .$$

We denote by $M_1'' \doteq M_1 \oplus M_1'$ and $M_2'' \doteq M_2 \oplus M_2'$. Hence, the leakage can be expressed as:

$$\begin{aligned} L &= \mathsf{HW}(Z \oplus M_1'' \oplus M_2'') + \mathsf{HW}(F_1(M_1) \oplus F_1(M_1 \oplus M_1'')) + \mathsf{HW}(F_2(M_2) \oplus F_2(M_2 \oplus M_2'')) \\ &= \mathsf{HW}(Z \oplus M_1'' \oplus M_2'') + \mathsf{HW}(D_{M_1''} F_1(M_1)) + \mathsf{HW}(D_{M_2''} F_2(M_2)) \ . \end{aligned} \tag{11.1}$$

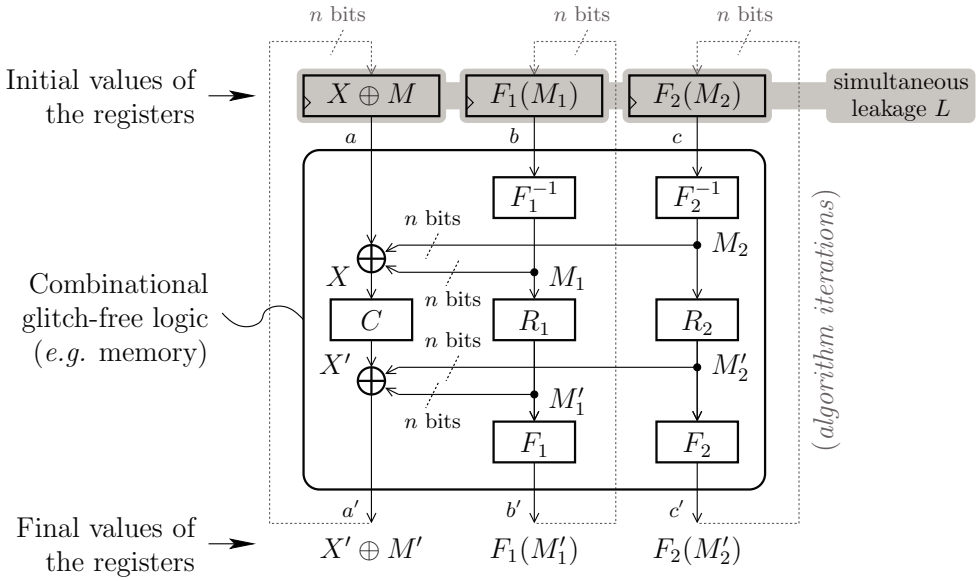


Figure 11.2: Setup of the second-order leakage squeezing masking countermeasure with bijections F_1 and F_2.

11.2.2 Motivation

It could be argued that the security brought by first-order leakage squeezing is already high enough, and resisting at still higher orders is a superfluous refinement. Admittedly, it has seldom been question of higher-order attacks of order strictly greater than two in the abundant public literature.

However, searching for greater security can be motivated by "forward security" concerns. Secure elements (*e.g.* smart cards, RFID chips, hardware security modules, *etc.*) contain high-value secrets, and cannot be upgraded. Therefore, one can imagine buying one of these today, and having it attacked with tomorrow's know-how. For instance, with the advance of science, measurements apparati will have a lower noise figure and a greater vertical resolution in the future, thereby reducing the noise in side channel acquisitions. Now, it is known that the limiting factor for the higher-order attacks is the noise variance [WW04]. Also, it is now well understood how to combine partially successful side channel attacks with brute force search [Dic11, VCGRS11]. Therefore, computer-assisted side channel attacks might greatly enhance what can be done today. Thus, to avoid tomorrow successful attacks of orders greater of one, two, or more orders than what is possible today, precautions must be envisioned today. A parallel can be made with the evolution of:

- the key size of block ciphers,
- the modulus size of asymmetric primitive, or
- the internal state of hash functions.

Those have continuously been increasing over the last years. Besides, the regulation in terms of security compliance standards is always one step ahead the state-of-the-art attacks. Consequently, it is not absurd that side channel resistance of very high order be demanded soon (*e.g.* with the forthcoming standard ISO/IEC 17 825), hence an incentive for the research in really higher-order countermeasures.

Finally, some products supporting second-order countermeasures are already deployed in the field. The second-order leakage squeezing can be mapped in the devices of this installed base at virtually no extra cost, and so the application of this method in real products does not require further architectural development costs. The sole modification is the entry of the masking material in the (F_1, F_2) bijections, and their leaving at the end of the cryptographic application.

11.3 Formalization of Second-Order Leakage Squeezing

The attack fails at order d if $\forall i \leq d, \mathbb{E}\left(L^i \mid Z = z\right)$ does not depend on z. Indeed, the attacker has thus no bias to relate the leakage at order $i \geq 1$ to the (predictable and key-dependent) sensitive variable Z. Now, the goal of the attacker is to exhibit a bias in $\mathbb{E}\left(L^d \mid Z = z\right)$ for an exponent d as small as possible, because the noise in L^d evolves as $\left(\sigma^2\right)^d$ [WW04], where σ^2 is the variance of the noise (for $d = 1$). Taking into account the formula of L from Eqn. (11.1), we have the following expression for $\mathbb{E}\left(L^i \mid Z = z\right)$:

$$
\begin{aligned}
& \mathbb{E}\left(\left(\mathsf{HW}(Z \oplus M_1'' \oplus M_2'') + \mathsf{HW}(D_{M_1''}F_1(M_1)) + \mathsf{HW}(D_{M_2''}F_2(M_2))\right)^i \mid Z = z\right) \\
= & \frac{1}{2^{4n}} \sum_{m_1'', m_2''} \sum_{m_1, m_2} \left(\mathsf{HW}(z \oplus m_1'' \oplus m_2'') + \mathsf{HW}(D_{m_1''}F_1(m_1)) + \mathsf{HW}(D_{m_2''}F_2(m_2))\right)^i \\
= & \frac{1}{2^{4n}} \sum_{\substack{m_1'', m_2'' \\ m_1, m_2}} \left(\underbrace{\mathsf{HW}(z \oplus m_1'' \oplus m_2'')}_{\text{Term \#0}} + \underbrace{\mathsf{HW}(D_{m_1''}F_1(m_1))}_{\text{Term \#1}} + \underbrace{\mathsf{HW}(D_{m_2''}F_2(m_2))}_{\text{Term \#2}}\right)^i .
\end{aligned}
$$

This equation can be developed, to yield a sum of products of the three terms. Let us denote by p, q and r the degrees of each term, that satisfy $p + q + r = i$. So attacks fail at order d if for all p, q and r such as $p + q + r \leq d$, the function

$$
\begin{aligned}
& z \mapsto f(z) \\
\doteq & \sum_{m_1'', m_2''} \sum_{m_1, m_2} \mathsf{HW}^p(z \oplus m_1'' \oplus m_2'') \cdot \mathsf{HW}^q(D_{m_1''}F_1(m_1)) \cdot \mathsf{HW}^r(D_{m_2''}F_2(m_2)) \\
= & \sum_{m_1'', m_2''} \mathsf{HW}^p\left(z \oplus m_1'' \oplus m_2''\right) \cdot \sum_{m_1} \mathsf{HW}^q\left(D_{m_1''}F_1(m_1)\right) \cdot \sum_{m_2} \mathsf{HW}^r\left(D_{m_2''}F_2(m_2)\right) \\
= & \sum_{m_1'', m_2''} \mathsf{HW}^p\left(z \oplus m_1'' \oplus m_2''\right) \cdot \mathbb{E}[\mathsf{HW}^q\left(D_{m_1''}F_1(M_1)\right)] \cdot \mathbb{E}[\mathsf{HW}^r\left(D_{m_2''}F_2(M_2)\right)] \\
= & \left\{\mathsf{HW}^p \otimes \mathbb{E}[\mathsf{HW}^q \circ D_{(\cdot)}F_1(M_1)] \otimes \mathbb{E}[\mathsf{HW}^r \circ D_{(\cdot)}F_2(M_2)]\right\}(z) \qquad (11.2)
\end{aligned}
$$

is constant. From Eqn. (11.2), we see that every term to be kept constant is a double convolution product.

Keeping f constant is equivalent to having the Fourier transform $\hat{f}$ of f null everywhere but in zero. The Fourier transform turns a convolution product into a product; therefore,

$$\hat{f} \quad = \quad \widehat{\mathsf{HW}^p} \cdot \mathbb{E}[\widehat{\mathsf{HW}^q \circ D_{(.)}F_1}(M_1)] \cdot \mathbb{E}[\widehat{\mathsf{HW}^r \circ D_{(.)}F_2}(M_2)] \ .$$

In summary, to resist at order d, we are attempting to find two bijections F_1 and F_2 such as:

$$\begin{aligned} \forall a \in \mathbb{F}_2^{n*}, \quad \widehat{\mathsf{HW}^p}(a) = 0 \quad &\text{or} \quad \mathbb{E}[\widehat{\mathsf{HW}^q \circ D_{(.)}F_1}(M)](a) = 0 \\ &\text{or} \quad \mathbb{E}[\widehat{\mathsf{HW}^r \circ D_{(.)}F_2}(M)](a) = 0 \ , \end{aligned} \tag{11.3}$$

for every triple of integers p, q and r such as $p + q + r \leq d$, d being the targeted protection order.

The Fourier support of a function $\psi : \mathbb{F}_2^n \to \mathbb{Z}$ is the set $\left\{a \in \mathbb{F}_2^n; \hat{\psi}(a) \neq 0\right\}$. The equation (11.3) expresses the fact that the Fourier supports of HW^p, $\mathbb{E}[\mathsf{HW}^q \circ D_{(.)}F_1(M)]$ and $\mathbb{E}[\mathsf{HW}^r \circ D_{(.)}F_2(M)]$ intersect only in the singleton $\{0\}$.

11.3.1 Gaining at Least one Order with Two Masks instead of One

It is a well known property that adding one mask increases the security by one order [WW04]. We here prove that the same benefit can be expected from the leakage squeezing.

Proposition 2. *Let F_1 be a bijection such that the security is reached at order d with one mask. Then, by introducing a second mask processed through whatever bijection F_2, the security is reached at order $d + 1$ at least.*

Proof. Let (p, q, r) be any triple of integers such as $p + q + r \leq d + 1$. Then:

- if $r = 0$, $\widehat{\mathsf{HW}^r \circ D_{(.)}F_2} = \widehat{1 \circ D_{(.)}F_2} = \widehat{1} = \delta$ is a Kronecker symbol function, hence null for all $a \neq 0$,
- otherwise, $r > 0$ and for all p, q, we have $p + q \leq d + 1 - r$ (by hypothesis), and so $p + q \leq d$. Thus, we have $\widehat{\mathsf{HW}^p}(a) \cdot \widehat{\mathsf{HW}^q \circ F_1}(a) = 0$, which implies that either $\widehat{\mathsf{HW}^p}(a) = 0$ or $\widehat{\mathsf{HW}^q \circ F_1}(a) = 0$ for $a \neq 0$.

□

11.3.2 Problem Formulation in Terms of Boolean Theory

In the following, we shall need the next lemma.

Lemma 5. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be any function, let q be an integer such that $0 < q < n$ and let $a \in \mathbb{F}_2^n$ be nonzero. We have $\sum_{z,m} \mathsf{HW}^{q'}(F(m) \oplus F(m \oplus z))(-1)^{a \cdot z} = 0$ for every $0 < q' \leq q$ if and only if $\widehat{b \cdot F}(a) = 0$ for every $b \in \mathbb{F}_2^n$ such that $\mathsf{HW}(b) \leq q$.*

Proof. According to Eqn. (10.9) in Chapter 10, we have:

$$\sum_{z,m} \mathsf{HW}^{q'}(F(m) \oplus F(m \oplus z))(-1)^{a \cdot z} = \tag{11.4}$$

$$\frac{1}{2^{q'}} \sum_{j=0}^{q'} \binom{q'}{j} n^{q'-j} (-1)^j \sum_{k_1+\cdots+k_n=j} \binom{j}{k_1, \cdots, k_n} \left(\sum_{x \in \mathbb{F}_2^n} (-1)^{\left(\oplus_{i=1}^n k_i e_i\right) \cdot F(x) + a \cdot x} \right)^2 .$$

Since, for $b = \oplus_{i=1}^n k_i e_i$, we have $\sum_{x \in \mathbb{F}_2^n} (-1)^{\left(\oplus_{i=1}^n k_i e_i\right) \cdot F(x) + a \cdot x} = -2\widehat{b \cdot F}(a)$, the condition "$\widehat{b \cdot F}(a) = 0$ for every $b \in \mathbb{F}_2^n$ such that $\mathsf{HW}(b) \leq q$" is then clearly sufficient. Conversely, let the condition "$\widehat{b \cdot F}(a) = 0$ for every $b \in \mathbb{F}_2^n$ such that $\mathsf{HW}(b) \leq k$" be denoted by $P(k)$. We prove $P(k)$ by induction on $k \in \mathbb{N}$. $P(0)$ is clearly satisfied since $a \neq 0$. Assume that $P(k)$ is satisfied for some $0 \leq k \leq q-1$, then applying the hypothesis with $q' = k+1$ implies that $\widehat{b \cdot F}(a) = 0$ for every b such that $\mathsf{HW}(b) = k+1$ since we have only squares in Eqn. (11.4) multiplied by coefficients which are all of the same sign and $P(k+1)$ is then satisfied. This completes the proof by induction.

□

Incidentally, we remark that the Theorem 7 of previous Sec. 11.1.1 is also an immediate consequence of Lemma 5 with $F = \mathsf{Id}$. We characterize now Eqn. (11.3) in terms of Fourier transform.

Proposition 3. *Let F_1 and F_2 be two permutations of $\mathbb{F}_2^n$ and d an integer smaller than n. The condition:*

$$\forall a \neq 0, \forall (p,q,r), \tag{11.5}$$

$$(p+q+r \leq d) \Longrightarrow \begin{cases} \widehat{\mathsf{HW}^p}(a) = 0 \ \textit{or} \\ \sum_{z,m} \mathsf{HW}^q(F_1(m) \oplus F_1(m \oplus z))(-1)^{a \cdot z} = 0 \ \textit{or} \\ \sum_{z,m} \mathsf{HW}^r(F_2(m) \oplus F_2(m \oplus z))(-1)^{a \cdot z} = 0 \ . \end{cases}$$

is satisfied if and only if:

$$\forall a \in \mathbb{F}_2^n, a \neq 0, \exists q, r / \begin{cases} \mathsf{HW}(a) + q + r = d - 1, \\ \forall b \in \mathbb{F}_2^n, \mathsf{HW}(b) \leq q \Longrightarrow \widehat{b \cdot F_1}(a) = 0, \\ \forall c \in \mathbb{F}_2^n, \mathsf{HW}(c) \leq r \Longrightarrow \widehat{c \cdot F_2}(a) = 0. \end{cases} \tag{11.6}$$

Proof. Condition (11.5) is satisfied for every (p,q,r) such that $p+q+r \leq d$ if and only if it is satisfied when p is minimal such that $\widehat{\mathsf{HW}^p}(a) \neq 0$, r is minimal such that $\sum_{z,m} \mathsf{HW}^r(F_2(m) \oplus F_2(m \oplus z))(-1)^{a \cdot z} \neq 0$ and $p+q+r \leq d$. We know that the minimum value of p such that $\widehat{\mathsf{HW}^p}(a) \neq 0$ equals $\mathsf{HW}(a)$. Let r be the minimal element defined above. Condition (11.5) implies then:

$$\forall q \leq d - \mathsf{HW}(a) - r, \quad \sum_{z,m} \mathsf{HW}^q(F_1(m) \oplus F_1(m \oplus z))(-1)^{a \cdot z} = 0.$$

According to Lemma 5, this latter condition is equivalent to $\forall b, \mathsf{HW}(b) \leq d - \mathsf{HW}(a) - r \Longrightarrow \widehat{b \cdot F_1}(a) = 0$ and we obtain the condition:

$$\forall a \neq 0, \exists r / \begin{cases} \forall b, \mathsf{HW}(b) \leq d - \mathsf{HW}(a) - r & \Longrightarrow \widehat{b \cdot F_1}(a) = 0, \\ \forall c, \mathsf{HW}(c) < r & \Longrightarrow \widehat{c \cdot F_2}(a) = 0. \end{cases}$$

Now, let us replace r by $r' \doteq r - 1$. Thus $\mathsf{HW}(c) < r$ is equivalent to $\mathsf{HW}(c) \leq r'$, and condition $\mathsf{HW}(a) + q + r = d$ is equivalent to $\mathsf{HW}(a) + q + r' = d - 1$. This shows that Eqn. (11.6) is necessary. Clearly, it is also sufficient.

□

It is clear from Proposition 3 that any choice of F_2 allows to increase by one the resistance order provided by F_1 (this has already been mentioned in Sec. 11.3.1). Indeed, let us denote by d_1 the maximal order of resistance of F_1 in the one mask situation. Then, $\forall a \neq 0$, $\forall p, q$, $p + q \leq d_1 \Longrightarrow \widehat{b \cdot F_1}(a) = 0$. By reference to Eqn. (11.6), for a given $a \neq 0$, we choose:

- $q = d_1 - \mathsf{HW}(a)$, thus $\forall b \in \mathbb{F}_2^n, \mathsf{HW}(b) \leq q \Longrightarrow \widehat{b \cdot F_1}(a) = 0$ (by definition of d_1).
- $r = 0$, thus $\forall c \in \mathbb{F}_2^n, \mathsf{HW}(c) \leq r \Longrightarrow \widehat{c \cdot F_2}(a) = 0$ (indeed, $c = 0$, hence $\widehat{c \cdot F_2}(a) = \delta(a) = 0$ since $a \neq 0$).

Consequently, Eqn. (11.6) is met with $d = d_1 + 1$.

So, one strategy can be to start from F_1, the optimal solution with one mask (this solution is known from the previous chapter), and then to choose F_2 so as to increase as much as possible the resistance degree. Another strategy is to find F_1 and F_2 jointly. This problem seems not to be a classical one in the general case. In the next section, we show however that it becomes a problem of coding theory when F_1 and F_2 are linear.

11.4 Solutions when F_1 and F_2 are Linear

In this section, F_1 and F_2 are assumed to be linear. For every $b, x \in \mathbb{F}_2^n$, we have $b \cdot F_1(x) = F_1^t(b) \cdot x$, where F_1^t is the so-called *adjoint operator* of F_1, that is, the linear mapping whose matrix is the transpose of the matrix of F_1. Then, for every nonzero $a \in \mathbb{F}_2^n$, we have $\widehat{b \cdot F_1}(a) = -\frac{1}{2} \sum_{x \in \mathbb{F}_2^n} (-1)^{(F_1^t(b) \oplus a) \cdot x}$, which equals $-2^{n-1} \neq 0$ if $F_1^t(b) = a$ and is null otherwise. Let us denote by L_1 (resp. L_2) the inverse of mapping F_1^t (resp. F_2^t). Then, $\widehat{b \cdot F_1}(a)$ (resp. $\widehat{c \cdot F_2}(a)$) equals $-2^{n-1} \neq 0$ if $b = L_1(a)$ (resp. if $c = L_2(a)$) and is null otherwise.

Let also $a \neq 0$. From Eqn. (11.6) of Proposition 3, we can choose:

- $q = \mathsf{HW}(L_1(a)) - 1$ and
- $r = \mathsf{HW}(L_2(a)) - 1$.

Thus $d = \min \{\mathsf{HW}(a) + \mathsf{HW}(L_1(a)) + \mathsf{HW}(L_2(a)) - 1; a \neq 0\}$, which is exactly the minimal distance of the code $\left\{(x, L_1^t(x), L_2^t(x)); x \in \mathbb{F}_2^n\right\}$ (of rate 1/3 and with three disjoint information sets) minus the number 1.

11.4.1 Example for Linear F_1 and F_2 for $n = 8$

If F_1 and F_2 are two linear bijections then the linear code $\{(x, F_1(x), F_2(x)); x \in \mathbb{F}_2^n\}$ has $\{1, \cdots, n\}$, $\{n+1, \cdots, 2n\}$ and $\{2n+1, \cdots, 3n\}$ for information sets, since the restriction of the generator matrix of this code to the columns indexed in each of these three sets is invertible. Conversely, if a $[3n, n, d]$ code C is known with three disjoint information sets, then after rearranging the columns of its generator matrix so that these three information sets are $\{1, \cdots, n\}$, $\{n+1, \cdots, 2n\}$ and $\{2n+1, \cdots, 3n\}$, we have $C = \{(\varphi_0(x), \varphi_1(x), \varphi_2(x)); x \in \mathbb{F}_2^n\}$ where φ_0, φ_1 and φ_2 are bijective. Then, by trading the dummy variable x for $y = \varphi_0(x)$ through one-to-one function φ_0, we get $C = \{(y, \varphi_1 \circ \varphi_0^{-1}(y), \varphi_2 \circ \varphi_0^{-1}(y)); y \in \mathbb{F}_2^n\}$ and we can take $F_1 = \varphi_1 \circ \varphi_0^{-1}$ and $F_2 = \varphi_2 \circ \varphi_0^{-1}$.

One generator matrix for the $[24, 8, 8]$ code can be obtained as a submatrix of extended quadratic-residue (QR) code of length 23[1], such as:

$$\begin{array}{cccccccccccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\left(\begin{array}{cccccccccccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}\right)
\end{array}.$$

The goal is to rearrange the columns of this matrix to get a form:

$$\begin{pmatrix} M_0^t & M_1^t & M_2^t \end{pmatrix}, \tag{11.7}$$

where M_0, M_1 and M_2 are 8×8 invertible matrices with elements in $\mathbb{F}_2$. The research algorithm is as follows: first, an invertible 8×8 matrix (M_0) is searched. There are $\binom{24}{8} = 735,471$ of them[2]. We find one M_0^t by considering the columns $[\![2, 9]\!]$. Second, the $\binom{16}{8} = 12,870$ permutations of columns $\{1\} \cup [\![10, 24]\!]$ are tested for a partitioning into two invertible matrices $\begin{pmatrix} M_1^t & M_2^t \end{pmatrix}$. For instance, M_1^t can be the columns $\{1, 10, 11, 12, 13, 15, 17, 18\}$ and M_2^t the columns $\{14, 16\} \cup [\![19, 24]\!]$. Those define the three bijections $\varphi_i : \mathbb{F}_2^8 \to \mathbb{F}_2^8, x \mapsto M_i \times x$, for $i \in \{0, 1, 2\}$. After that, we get a generating matrix in systematic form $\begin{pmatrix} I_8 & L_1^t & L_2^t \end{pmatrix}$; The matrices L_1^t and L_2^t are defined as $L_1^t = M_1 \times M_0^{-1} = \left((M_0^t)^{-1} \times M_1^t\right)^t$ and $L_2^t = M_2 \times M_0^{-1} =$

[1] See: http://www.mathe2.uni-bayreuth.de/cgi-bin/axel/codedb?extensioncodeid+39649+2+8 [Gra07].

[2] This amount of tries is still manageable on a standard desktop personal computer; all the more so as, in practice, we find very quickly a solution as the number of partitionings that yield an invertible 8×8 matrix is $310,400$ (which represents around 42% of the possible partitionings).

$\left((M_0^t)^{-1} \times M_2^t\right)^t$, and their values are given in the following equation:

$$L_1^t = \begin{pmatrix} 1&0&0&0&0&1&0&1\\ 1&0&0&0&0&1&1&1\\ 1&0&1&1&1&0&0&1\\ 1&0&1&1&1&0&1&0\\ 1&0&1&1&1&1&1&0\\ 0&1&1&0&0&1&1&1\\ 0&1&0&1&0&1&1&1\\ 0&1&0&0&1&0&0&0 \end{pmatrix}, \quad L_2^t = \begin{pmatrix} 0&1&1&0&0&1&1&0\\ 0&0&0&1&1&0&1&0\\ 1&1&1&1&0&1&1&1\\ 1&0&1&0&0&0&0&1\\ 1&1&0&1&1&0&1&0\\ 1&1&1&1&0&0&0&1\\ 1&1&1&1&1&1&1&0\\ 1&0&0&0&1&1&1&1 \end{pmatrix}. \quad (11.8)$$

Those matrices are of full rank, namely 8, and their inverses are:

$$(L_1^t)^{-1} = \begin{pmatrix} 0&1&1&0&1&0&0&0\\ 1&0&1&1&1&1&1&1\\ 1&0&0&1&0&0&1&1\\ 1&0&0&1&0&1&0&1\\ 1&0&1&1&1&1&1&0\\ 0&0&0&1&1&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0 \end{pmatrix}, \quad (L_2^t)^{-1} = \begin{pmatrix} 0&0&0&0&0&1&1&1\\ 0&1&0&0&1&1&1&1\\ 1&1&1&0&1&0&1&1\\ 0&1&0&1&1&0&1&1\\ 1&1&0&1&1&1&1&0\\ 1&1&1&0&0&0&0&1\\ 1&1&0&0&0&1&0&1\\ 1&1&1&1&1&1&0&0 \end{pmatrix}.$$

The resulting linear functions F_1 and F_2 of $\mathbb{F}_2^8$, that provide resistance up to level 7, are tabulated in Tab. 11.1.

We note that the binary linear code $\{(x, F_1(x)); x \in \mathbb{F}_2^8\}$ has minimal distance 3, and that the binary linear code $\{(x, F_2(x)); x \in \mathbb{F}_2^8\}$ has minimal distance 4. So, those two codes are non-optimal, because the best linear code of length 16 and dimension 8 has minimal distance 5 (see Tab. 10.2 in Chapter 10). This noting justifies that it is indeed relevant to search for the bijections doublet (F_1, F_2) together instead of one after the other, independently. It also suggests that non-linear codes might still achieve better.

11.4.2 Example for Linear F_1 and F_2 for $n = 4$

It is also possible to construct a rate 1/3 linear code of dimension 4 with three distinct information sets, which is suitable to protect DES. The same research algorithm can be used for the optimal code of length 12 and dimension 4, *i.e.* $[12, 4, 6]$. Its minimal distance is 6. One generator matrix for the $[12, 4, 6]$ code is[3]:

$$\begin{matrix} 1&2&3&4&5&6&7&8&9&10&11&12\\ \downarrow&\downarrow&\downarrow&\downarrow&\downarrow&\downarrow&\downarrow&\downarrow&\downarrow&\downarrow&\downarrow&\downarrow \end{matrix}$$
$$\begin{pmatrix} 1&0&0&1&1&1&1&0&0&1&1&1\\ 0&1&0&1&1&0&0&1&0&1&1&0\\ 0&0&1&1&0&1&0&0&1&1&0&1\\ 0&0&0&0&0&0&1&1&1&1&1&1 \end{pmatrix}.$$

By gathering columns $\{1, 4, 7, 2\}$ as M_0^t, columns $\{10, 5, 8, 3\}$ as M_1^t and columns $\{11, 6, 9, 12\}$ as M_2^t, the code rewrites in the form of Eqn. (11.7) where M_0, M_1 and

[3] See: http://www.math.colostate.edu/~betten/research/codes/BOUNDS/bounds_GF2.html.

Table 11.1: Truth table for the found linear bijections F_1 and F_2 of $\mathbb{F}_2^8$.

$\{F_1(x); x \in \mathbb{F}_2^8\} =$

```
{ 0x00, 0x70, 0x68, 0x18, 0x58, 0x28, 0x30, 0x40,
  0xcc, 0xbc, 0xa4, 0xd4, 0x94, 0xe4, 0xfc, 0x8c,
  0x7d, 0x0d, 0x15, 0x65, 0x25, 0x55, 0x4d, 0x3d,
  0xb1, 0xc1, 0xd9, 0xa9, 0xe9, 0x99, 0x81, 0xf1,
  0xc9, 0xb9, 0xa1, 0xd1, 0x91, 0xe1, 0xf9, 0x89,
  0x05, 0x75, 0x6d, 0x1d, 0x5d, 0x2d, 0x35, 0x45,
  0xb4, 0xc4, 0xdc, 0xac, 0xec, 0x9c, 0x84, 0xf4,
  0x78, 0x08, 0x10, 0x60, 0x20, 0x50, 0x48, 0x38,
  0x83, 0xf3, 0xeb, 0x9b, 0xdb, 0xab, 0xb3, 0xc3,
  0x4f, 0x3f, 0x27, 0x57, 0x17, 0x67, 0x7f, 0x0f,
  0xfe, 0x8e, 0x96, 0xe6, 0xa6, 0xd6, 0xce, 0xbe,
  0x32, 0x42, 0x5a, 0x2a, 0x6a, 0x1a, 0x02, 0x72,
  0x4a, 0x3a, 0x22, 0x52, 0x12, 0x62, 0x7a, 0x0a,
  0x86, 0xf6, 0xee, 0x9e, 0xde, 0xae, 0xb6, 0xc6,
  0x37, 0x47, 0x5f, 0x2f, 0x6f, 0x1f, 0x07, 0x77,
  0xfb, 0x8b, 0x93, 0xe3, 0xa3, 0xd3, 0xcb, 0xbb,
  0x7b, 0x0b, 0x13, 0x63, 0x23, 0x53, 0x4b, 0x3b,
  0xb7, 0xc7, 0xdf, 0xaf, 0xef, 0x9f, 0x87, 0xf7,
  0x06, 0x76, 0x6e, 0x1e, 0x5e, 0x2e, 0x36, 0x46,
  0xca, 0xba, 0xa2, 0xd2, 0x92, 0xe2, 0xfa, 0x8a,
  0xb2, 0xc2, 0xda, 0xaa, 0xea, 0x9a, 0x82, 0xf2,
  0x7e, 0x0e, 0x16, 0x66, 0x26, 0x56, 0x4e, 0x3e,
  0xcf, 0xbf, 0xa7, 0xd7, 0x97, 0xe7, 0xff, 0x8f,
  0x03, 0x73, 0x6b, 0x1b, 0x5b, 0x2b, 0x33, 0x43,
  0xf8, 0x88, 0x90, 0xe0, 0xa0, 0xd0, 0xc8, 0xb8,
  0x34, 0x44, 0x5c, 0x2c, 0x6c, 0x1c, 0x04, 0x74,
  0x85, 0xf5, 0xed, 0x9d, 0xdd, 0xad, 0xb5, 0xc5,
  0x49, 0x39, 0x21, 0x51, 0x11, 0x61, 0x79, 0x09,
  0x31, 0x41, 0x59, 0x29, 0x69, 0x19, 0x01, 0x71,
  0xfd, 0x8d, 0x95, 0xe5, 0xa5, 0xd5, 0xcd, 0xbd,
  0x4c, 0x3c, 0x24, 0x54, 0x14, 0x64, 0x7c, 0x0c,
  0x80, 0xf0, 0xe8, 0x98, 0xd8, 0xa8, 0xb0, 0xc0 },
```

$\{F_2(x); x \in \mathbb{F}_2^8\} =$

```
{ 0x00, 0xf6, 0xf8, 0x0e, 0xcb, 0x3d, 0x33, 0xc5,
  0x79, 0x8f, 0x81, 0x77, 0xb2, 0x44, 0x4a, 0xbc,
  0x19, 0xef, 0xe1, 0x17, 0xd2, 0x24, 0x2a, 0xdc,
  0x60, 0x96, 0x98, 0x6e, 0xab, 0x5d, 0x53, 0xa5,
  0x25, 0xd3, 0xdd, 0x2b, 0xee, 0x18, 0x16, 0xe0,
  0x5c, 0xaa, 0xa4, 0x52, 0x97, 0x61, 0x6f, 0x99,
  0x3c, 0xca, 0xc4, 0x32, 0xf7, 0x01, 0x0f, 0xf9,
  0x45, 0xb3, 0xbd, 0x4b, 0x8e, 0x78, 0x76, 0x80,
  0x7f, 0x89, 0x87, 0x71, 0xb4, 0x42, 0x4c, 0xba,
  0x06, 0xf0, 0xfe, 0x08, 0xcd, 0x3b, 0x35, 0xc3,
  0x66, 0x90, 0x9e, 0x68, 0xad, 0x5b, 0x55, 0xa3,
  0x1f, 0xe9, 0xe7, 0x11, 0xd4, 0x22, 0x2c, 0xda,
  0x5a, 0xac, 0xa2, 0x54, 0x91, 0x67, 0x69, 0x9f,
  0x23, 0xd5, 0xdb, 0x2d, 0xe8, 0x1e, 0x10, 0xe6,
  0x43, 0xb5, 0xbb, 0x4d, 0x88, 0x7e, 0x70, 0x86,
  0x3a, 0xcc, 0xc2, 0x34, 0xf1, 0x07, 0x09, 0xff,
  0x2f, 0xd9, 0xd7, 0x21, 0xe4, 0x12, 0x1c, 0xea,
  0x56, 0xa0, 0xae, 0x58, 0x9d, 0x6b, 0x65, 0x93,
  0x36, 0xc0, 0xce, 0x38, 0xfd, 0x0b, 0x05, 0xf3,
  0x4f, 0xb9, 0xb7, 0x41, 0x84, 0x72, 0x7c, 0x8a,
  0x0a, 0xfc, 0xf2, 0x04, 0xc1, 0x37, 0x39, 0xcf,
  0x73, 0x85, 0x8b, 0x7d, 0xb8, 0x4e, 0x40, 0xb6,
  0x13, 0xe5, 0xeb, 0x1d, 0xd8, 0x2e, 0x20, 0xd6,
  0x6a, 0x9c, 0x92, 0x64, 0xa1, 0x57, 0x59, 0xaf,
  0x50, 0xa6, 0xa8, 0x5e, 0x9b, 0x6d, 0x63, 0x95,
  0x29, 0xdf, 0xd1, 0x27, 0xe2, 0x14, 0x1a, 0xec,
  0x49, 0xbf, 0xb1, 0x47, 0x82, 0x74, 0x7a, 0x8c,
  0x30, 0xc6, 0xc8, 0x3e, 0xfb, 0x0d, 0x03, 0xf5,
  0x75, 0x83, 0x8d, 0x7b, 0xbe, 0x48, 0x46, 0xb0,
  0x0c, 0xfa, 0xf4, 0x02, 0xc7, 0x31, 0x3f, 0xc9,
  0x6c, 0x9a, 0x94, 0x62, 0xa7, 0x51, 0x5f, 0xa9,
  0x15, 0xe3, 0xed, 0x1b, 0xde, 0x28, 0x26, 0xd0 }.
```

M_2 are invertible. Specifically, we have:

$$M_0^t = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad M_1^t = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad M_2^t = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix};$$

$$(M_0^t)^{-1} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad (M_1^t)^{-1} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad (M_2^t)^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$

So, by seeing x as a column $x \doteq \begin{pmatrix} x_{n-1} \\ \vdots \\ x_0 \end{pmatrix}$, we also define:

$$F_1(x) \doteq \left[(M_1^t)^{-1} \times M_0^t\right] \times x = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \times x \ ,$$

$$F_2(x) \doteq \left[(M_2^t)^{-1} \times M_0^t\right] \times x = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \times x \ . \tag{11.9}$$

The resulting linear functions F_1 and F_2 of $\mathbb{F}_2^4$, that provide resistance up to level 5, are tabulated as follows:

$$\left\{F_1(x); x \in \mathbb{F}_2^4\right\} = \begin{Bmatrix} \texttt{0x0}, & \texttt{0xf}, & \texttt{0x6}, & \texttt{0x9}, & \texttt{0x5}, & \texttt{0xa}, & \texttt{0x3}, & \texttt{0xc}, \\ \texttt{0xb}, & \texttt{0x4}, & \texttt{0xd}, & \texttt{0x2}, & \texttt{0xe}, & \texttt{0x1}, & \texttt{0x8}, & \texttt{0x7} \end{Bmatrix},$$

$$\left\{F_2(x); x \in \mathbb{F}_2^4\right\} = \begin{Bmatrix} \texttt{0x0}, & \texttt{0xe}, & \texttt{0x6}, & \texttt{0x8}, & \texttt{0xa}, & \texttt{0x4}, & \texttt{0xc}, & \texttt{0x2}, \\ \texttt{0x3}, & \texttt{0xd}, & \texttt{0x5}, & \texttt{0xb}, & \texttt{0x9}, & \texttt{0x7}, & \texttt{0xf}, & \texttt{0x1} \end{Bmatrix}.$$

11.5 Security Evaluation

An implementation with a leakage function L is vulnerable at order d if $\mathbb{E}[(L - \mathbb{E}[L])^d \mid Z = z]$ depends on z. In this case, the asymptotic HO-CPA correlation coefficient $\rho_{\text{opt}}^{(d)}$, equal to [PRB09, Eqn. (15)]:

$$\rho_{\text{opt}}^{(d)} \doteq \sqrt{\frac{\mathsf{Var}[\mathbb{E}[(L - \mathbb{E}[L])^d \mid Z]]}{\mathsf{Var}[(L - \mathbb{E}[L])^d]}} \ , \tag{11.10}$$

is non-zero. In the middle of Tab. 11.2, an intermediate case is shown: it corresponds to a "partial" leakage squeezing, where a bijection is applied only on one mask out of the two. We notice that the simulation results of partial leakage squeezing are in line with the theoretical analysis carried out in Sec. 11.3.1: the order of resistance is indeed incremented by one. The two variances involved in Eqn. (11.10) were computed using a multiprecision integer library; therefore, when $\rho_{\text{opt}}^{(d)}$ is reported as

0 (integer zero, not the approximated floating number 0.000000), we really mean that $\mathbb{E}[(L - \mathbb{E}[L])^d \mid Z = z]$ does not depend on z.

For the sake of comparison, we also report in this table the results obtained with one mask. In such case, both the best linear and non-linear squeezing bijection F can be characterized. It is relevant to consider the linear bijections F as they allow an efficient protection against HO-CPA, whether the device leaks in Hamming weight or distance. The best linear F for leakage squeezing with one mask is secure against attacks of orders up to 4. It can be used with two masks, thereby granting a security up to order $4 + 1 = 5$. Our results, that are not based on the extension of a single mask solution, provide security against HO-CPA of orders up to 7. Therefore, our method provides a free advantage of two orders. Now, with one mask, the best achievable security is gotten by the use of a non-linear F. This function does only protect against attacks that exploit the Hamming distance (and not the Hamming weight), but allows to reach a resistance up to HO-CPA of order 5. Here also, our linear solution with two masks is better than merely this code used with one mask extended with another mask: it protected at order up to $7 > 5 + 1$. Besides, it is interesting to compare the first nonzero correlation coefficients with and without leakage squeezing:

- with one mask, $\rho_{\text{opt}}^{(d=2)}(\text{no LS})/\rho_{\text{opt}}^{(d=5)}(\text{LS}) = 0.258199/0.023258 \approx 11$, and
- with two masks, $\rho_{\text{opt}}^{(d=3)}(\text{no LS})/\rho_{\text{opt}}^{(d=8)}(\text{LS}) = 0.038886/0.000446 \approx 87$.

So, in front of leakage squeezing, not only the attacker shall conduct an attack of much higher order, but also he will get a very degraded distinguisher value.

On $n = 4$ bits, the optimal first-order leakage squeezing is linear and allows to reach resistance order of 3. The used optimal code is $[8, 4, 4]$. For the second-order leakage squeezing, we can resort to the linear code $[12, 4, 6]$, that improves by two ($6 - 4 = 2$) orders the resistance against HO-CPA. By the trivial construct of Sec. 11.3.1, only one additional order of resistance would have been gained. A summary of the results is shown in Tab. 11.3. The improvement from the "straightforward" to the "squeezed" masking is of two orders with one mask and three orders with two masks.

11.6 Conclusions

In this chapter, we investigated the potential of leakage squeezing extension to second-order leakage squeezing. Our analysis allows to characterize (in Proposition 3) the conditions to reach higher resistance. The optimal solutions are not as easy to find as in the case with one mask. Nonetheless, for the special case of linear bijections, we found that one solution (probably not optimal) consists in finding a rate 1/3 linear code of maximal minimal distance with three disjoint information sets. The optimal $[24, 8, 8]$ linear code fulfills this condition, and makes it possible to resist attacks of all orders from 1 to 7 included. Concretely speaking, this result means that the same security level as a 7^{th}-order attack is attainable with 2 instead of 7 masks, thus at a much lower implementation cost.

Table 11.2: Optimal 2O-CPA correlation coefficient for d^{th}-order attacks, without and with leakage squeezing (LS) on $n = 8$ bits. Results are rounded at the sixth decimal.

	One mask			**Two masks**			
Order d	**Without LS**	**With LS**		**Without LS**	**With "partial" LS**		**With LS**
	$F = \mathsf{Id}$	**Optimal linear** (Sec. 11.4.1)	**Optimal non-linear** (Sec. 10.3.3)	$F_1 = F_2 = \mathsf{Id}$	$F_1 = \mathsf{Id}$, **but** F_2 **as (Sec. 11.4.1)**	$F_1 = \mathsf{Id}$, **but** F_2 **as (Sec. 10.3.3)**	**(Non-optimal) linear** (*cf.* Sec. 11.4)
1	0	0	0	0	0	0	0
2	0.258199	0	0	0	0	0	0
3	0	0	0	0.038886	0	0	0
4	0.235341	0	0	0	0	0	0
5	0	0.023231	0	0.049669	0	0	0
6	0.197908	0.016173	0.023258	0.003403	0.001286	0	0
7	0	0.042217	0	0.045585	0.000868	0.000726	0
8	0.164595	0.032796	0.046721	0.006820	0.002644	0.000682	0.000446

Table 11.3: Optimal 2O-CPA correlation coefficient for d^{th}-order attacks, without and with leakage squeezing (LS) on $n = 4$ bits. Results are rounded at the sixth decimal.

	One mask		**Two masks**		
Order d	**Without LS**	**With LS**	**Without LS**	**With "partial" LS**	**With LS**
	$F = \mathsf{Id}$	**Optimal linear** ([MGCD11, Eqn. (13)])	$F_1 = F_2 = \mathsf{Id}$	$F_1 = \mathsf{Id}$, **and** $F_2 \neq \mathsf{Id}$ **is [MGCD11, Eqn. (13)]**	**(Non-optimal) linear** (*cf.* Sec. 11.4)
1	0	0	0	0	0
2	0.377964	0	0	0	0
3	0	0	0.081289	0	0
4	0.363815	0.191663	0	0	0
5	0	0	0.105175	0.021035	0
6	0.346246	0.283546	0.015973	0.015973	0.022590

Part V
Conclusions and Perspectives

Chapter 12

Conclusions and Perspectives

12.1 Summary

In this thesis, we have investigated new techniques of both higher-order SCA attacks and countermeasures based on Boolean masking.

The first part of this thesis was focused on the study of higher-order side channel attacks. We proposed three novel attacks. First, we introduced the VPA attack based on variance analysis of the leakage measurements. This attack is shown to be powerful enough to break an FPGA implementation of a masked DES and requires a reasonable number of traces (12K). We demonstrated that VPA attack can be expressed as a second-order univariate CPA by changing the leakage model. Therefore, both distinguishers have the same soundness if the leakage model is perfectly known. Second, we proposed the EPA attack based on entropy analysis which succeeded also in breaking a hardware masked DES implementation. We compared it with the VPA attack, and we observed that the VPA performs better than the EPA attack when the leakage model is known. Finally, we suggested another distinguisher, termed IIA. Conversely to MIA or KSA, it consists in comparing the conditional leakages between themselves, pairwise. We compared theoretically IIA with MIA, and studied its efficiency in simulation for some types of leakages in the presence of masking countermeasure. Indeed, our theoretical analysis gives some reasons to show that IIA is more efficient in discriminating key hypotheses than MIA. Attacks simulations confirm that the new IIA distinguisher compares favorably to MIA, even when masking is applied to ensure the protection.

The second part was devoted to a formal security evaluation of Boolean hardware masking schemes. First, we proposed a new SCA metric called HO-CPA immunity to evaluate the robustness of higher-order Boolean masking countermeasures. Second, we showed that the ways to catch most of the leakage and to exploit it are more relevant for software implementation than hardware, as hardware has more algorithmic noise and is limited to the arithmetic sum of the leakages as combining function. Therefore, for these two important reasons, we proved that masking is a countermeasure more efficient in hardware than in software.

The third part was dedicated to the description and the evaluation of some new masking countermeasures. The first proposed solution is the leakage-free countermeasure. We argued that the sensible information leaked is null under some realistic assumptions about the device architecture, namely the symmetry of the transitions. The solution was evaluated within an information-theoretic study, proving its security against univariate SCA attacks under any distance model. When the leakage

function deviates slightly from this assumption, our solution still achieves excellent results. The second countermeasure consists in diversifying the mask update thanks to the Gray code. Practical implementations showed that the performances decrease in terms of complexity and speed are very limited, which is particularly true for the proposed USM implementation. The third solution proposed is the leakage squeezing countermeasure. Its principle consists in modifying the mask value by using a bijective transformation before it is stored in the mask register. The goal is to reduce the dependency between the mask and the masked data. We showed how to find optimal linear (resp. non-linear) bijection, that protects first-order masking against HO-CPA of orders $d \leq 4$ (resp. $d \leq 5$). Moreover, an information-theoretic study reveals that the mutual information between the leakage and the sensitive variable is lower than the same metric computed on the straightforward second-order masking CM. Then, we extended the leakage squeezing countermeasure to the second-order context. Our analysis allows to characterize the conditions to reach higher resistance when applying the leakage squeezing to the second-order masking CM. We demonstrated that the optimal $[24, 8, 8]$ linear code fulfills these conditions, and makes it possible to resist HO-CPA of orders $d \leq 7$.

12.2 Perspectives

An immediate perspective to the proposed EPA attack would be to find the optimal theoretical weight set for this distinguisher in order to enhance its robustness. A second perspective is to compare EPA and VPA attacks with multivariate SCA attacks (*e.g.* MMIA [GBPV10]), using multiple sensors (*e.g.* two magnetic probes) placed at different (X, Y, Z, ϑ) locations over a masked cryptoprocessor (*i.e. cartography*). Also, the IIA distinguisher should be deeply analysed, in particular when considering real measurements.

Another future work is to compare our countermeasures based fundamentally on Boolean masking with others solutions such as the multiplicative or the affine masking [FMPR10] schemes which also provide good performance-security trade-off against higher-order SCA attacks.

Another perspective is to find ways for the leakage-free countermeasure (with "mask update" primitives) to get protection also against multivariate SCA attacks. A perspective related to the leakage squeezing is to find better bijections, for instance non-linear, to further improve the obtained results. In particular, a thorough study of rate 1/2 codes with two complementary information sets exists [CGKS12]. However, such work is missing in general for rate $1/d$ codes with $d > 2$ distinct information sets.

Finally, a future work is to integrate the second-order leakage squeezing with "hyperpipelined" designs [MM12a], "threshold implementations" [NRS08] or "multi-party computation" [PR11] masking schemes, so as to improve their order of resistance while at the same time removing the latent leakage by glitches (if the logic is not concealed in memories).

List of Publications

[A] Houssem Maghrebi, Jean-Luc Danger, Florent Flament and Sylvain Guilley, **Evaluation of Countermeasures Implementation Based on Boolean Masking to Thwart First and Second Order Side-Channel Attacks**, International Conference on Signals, Circuits and Systems (SCS) 2009, pages 1-6, November 6th-7th, Jerba, Tunisia. DOI: 10.1109/ICSCS.2009.5412597.

[B] Houssem Maghrebi, Sylvain Guilley and Jean-Luc Danger, **Entropy-based Power Attack**, IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), IEEE Computer Society, pages 1-6, 2010, June 13-14, Anaheim Convention Center, Anaheim, CA, USA. DOI: 10.1109/HST.2010.5513124.

[C] Florent Flament, Houssem Maghrebi, Moulay Aziz Elaabid, Jean-Luc Danger, Sylvain Guilley and Laurent Sauvage, **About Probability Density Function Estimation for Side Channel Analysis**, International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE), 2010, February 4-5, pages 15-23, Darmstadt, Germany. http://cosade2010.cased.de/files/proceedings/cosade2010_paper_4.pdf.

[D] Houssem Maghrebi, Jean-Luc Danger and Sylvain Guilley, **Leakage Squeezing Countermeasure Against High Order Attacks**, International Workshops on Cryptographic Architectures Embedded in Reconfigurable Devices (CryptArchi), 2010, June 27-30, Gif-sur-Yvette, France. http://labh-curien.univ-st-etienne.fr/cryptarchi/workshop10/abstracts/maghrebi.pdf.

[E] Sylvain Guilley, Houssem Maghrebi, Youssef Souissi, Laurent Sauvage and Jean-Luc Danger, **Quantifying the Quality of Side-Channel Acquisitions**, International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE), 2011, February 24-25, pages 16-28, Darmstadt, Germany. http://cosade2011.cased.de/files/2011/cosade2011_talk2_paper.pdf.

[F] Sylvain Guilley, Houssem Maghrebi, Aziz Elaabid, Shivam Bhasin, Youssef Souissi, Nicolas Debande, Laurent Sauvage and Jean-Luc Danger, **Vade Mecum on Side-Channels Attacks and Countermeasures for the Designer and the Evaluator**, Design & Technologies of Integrated Systems (DTIS), IEEE, 2011, March 6-8, Athens, Greece. DOI: 10.1109/DTIS.2011.5941419. http://hal.archives-ouvertes.fr/hal-00579020/en/.

[G] Houssem Maghrebi, Sylvain Guilley and Jean-Luc Danger, **Leakage Squeezing Countermeasure Against High-Order Attacks**, Workshop in Information Security Theory and Practice (WISTP), LNCS, volume 6633, pages 208-223, Springer, June 1-3 2011, Heraklion, Greece. DOI: 10.1007/978-3-642-21040-2_14. "**BEST PAPER AWARD**".

[H] Houssem Maghrebi, Sylvain Guilley and Jean-Luc Danger, **Formal Security Evaluation of Hardware Boolean Masking against Second-Order Attacks**, IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), IEEE Computer Society, pages 40-46, 2011, June 5-6, Convention Center, San Diego, CA, USA.

[I] Houssem Maghrebi, Emmanuel Prouff, Sylvain Guilley and Jean-Luc Danger, **A First-Order Leak-Free Masking Countermeasure**, CT-RSA, Springer, LNCS, 2012, 7178, pages 156-170, February, San Francisco, CA, USA. DOI: 10.1007/978-3-642-27954-6_10.

[J] Houssem Maghrebi, Emmanuel Prouff, Sylvain Guilley and Jean-Luc Danger, **Register Leakage Masking Using Gray Code**, IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), IEEE Computer Society, 2012, June 2-3, Moscone Center, San Francisco, CA, USA.

[K] Houssem Maghrebi, Sylvain Guilley, Claude Carlet and Jean-Luc Danger, **Optimal First-Order Masking with Linear and Non-Linear Bijections**, AFRICACRYPT (Aikaterini Mitrokotsa and Serge Vaudenay, eds.), Lecture Notes in Computer Science, volume 7374, Springer, 2012, pages 360-377.

[L] Claude Carlet, Jean-Luc Danger, Sylvain Guilley and Houssem Maghrebi, **Leakage Squeezing of Order Two**, INDOCRYPT 2012, December 9-12, Springer LNCS number 7668, pages 120-139, Kolkata, India.

[M] Houssem Maghrebi, Olivier Rioul, Sylvain Guilley and Jean-Luc Danger, **Comparison between Side Channel Analysis Distinguishers**, ICICS 2012, October 29-31, Springer LNCS volume 7618, pages 331-340, Hong Kong.

Bibliography

[14001] Federal Information Processing Standards (FIPS) Publication 140-2, *Security requirements for cryptographic modules*, May 25 2001, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf. (Cited on page 7.)

[14009] NIST FIPS (Federal Information Processing Standards) publication 140-3, *Security Requirements for Cryptographic Modules (Draft, Revised)*, 09/11 2009, http://csrc.nist.gov/groups/ST/FIPS140_3/, p. 63. (Cited on page 10.)

[AG01] Mehdi-Laurent Akkar and Christophe Giraud, *An Implementation of DES and AES Secure against Some Attacks*, Proceedings of CHES'01 (LNCS, ed.), LNCS, vol. 2162, Springer, May 2001, Paris, France, pp. 309–318. (Cited on pages xxxvii and 74.)

[AKM+08] Toru Akishita, Masanobu Katagi, Yoshikazu Miyato, Asami Mizuno, and Kyoji Shibutani, *A Practical DPA Countermeasure with BDD Architecture*, CARDIS, Lecture Notes in Computer Science, vol. 5189, Springer, Sept 2008, London, UK, pp. 206–217. (Cited on page 21.)

[BCO04] Éric Brier, Christophe Clavier, and Francis Olivier, *Correlation Power Analysis with a Leakage Model*, CHES, LNCS, vol. 3156, Springer, August 11–13 2004, Cambridge, MA, USA, pp. 16–29. (Cited on pages xxxvii, 17, 19, 41 and 71.)

[BGEC04] Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi, *Cryptanalysis of a White Box AES Implementation*, Selected Areas in Cryptography, 2004, pp. 227–240. (Cited on page 140.)

[BGP+11] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon, *Mutual Information Analysis: a Comprehensive Study*, J. Cryptology **24** (2011), no. 2, 269–291. (Cited on pages 18, 40, 41 and 125.)

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe, *PRESENT: An Ultra-Lightweight Block Cipher*, CHES, Lecture Notes in Computer Science, vol. 4727, Springer, September 10-13 2007, Vienna, Austria, pp. 450–466. (Cited on page 57.)

[BP10] Olivier Benoît and Thomas Peyrin, *Side-Channel Analysis of Six SHA-3 Candidates*, CHES, Lecture Notes in Computer Science, vol. 6225, Springer, August 17-20 2010, Santa Barbara, CA, USA, pp. 140–157. (Cited on page 60.)

[BS97] Eli Biham and Adi Shamir, *Differential Fault Analysis of Secret Key Cryptosystems*, CRYPTO, LNCS, vol. 1294, Springer, August 1997, Santa Barbara, California, USA. DOI: 10.1007/BFb0052259, pp. 513–525. (Cited on page 10.)

[BZ08] Karthik Baddam and Mark Zwolinski, *Divided Backend Duplication Methodology for Balanced Dual Rail Routing*, CHES (Washington, DC, USA), LNCS, vol. 5154, Springer, aug 2008, DOI: 10.1007/978-3-540-85053-3_25, pp. 396–410. (Cited on page 21.)

[Car10a] Claude Carlet, *Boolean Functions for Cryptography and Error Correcting Codes: Chapter of the monography Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, Cambridge University Press, Y. Crama and P. Hammer eds, 2010, Preliminary version available at (http://www.math.univ-paris13.fr/ carlet/chap-fcts-Bool-corr.pdf), pp. 257–397. (Cited on page 117.)

[Car10b] ———, *Vectorial Boolean Functions for Cryptography: Chapter of the monography Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, Cambridge University Press, Y. Crama and P. Hammer eds, 2010, Preliminary version available at (http://www.math.univ-paris13.fr/ carlet/pubs.html), pp. 398–469. (Cited on page 94.)

[CCCS91] Paul Camion, Claude Carlet, Pascale Charpin, and Nicolas Sendrier, *On Correlation-Immune Functions*, CRYPTO (Joan Feigenbaum, ed.), Lecture Notes in Computer Science, vol. 576, Springer, 1991, pp. 86–100. (Cited on page 120.)

[CDGM12] Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi, *Leakage Squeezing of Order Two*, INDOCRYPT, LNCS, vol. 7668, Springer, December 9-12 2012, Kolkata, India, pp. 120–139. (Cited on page 145.)

[CEJvO02] Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot, *A White-Box DES Implementation for DRM Applications*, Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002, LNCS, vol. 2696, Springer, 2002, pp. 1–15. (Cited on page 140.)

[CGKS12] Claude Carlet, Philippe Gaborit, Jon-Lark Kim, and Patrick Solé, *A new class of codes for boolean masking of cryptographic computations*, IEEE Transactions on Information Theory **58** (2012), no. 9, 6000–6011. (Cited on pages 122, 124, 125, 147 and 164.)

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi, *Towards Sound Approaches to Counteract Power-Analysis Attacks*, CRYPTO, LNCS, vol. 1666, Springer, August 15-19 1999, Santa

Barbara, CA, USA. ISBN: 3-540-66347-9. (Cited on pages xxxvii, 22, 67 and 75.)

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi, *Template Attacks*, CHES, LNCS, vol. 2523, Springer, August 2002, San Francisco Bay (Redwood City), USA, pp. 13–28. (Cited on pages 16 and 18.)

[CT06] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*, Wiley-Interscience, July 18 2006. (Cited on pages 52 and 54.)

[CZ06] Zhimin Chen and Yujie Zhou, *Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage*, CHES, LNCS, vol. 4249, Springer, October 10-13 2006, Yokohama, Japan, http://dx.doi.org/10.1007/11894063_20, pp. 242–254. (Cited on page 21.)

[Del73] Philippe Delsarte, *An algebraic approach to the association schemes of coding theory*, Ph.D. thesis, Université Catholique de Louvain, Belgium, 1973. MR MR0384310 (52 #5187) (Cited on page 121.)

[Dic11] Markus Dichtl, *A new method of black box power analysis and a fast algorithm for optimal key search*, J. Cryptographic Engineering **1** (2011), no. 4, 255–264. (Cited on page 148.)

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum-likelihood from incomplete data via the EM algorithm*, Journal of Royal Statistical Society B **39** (1977), 1–38. (Cited on page 38.)

[DM09] A. Das and C.E.V. Madhavan, *Public-key cryptography: Theory and practice*, Pearson Education, 2009. (Cited on page 9.)

[DPRS11] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert, *Univariate side channel attacks and leakage modeling*, J. Cryptographic Engineering **1** (2011), no. 2, 123–144. (Cited on pages 33 and 139.)

[Dwo01] NIST/ITL/CSD (Morris Dworkin), *Recommendation for Block Cipher Modes of Operation.* Methods and Techniques, December 2001, (Online reference). (Cited on page 5.)

[Eck85] Wim Van Eck, *Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?*, Computers Security, 1985. (Cited on page 14.)

[Ed.09] Neil J. A. Sloane Ed., *The On-Line Encyclopedia of Integer Sequences, published electronically at http://www.research.att.com/~njas/sequences/, Sequence A008277: Triangle of Stirling numbers of 2nd kind, $S2(n,k)$, $n \geq 1$, $1 \leq k \leq n$*, 2009, http://oeis.org/A008277. (Cited on page 118.)

[FG05] Wieland Fischer and Berndt M. Gammel, *Masking at Gate Level in the Presence of Glitches*, CHES, Lecture Notes in Computer Science, vol. 3659, Springer, August 29 – September 1 2005, Edinburgh, UK, pp. 187–200. (Cited on page 21.)

[FMPR10] Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain, *Affine Masking against Higher-Order Side Channel Analysis*, Selected Areas in Cryptography (Alex Biryukov, Guang Gong, and Douglas R. Stinson, eds.), LNCS, vol. 6544, Springer, 2010, pp. 262–280. (Cited on page 164.)

[FST92] G. David Forney, Jr., Neil J. A. Sloane, and Mitchell D. Trott, *The Nordstrom-Robinson Code is the Binary Image of the Octacode*, Coding and Quantization: DIMACS / IEEE Workshop (R. Calderbank Amer. Math. Soc., Jr. G. D. Forney, and N. Moayeri (Eds), eds.), October 19-21 1992, pp. 19–26. (Cited on page 124.)

[GBPV10] Benedikt Gierlichs, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede, *Revisiting Higher-Order DPA Attacks: Multivariate Mutual Information Analysis*, CT-RSA, LNCS, vol. 5985, Springer, March 1-5 2010, San Francisco, CA, USA, pp. 221–234. (Cited on pages xxxvii, 22, 41, 68 and 164.)

[GBTP08] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel, *Mutual information analysis*, CHES, 10th International Workshop, Lecture Notes in Computer Science, vol. 5154, Springer, August 10-13 2008, Washington, D.C., USA, pp. 426–442. (Cited on pages 17, 18, 37, 40, 44, 48 and 57.)

[GCS+08] Sylvain Guilley, Sumanta Chaudhuri, Laurent Sauvage, Tarik Graba, Jean-Luc Danger, Philippe Hoogvorst, Vinh-Nga Vong, and Maxime Nassar, *Place-and-Route Impact on the Security of DPL Designs in FPGAs*, HOST (Hardware Oriented Security and Trust), IEEE (Anaheim, CA, USA), jun 2008, pp. 29–35. (Cited on page 21.)

[GHP04] Sylvain Guilley, Philippe Hoogvorst, and Renaud Pacalet, *Differential Power Analysis Model and some Results*, Proceedings of WCC/CARDIS (Kluwer, ed.), Aug 2004, Toulouse, France. DOI: 10.1007/1-4020-8147-2_9, pp. 127–142. (Cited on pages 41 and 60.)

[GM11] Tim Güneysu and Amir Moradi, *Generic side-channel countermeasures for reconfigurable devices*, in Preneel and Takagi [PT11], pp. 33–48. (Cited on page 130.)

[GO04] T.Aaron Gulliver and Patric R.J. Östergård, *Binary optimal linear rate 1/2 codes*, Discrete Mathematics **283** (2004), no. 1-3, 255 – 261. (Cited on page 122.)

[GP99] Louis Goubin and Jacques Patarin, *DES and Differential Power Analysis. The "Duplication" Method*, CHES, LNCS, Springer, Aug 1999, Worcester, MA, USA, pp. 158–172. (Cited on pages xxxvii and 67.)

[Gra07] Markus Grassl, *Bounds on the minimum distance of linear codes and quantum codes*, Online available at `http://www.codetables.de/`, 2007, Accessed on 2012-07-23. (Cited on page 153.)

[GSD+11] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Nidhal Selmane, and Denis Réal, *Performance Evaluation of Protocols Resilient to Physical Attacks*, HOST, IEEE Computer Society, June 5-6 2011, Convention Center, San Diego, California, USA. DOI: 10.1109/HST.2011.5954995, pp. 51–56. (Cited on page 107.)

[GSF+10] Sylvain Guilley, Laurent Sauvage, Florent Flament, Philippe Hoogvorst, and Renaud Pacalet, *Evaluation of Power-Constant Dual-Rail Logics Counter-Measures against DPA with Design-Time Security Metrics*, IEEE Transactions on Computers **9** (2010), no. 59, 1250–1263, DOI: 10.1109/TC.2010.104. (Cited on page 21.)

[HPS99] Helena Handschuh, Pascal Paillier, and Jacques Stern, *Probing Attacks on Tamper-Resistant Devices*, CHES, LNCS, vol. 1717, Springer, August 12-13 1999, Worcester, MA, USA, pp. 303–315. (Cited on page 11.)

[HSS12] Annelie Heuser, Werner Schindler, and Marc Stöttinger, *Revealing side-channel issues of complex circuits by enhanced leakage models*, DATE, 2012, pp. 1179–1184. (Cited on page 60.)

[iso] *ISO/IEC 18033-3:2010*, Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers. (Cited on page 122.)

[Jap] Japanese RCIS-AIST: `http://www.risec.aist.go.jp/project/sasebo/`. (Cited on page 29.)

[JPS05] Marc Joye, Pascal Paillier, and Berry Schoenmakers, *On Second-Order Differential Power Analysis*, CHES, LNCS, vol. 3659, Springer, August 29 – September 1st 2005, Edinburgh, UK, pp. 293–308. (Cited on page 22.)

[KJJ96] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, Proceedings of CRYPTO'96, LNCS, vol. 1109, Springer-Verlag, 1996, (PDF), pp. 104–113. (Cited on pages xxxvii and 14.)

[KJJ99] ———, *Differential Power Analysis*, Proceedings of CRYPTO'99, LNCS, vol. 1666, Springer-Verlag, 1999, pp. 388–397. (Cited on pages xxxvii, 10, 14 and 17.)

[Koc05] Paul C. Kocher, *Design and Validation Strategies for Obtaining Assurance in Countermeasures to Power Analysis and Related Attacks*, September 26-29 2005, Honolulu, Hawai, USA; NIST's Physical Security Testing Workshop. Website: http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-3/physec/physecdoc.html. (Cited on page 107.)

[KV10] Jens-Peter Kaps and Rajesh Velegalati, *DPA Resistant AES on FPGA Using Partial DDL*, FCCM: 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, IEEE Computer Society, May 02–May 04 2010, Charlotte, North Carolina, USA. DOI: 10.1109/FCCM.2010.49, pp. 273–280. (Cited on page 21.)

[LB10] Thanh-Ha Le and Maël Berthier, *Mutual Information Analysis under the View of Higher-Order Statistics*, IWSEC (Isao Echizen, Noboru Kunihiro, and Ryôichi Sasaki, eds.), LNCS, vol. 6434, Springer, 2010, pp. 285–300. (Cited on page 73.)

[LCC+06] Thanh-Ha Le, Jessy Clédière, Cécile Canovas, Bruno Robisson, Christine Servière, and Jean-Louis Lacoume, *A proposition for correlation power analysis enhancement*, Proceedings of the 8th international conference on Cryptographic Hardware and Embedded Systems (Berlin, Heidelberg), CHES'06, Springer-Verlag, 2006, pp. 174–186. (Cited on page 33.)

[Man04] Stefan Mangard, *Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness*, CT-RSA, Lecture Notes in Computer Science, vol. 2964, Springer, 2004, San Francisco, CA, USA, pp. 222–235. (Cited on pages 19, 21 and 86.)

[MCGD12] Houssem Maghrebi, Claude Carlet, Sylvain Guilley, and Jean-Luc Danger, *Optimal first-order masking with linear and non-linear bijections*, AFRICACRYPT (Aikaterini Mitrokotsa and Serge Vaudenay, eds.), Lecture Notes in Computer Science, vol. 7374, Springer, 2012, pp. 360–377. (Cited on page 111.)

[MDFG09] Houssem Maghrebi, Jean-Luc Danger, Florent Flament, and Sylvain Guilley, *Evaluation of Countermeasures Implementation Based on Boolean Masking to Thwart First and Second Order Side-Channel Attacks*, SCS, IEEE, November 6–8 2009, Jerba, Tunisia. DOI: 10.1109/ICSCS.2009.5412597, pp. 1–6. (Cited on pages 27, 102 and 107.)

[MDS02] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan, *Examining Smart-Card Security under the Threat of Power Analysis Attacks*, IEEE Trans. Computers **51** (2002), no. 5, 541–552. (Cited on page 19.)

[Mes00a] Thomas S. Messerges, *Securing the AES Finalists Against Power Analysis Attacks*, Fast Software Encryption'00, Springer-Verlag, April 2000, New York, pp. 150–164. (Cited on page 74.)

[Mes00b] ———, *Using Second-Order Power Analysis to Attack DPA Resistant Software*, CHES, LNCS, vol. 1965, Springer-Verlag, August 17-18 2000, Worcester, MA, USA, pp. 238–251. (Cited on pages xxxvii, 22 and 27.)

[MGCD11] Houssem Maghebi, Sylvain Guilley, Claude Carlet, and Jean-Luc Danger, *Classification of High-Order Boolean Masking Schemes and Improvements of their Efficiency*, Cryptology ePrint Archive, Report 2011/520, September 2011, http://eprint.iacr.org/2011/520. (Cited on page 159.)

[MGD11a] Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger, *Formal Security Evaluation of Hardware Boolean Masking against Second-Order Attacks*, HOST, IEEE Computer Society, June 5-6 2011, Convention Center, San Diego, California, USA. DOI: 10.1109/HST.2011.5954993, pp. 40–46. (Cited on page 77.)

[MGD11b] ———, *Leakage Squeezing Countermeasure Against High-Order Attacks*, WISTP, LNCS, vol. 6633, Springer, June 1-3 2011, Heraklion, Greece. DOI: 10.1007/978-3-642-21040-2_14, pp. 208–223. (Cited on page 111.)

[MGDF10] Houssem Maghrebi, Sylvain Guilley, Jean-Luc Danger, and Florent Flament, *Entropy-based Power Attack*, HOST, IEEE Computer Society, June 13-14 2010, Anaheim Convention Center, Anaheim, CA, USA. DOI: 10.1109/HST.2010.5513124, pp. 1–6. (Cited on page 37.)

[MM12a] Amir Moradi and Oliver Mischke, *Glitch-free Implementation of Masking in Modern FPGAs*, HOST, IEEE Computer Society, June 2-3 2012, Moscone Center, San Francisco, CA, USA. DOI: 10.1109/HST.2012.6224326, pp. 89–95. (Cited on page 164.)

[MM12b] ———, *How far should theory be from practice? - evaluation of a countermeasure*, CHES (Emmanuel Prouff and Patrick Schaumont, eds.), Lecture Notes in Computer Science, vol. 7428, Springer, 2012, pp. 92–106. (Cited on pages 96 and 147.)

[MME10] Amir Moradi, Oliver Mischke, and Thomas Eisenbarth, *Correlation-Enhanced Power Analysis Collision Attack*, CHES, Lecture Notes in Computer Science, vol. 6225, Springer, August 17-20 2010, Santa Barbara, CA, USA, pp. 125–139. (Cited on page 96.)

[MMPS09] Amir Moradi, Nima Mousavi, Christof Paar, and Mahmoud Salmasizadeh, *A Comparative Study of Mutual Information Analysis under*

a Gaussian Assumption, WISA, LNCS, vol. 5932, Springer, August 25-27 2009, Busan, Korea, pp. 193–205. (Cited on page 54.)

[MOP06] Stefan Mangard, Elisabeth Oswald, and Thomas Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer, December 2006, ISBN 0-387-30857-1, http://www.dpabook.org/. (Cited on pages 11, 16, 22 and 67.)

[MOS11] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert, *One for All - All for One: Unifying Standard DPA Attacks*, Information Security, IET **5** (2011), no. 2, 100–111, ISSN: 1751-8709 ; Digital Object Identifier: 10.1049/iet-ifs.2010.0096. (Cited on pages 61, 63, 77 and 86.)

[MPGD12a] Houssem Maghrebi, Emmanuel Prouff, Sylvain Guilley, and Jean-Luc Danger, *A First-Order Leak-Free Masking Countermeasure*, CT-RSA, LNCS, vol. 7178, Springer, February 27 – March 2 2012, San Francisco, CA, USA. DOI: 10.1007/978-3-642-27954-6_10, pp. 156–170. (Cited on page 91.)

[MPGD12b] ———, *Register Leakage Masking Using Gray Code*, HOST, IEEE Computer Society, June 2-3 2012, Moscone Center, San Francisco, CA, USA. DOI: 10.1109/HST.2012.6224316, pp. 37–42. (Cited on page 101.)

[MRGD12] Houssem Maghrebi, Olivier Rioul, Sylvain Guilley, and Jean-Luc Danger, *Comparison between Side Channel Analysis Distinguishers*, CHES, LNCS, vol. 7618, Springer, October 29-31 2012, Hong Kong, pp. 331–340. (Cited on page 47.)

[MS77] F. Jessie MacWilliams and Neil J. A. Sloane, *The Theory of Error-Correcting Codes*, Elsevier, Amsterdam, North Holland, 1977, ISBN: 978-0-444-85193-2. (Cited on page 121.)

[MS06] Stefan Mangard and Kai Schramm, *Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations*, CHES, LNCS, vol. 4249, Springer, October 10-13 2006, Yokohama, Japan, pp. 76–90. (Cited on page 68.)

[NIS99] NIST/ITL/CSD, *Data Encryption Standard. FIPS PUB 46-3*, Oct 1999, http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf. (Cited on page 6.)

[NIS01] ———, *Advanced Encryption Standard (AES). FIPS PUB 197*, Nov 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. (Cited on page 7.)

[NRS08] Svetla Nikova, Vincent Rijmen, and Martin Schläffer, *Secure Hardware Implementation of Non-linear Functions in the Presence of Glitches*, ICISC, Lecture Notes in Computer Science, vol. 5461, Springer, 2008, Seoul, Korea, pp. 218–234. (Cited on pages xxxvii, 21 and 164.)

[OMHT06] Elisabeth Oswald, Stefan Mangard, Christoph Herbst, and Stefan Tillich, *Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers*, in Pointcheval [Poi06], pp. 192–207. (Cited on pages xxxvii and 22.)

[PGA06] Emmanuel Prouff, Christophe Giraud, and Sébastien Aumônier, *Provably Secure S-Box Implementation Based on Fourier Transform*, CHES, LNCS, vol. 4249, Springer, October 10-13 2006, Yokohama, Japan, pp. 216–230. (Cited on page 75.)

[Poi06] David Pointcheval (ed.), *Topics in cryptology - ct-rsa 2006, the cryptographers' track at the rsa conference 2006, san jose, ca, usa, february 13-17, 2006, proceedings*, LNCS, vol. 3860, Springer, 2006. (Cited on pages 175 and 177.)

[PR07] Emmanuel Prouff and Matthieu Rivain, *A Generic Method for Secure SBox Implementation*, WISA (Sehun Kim, Moti Yung, and Hyung-Woo Lee, eds.), Lecture Notes in Computer Science, vol. 4867, Springer, 2007, pp. 227–244. (Cited on pages 22, 74 and 75.)

[PR09] ———, *Theoretical and Practical Aspects of Mutual Information Based Side Channel Analysis*, ACNS (Springer, ed.), LNCS, vol. 5536, June 2-5 2009, Paris-Rocquencourt, France, pp. 499–518. (Cited on pages 18, 37, 44, 54 and 60.)

[PR11] Emmanuel Prouff and Thomas Roche, *Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols*, in Preneel and Takagi [PT11], pp. 63–78. (Cited on page 164.)

[PRB09] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan, *Statistical Analysis of Second Order Differential Power Analysis*, IEEE Trans. Computers **58** (2009), no. 6, 799–811. (Cited on pages xxxvii, 22, 86, 98, 113, 114, 115, 128 and 156.)

[PSDQ05] 'Eric Peeters, François-Xavier Standaert, Nicolas Donckers, and Jean-Jacques Quisquater, *Improved Higher-Order Side-Channel Attacks With FPGA Experiments*, CHES, LNCS, vol. 3659, Springer-Verlag, 2005, Edinburgh, UK, pp. 309–323. (Cited on pages 22, 27, 28, 29 and 93.)

[PSQ07] Éric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater, *Power and electromagnetic analysis: Improved model, consequences*

and comparisons, Integration, The VLSI Journal, special issue on "*Embedded Cryptographic Hardware*" **40** (2007), 52–60, DOI: 10.1016/j.vlsi.2005.12.013. (Cited on pages 97 and 99.)

[PT11] Bart Preneel and Tsuyoshi Takagi (eds.), *Cryptographic hardware and embedded systems - ches 2011 - 13th international workshop, nara, japan, september 28 – october 1, 2011. proceedings*, Lecture Notes in Computer Science, vol. 6917, Springer, 2011. (Cited on pages 170 and 175.)

[RDJ+01] Atri Rudra, Pradeep K. Dubey, Charanjit S. Jutla, Vijay Kumar, Josyula R. Rao, and Pankaj Rohatgi, *Efficient Rijndael Encryption Implementation with Composite Field Arithmetic*, CHES (Çetin Kaya Koç, David Naccache, and Christof Paar, eds.), vol. 2162, Lecture Notes in Computer Science, no. Generators, Springer, 2001, pp. 171–184. (Cited on page 75.)

[Rog11] Phillip Rogaway (ed.), *Advances in cryptology - crypto 2011 - 31st annual cryptology conference, santa barbara, ca, usa, august 14-18, 2011. proceedings*, Lecture Notes in Computer Science, vol. 6841, Springer, 2011. (Cited on pages 178 and 179.)

[Roh] Rohde and Schwarz: `http://www.rohde-schwarz.com/`. (Cited on page 31.)

[RP10] Matthieu Rivain and Emmanuel Prouff, *Provably Secure Higher-Order Masking of AES*, CHES (Stefan Mangard and François-Xavier Standaert, eds.), LNCS, vol. 6225, Springer, 2010, pp. 413–427. (Cited on page 75.)

[RPD09] Matthieu Rivain, Emmanuel Prouff, and Julien Doget, *Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers*, CHES, Lecture Notes in Computer Science, vol. 5747, Springer, September 6-9 2009, Lausanne, Switzerland, pp. 171–188. (Cited on page 21.)

[Sal96] A. Salomaa, *Public-key cryptography*, Texts in theoretical computer science, Springer, 1996. (Cited on page 9.)

[SGD+09] Laurent Sauvage, Sylvain Guilley, Jean-Luc Danger, Yves Mathieu, and Maxime Nassar, *Successful Attack on an FPGA-based WDDL DES Cryptoprocessor Without Place and Route Constraints*, DATE (Nice, France), IEEE Computer Society, apr 2009, pp. 640–645. (Cited on page 99.)

[SGV08] François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede, *Partition vs. Comparison Side-Channel Distinguishers: An*

Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices, ICISC, LNCS, vol. 5461, Springer, December 3-5 2008, Seoul, Korea, pp. 253–267. (Cited on pages 28 and 44.)

[Sil86] B. W. Silverman, *Density estimation for statistics and data analysis*, Chapman & Hall/CRC, April 1986. (Cited on page 38.)

[SLP05] Werner Schindler, Kerstin Lemke, and Christof Paar, *A Stochastic Model for Differential Side Channel Cryptanalysis*, CHES (LNCS, ed.), LNCS, vol. 3659, Springer, Sept 2005, Edinburgh, Scotland, UK, pp. 30–46. (Cited on pages 16, 18 and 131.)

[SMTM01] Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh, *A Compact Rijndael Hardware Architecture with S-Box Optimization*, ASIACRYPT (Colin Boyd, ed.), Lecture Notes in Computer Science, vol. 2248, Springer, 2001, pp. 239–254. (Cited on page 92.)

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*, EUROCRYPT, LNCS, vol. 5479, Springer, April 26-30 2009, Cologne, Germany, pp. 443–461. (Cited on pages 19, 20, 57, 59, 77, 96, 109 and 125.)

[Sno73] Stephen L. Snover, *The uniqueness of the Nordstrom-Robinson and the Golay binary codes*, Ph.D. thesis, Department of Mathematics, Michigan State University, USA, 1973, p. 150. (Cited on page 124.)

[SP00] George Saon and Mukund Padmanabhan, *Minimum Bayes Error Feature Selection for Continuous Speech Recognition*, ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 13, MIT Press, 2000, pp. 800–806. (Cited on page 49.)

[SP06] Kai Schramm and Christof Paar, *Higher Order Masking of the AES*, in Pointcheval [Poi06], pp. 208–225. (Cited on pages 22 and 87.)

[SRQ06] François-Xavier Standaert, Gaël Rouvroy, and Jean-Jacques Quisquater, *FPGA Implementations of the DES and Triple-DES Masked Against Power Analysis Attacks*, FPL, IEEE, August 2006, Madrid, Spain. (Cited on pages 29, 92, 98 and 141.)

[SSI04] Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa, *Random Switching Logic: A Countermeasure against DPA based on Transition Probability*, 2004, `http://eprint.iacr.org/2004/346`. (Cited on page 21.)

[SVCO+10] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard, *The World is Not Enough: Another Look on Second-*

Order DPA, ASIACRYPT, LNCS, vol. 6477, Springer, December 5-9 2010, Singapore. http://www.dice.ucl.ac.be/~fstandae/PUBLIS/88.pdf, pp. 112–129. (Cited on pages xxxviii, 41, 44, 75, 78, 79, 82, 83, 86 and 87.)

[SVKH10] Shaunak Shah, Rajesh Velegalati, Jens-Peter Kaps, and David Hwang, *Investigation of DPA Resistance of Block RAMs in Cryptographic Implementations on FPGAs*, ReConFig (Viktor K. Prasanna, Jürgen Becker, and René Cumplido, eds.), IEEE Computer Society, 2010, pp. 274–279. (Cited on page 112.)

[Tri03] Elena Trichina, *Combinational Logic Design for AES SubBytes Transformation on Masked Data*, 2003, Not published elsewhere. e.v.trichina@samsung.com 12368 received 11 Nov 2003. (Cited on pages 21 and 75.)

[Vau94] Serge Vaudenay, *On the Need for Multipermutations: Cryptanalysis of MD4 and SAFER*, FSE (Bart Preneel, ed.), Lecture Notes in Computer Science, vol. 1008, Springer, 1994, pp. 286–297. (Cited on page 122.)

[VCGRS11] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert, *An optimal Key Enumeration Algorithm and its Application to Side-Channel Attacks*, Cryptology ePrint Archive, Report 2011/610, 2011, http://eprint.iacr.org/2011/610/. (Cited on page 148.)

[VCS09] Nicolas Veyrat-Charvillon and François-Xavier Standaert, *Mutual Information Analysis: How, When and Why?*, CHES, LNCS, vol. 5747, Springer, September 6-9 2009, Lausanne, Switzerland, pp. 429–443. (Cited on pages 18, 57, 96 and 97.)

[VCS11] ———, *Generic Side-Channel Distinguishers: Improvements and Limitations*, in Rogaway [Rog11], pp. 354–372. (Cited on page 127.)

[Wan96] M.P. Wand, *Data-based choice of histogram bin width*, Statistics working paper, Australian Graduate School of Management, 13th May 1996. (Cited on page 38.)

[web] *Substitution-permutation network :http://www.rohde-schwarz.com/*. (Cited on page 8.)

[Wel77] Jon A. Wellner, *A Glivenko-Cantelli theorem and strong laws of large numbers for functions of order statistics*, Ann. Statist. **5** (1977), no. 3, 473–480. MR MR0651528 (58 #31359a) (Cited on page 60.)

[WL03] Robert Ware and Frank Lad, *Approximating the distribution for sums of products of normal variables*, Population English Edition **15** (2003), no. 1978, 1–50. (Cited on page 69.)

[WMGP07] Brecht Wyseur, Wil Michiels, Paul Gorissen, and Bart Preneel, *Cryptanalysis of White-Box DES Implementations with Arbitrary External Encodings*, Selected Areas in Cryptography, 14th International Workshop, SAC 2007, LNCS, vol. 4876, Springer, 2007, pp. 264–277. (Cited on page 140.)

[WO11] Carolyn Whitnall and Elisabeth Oswald, *A Comprehensive Evaluation of Mutual Information Analysis Using a Fair Evaluation Framework*, in Rogaway [Rog11], pp. 316–334. (Cited on pages 60 and 69.)

[WOM11] Carolyn Whitnall, Elisabeth Oswald, and Luke Mather, *An Exploration of the Kolmogorov-Smirnov Test as a Competitor to Mutual Information Analysis*, CARDIS (Emmanuel Prouff, ed.), Lecture Notes in Computer Science, vol. 7079, Springer, 2011, pp. 234–251. (Cited on pages 18, 60 and 63.)

[WW04] Jason Waddle and David Wagner, *Towards Efficient Second-Order Power Analysis*, CHES, LNCS, vol. 3156, Springer, 2004, Cambridge, MA, USA, pp. 1–15. (Cited on pages 22, 27, 71, 93, 114, 124, 148, 149 and 150.)

[XM88] Guo-Zhen Xiao and James L. Massey, *A spectral characterization of correlation-immune combining functions*, IEEE Transactions on Information Theory **34** (1988), no. 3, 569–571. (Cited on page 121.)

[Yan09] S.Y. Yan, *Primality testing and integer factorization in public-key cryptography*, Advances in information security, Springer, 2009. (Cited on page 9.)