Attaques par Canaux Auxiliaires :
Evaluations de Sécurité à Court et Long Terme

François-Xavier Standaert
UCL Crypto Group, Louvain-la-Neuve, Belgique

Connues depuis la fin des années 1990, les attaques par canaux auxiliaires sont devenues un élément important de la sécurité des implémentations cryptographiques embarquées. Elles peuvent exploiter différents types d'information physique (consommation électrique, rayonnement électromagnétique, temps de calcul, …) et leur efficacité a été démontrée dans de nombreux cas d'application. Ces attaques posent un problème fondamental à la théorie cryptographique : alors que les algorithmes (de chiffrement, d'authentification, de signature, …) utilisés dans des systèmes sécurisés sont habituellement considérés comme des éléments de confiance, la mise en œuvre de ces algorithmes permet des attaques simples et dévastatrices. Se convaincre (et convaincre un tiers) de la sécurité d'une implémentation contre ces attaques est en outre complexe à cause de leur dépendance en des détails de conception pas toujours publics ni bien compris. Tour d'horizon académique des solutions actuellement déployées et des évolutions possibles…

MOTS CLES : SECURITE EMBARQUEE, ATTAQUES PAR CANAUX AUXILIAIRES, CERTIFICATION

Partons d'un cas simple pour se convaincre que le problème est difficile. Soit un chiffrement par bloc **E**, par exemple le standard AES (Advanced Encryption Standard). En cryptanalyse classique, un adversaire borné en temps de calcul *T* essaye de « casser » le chiffrement à partir d'une quantité *D* de messages interceptés. La cryptographie a pour hypothèse qu'il est possible de construire des algorithmes tels qu'indépendamment de la stratégie de l'adversaire, il lui sera impossible de récupérer l'information protégée si il n'utilise pas plus qu'un temps de calcul *T* et une quantité de données *D*, qu'on peut faire croître exponentiellement grâce à un paramètre de sécurité : la taille de la clé secrète n. Typiquement, on suppose que la meilleure attaque sera de complexité similaire à une recherche exhaustive parmi les 2^n clés possibles. Imaginons maintenant que l'adversaire n'intercepte pas seulement des textes clairs et chiffrés, mais également des mesures de consommation électrique de l'implémentation les manipulant. Supposons en outre que cette consommation électrique soit corrélée avec les valeurs intermédiaires manipulées [KJJ99], et qu'à chaque chiffrement, une quantité d'information de λ bits soit révélée sur la clé. En faisant l'hypothèse que ces informations sont indépendantes pour chaque texte chiffré, la clé sera complètement divulguée après $\left\lceil \frac{n}{\lambda} \right\rceil$ mesures.

Cette description (simplifiée) illustre les deux problèmes principaux posés par les attaques par canaux auxiliaires. D'une part *ces attaques sont extrêmement puissantes*, car elles réduisent la sécurité de la clé secrète exponentiellement en le nombre de mesures effectuées par l'adversaire (alors qu'une recherche exhaustive ne la réduit que linéairement en le nombre de clés testées par celui-ci). D'autre part, *la quantité d'information extraite est difficile à quantifier* et à borner, car elle dépend de la qualité des mesures et des hypothèses qu'un adversaire est capable de poser sur l'implémentation qu'il veut compromettre.

Une conséquence de ces observations est que l'évaluation de la sécurité contre les attaques auxiliaires est un processus complexe : il ne s'agit en effet pas d'un problème mathématique aux contours bien définis, mais d'un problème physique nécessitant une approche interdisciplinaire. Il implique une succession d'étapes parfois difficiles à décrire formellement. Les techniques d'évaluations actuelles se basent dès lors sur une série d'heuristiques dont il est important de comprendre l'impact en termes de risques.

Etape 1 : définir les capacités de l'adversaire

En cryptanalyse classique, les capacités de l'adversaire sont faciles à définir. On suppose que les spécifications de l'algorithme sont publiques (cf. principes de Kerckhoffs[1]). Comme mentionné ci-dessus, l'adversaire peut faire *D* requêtes de chiffrement et/ou de déchiffrement et utiliser une puissance de calcul *T* pour exploiter ces données. Ceci permet d'évaluer comment le coût d'une attaque évoluera au cours du temps[2] : la puissance de calcul de l'adversaire augmentant avec les années (cf. loi de Moore[3]).

La situation est différente dans le cas d'une attaque physique. D'une part ces attaques nécessitent du matériel de mesure (probes, oscilloscopes, amplificateurs à faible bruit, …) dont l'évolution du coût est beaucoup moins étudiée que celle de la puissance de calcul. La qualité de l'environnement de mesure est ainsi une première source de risque : une évaluation basée sur de mauvaises mesures surévaluera le niveau de sécurité. Dans ce cadre, l'utilisation d'environnements standardisés, utilisables à titre de référence (et la comparaison de nouveaux résultats à ces derniers), semble être une piste intéressante pour minimiser ce risque. D'autre part, la question de la description des implémentations se pose de façon critique : il n'est en effet pas (encore) courant que des circuits sécurisés soient basés sur une architecture ouverte, pour laquelle tous les détails de conception sont rendus publics. Au sein des schémas de certification actuels de type « Critères Communs »[4], ces éléments sont évalués dans le « potentiel de l'attaque » : si attaquer une implémentation nécessite du matériel coûteux et un accès à des détails de conception, elle sera considérée comme plus sûre que si l'attaque peut être réalisée à faible coût et sans connaissance de ces détails.

D'un point de vue cryptographique, il faut ici noter la contradiction entre ces schémas de certification et les principes de Kerckhoffs, ainsi que les risques que cette contradiction implique. Considérer la connaissance de détails de conception d'une implémentation comme faisant partie du potentiel de l'adversaire revient à considérer ces détails comme un secret à long terme (alors qu'il ne sera typiquement pas protégé par les contremesures standard aux attaques par canaux auxiliaires). La divulgation de ces détails peut donc mettre à mal les conclusions de l'évaluation - problème connu en cryptanalyse classique [BBK08,G+08].

Enfin, il faut noter que la définition des requêtes permises à l'adversaire est également plus critique dans le cas d'attaques physiques. En particulier, une question importante est de savoir si l'adversaire peut (dans une phase préliminaire à l'attaque) utiliser une implémentation de référence dont il contrôlerait la clé (et éventuellement l'aléa interne utilisé par des contremesures) afin d'estimer un modèle pour cette dernière : on parlera d'*attaques profilées* sur la clé et/ou l'aléa si c'est possible, d'*attaques non profilées* autrement. Comme démontré par Whitnall et al., une évaluation « au pire cas » (dont le but est de s'approcher des garanties cryptographiques standard où la sécurité est indépendante de la stratégie de l'adversaire) nécessite formellement de travailler dans un contexte d'attaques profilées [WOS14].

Etape 2 : choisir un type d'évaluation

En simplifiant, on peut distinguer trois types d'évaluations dans la littérature actuelle : les évaluations basées sur de la *détection d'information*, sur des *attaques concrètes* et sur des *preuves* (ou arguments) *de sécurité*. Elles se distinguent principalement par la question posée à l'évaluateur : les mesures obtenues par l'adversaire contiennent-elles de l'information liée aux données (indépendamment du fait que cette information soit exploitable) pour la détection ; les mesures obtenues peuvent-elles être exploitées par un adversaire borné (en un temps *T* et avec une quantité de mesures *D* accessibles à l'évaluateur) pour les attaques concrètes ; peut-on se convaincre qu'aucune attaque utilisant un temps *T* et une quantité de mesures *D* supérieurs aux ressources de l'évaluateur n'est possible pour les preuves. Les caractéristiques de ces trois types d'évaluations sont résumées en Figure 1, dont on peut extraire les conclusions suivantes.

	détection	*attaques*	*preuves*
méthodologie	*test de conformité*	*analyse experte*	*analyse experte extrapolée*
type de résultat	*concluant ou non-concluant*	*concluant ou non concluant*	*concluant*
complexités évaluées	*données (mesures) uniquement*	*données (mesures) et temps*	*données (mesures) et temps*
objectif	*sécurité minimum*	*sécurité à court terme*	*sécurité à long terme*

Figure 1. Types d'évaluations contre les attaques par canaux auxiliaires.

[1] https://fr.wikipedia.org/wiki/Principe_de_Kerckhoffs
[2] Voir par exemple https://www.keylength.com/fr/
[3] https://fr.wikipedia.org/wiki/Loi_de_Moore
[4] https://www.commoncriteriaportal.org/

D’une part, l’élément principal séparant les évaluations par détection et par attaques des évaluations par preuve est la *durée de validité des conclusions*. Les premières n’apportent aucune garantie au-delà de la quantité de données *D* et du temps de calcul *T* accessibles à l’évaluateur ; les secondes ont pour objectif d’apporter des garanties à long terme, pour des complexités bien supérieures, comme typiquement requis pour des implémentations à très hauts niveaux de sécurité. A ce sujet, il faut noter que les solutions actuellement mises en œuvre pour la certification de produits cryptographiques industriels sont principalement de type détection et attaques. D’autre part, l’élément principal séparant les évaluations par détection des évaluations par attaques et par preuve est le *type d’expertise* requis pour l’évaluateur. Les premières ont pour but d’être utilisables de façon presque automatique et sans haut niveau d’expertise ; les secondes nécessitent le concours d’experts du domaine afin d’analyser des faiblesses éventuelles, spécifiques aux implémentations étudiées. Notons en outre que, comme déjà mentionné, l’élément commun de toutes ces évaluations est qu’elles s’accompagnent d’un important facteur de risque (de surévaluation de la sécurité) dont l’analyse est essentielle et dont nous rediscuterons en conclusion de cet article.

Par ailleurs, et concrètement, la détection ne s’intéresse qu’à la présence d’information (et ses conclusions sont dès lors indépendantes de la puissance de calcul qu’un adversaire pourrait utiliser afin de récupérer une clé, par exemple via de l’énumération [PSG16]). La complexité temporelle est par contre intégrée aux évaluations par attaques et par preuve. Enfin les évaluations par détection et attaques peuvent être concluantes (en exhibant une détection ou une attaque) ou non concluantes, l’interprétation de ce second cas étant généralement compliquée. En effet, ne pas parvenir à exhiber une détection ou une attaque ne garantit aucunement qu’elles ne soient pas possibles, avec plus de mesures ou de meilleures hypothèses de travail. A l’opposé, les évaluations par preuve ne peuvent qu’être concluantes : elles se basent en effet généralement sur l’analyse d’implémentations « à sécurité réduite » et l’extrapolation des résultats obtenus dans ce cadre pour prédire la sécurité avec toutes les fonctions de sécurité actives. Sans résultats concluants sur l’implémentation simplifiée, ce genre d’extrapolation n’est donc pas possible.

Etape 3 : choisir des outils et des métriques

Le type d’évaluation influence les outils (de mesure, statistiques, cryptographiques) et les métriques à utiliser lors de l’évaluation, ainsi que le degré de liberté laissé à l’évaluateur dans le choix d’outils et métriques.

Par exemple, les évaluations par détection ayant pour objectif global de garantir un niveau de sécurité minimum avec un minimum d’expertise, elles se basent habituellement sur une méthodologie de type « *test de conformité* ». Les analyses à réaliser par l’évaluateur peuvent être spécifiées précisément, pourraient éventuellement être standardisées et doivent être facilement reproductibles. Une illustration populaire de ce type d’approche est la détection basée sur l’estimation de moments statistiques [SM16]. De nombreuses variantes ont été introduites dans la littérature et s’expriment généralement comme un compromis entre l’efficacité de la détection et sa généricité (capture-t-elle toute forme d’information?).

A l’opposé, il est difficile (et probablement pas souhaitable) de spécifier précisément les outils à utiliser pour une évaluation par attaques qui correspond à une méthodologie de type « *analyse experte* ». Ces dernières sont en effet liées de façon inhérente à l’implémentation et aux contremesures à évaluer. Dans ce cadre, c’est l’évaluateur qui choisit la combinaison d’outils qui lui permet d’évaluer au mieux une implémentation – c’est également à lui qu’il incombe de justifier pourquoi cette combinaison est pertinente. Pour ce faire, il peut être utile d’énumérer les grandes phases d’une analyse experte : (a) mesure et « preprocessing », (b) détection des points d’intérêt, (c) exploitation. La phase (a) est commune à toutes les attaques : son objectif est d’obtenir des mesures les moins bruitées possibles. La phase (b) est une extension de l’évaluation par détection dans laquelle l’évaluateur cherche en outre à localiser les parties des mesures qui sont exploitables par un attaquant et à les lier avec les opérations exécutées [DS16]. La phase (c) est la plus spécifique à ce type d’analyse et se divise généralement en une tâche de modélisation des mesures (optionnelle : elle n’est possible que dans le cadre d’évaluations profilées), une tâche d’extraction de l’information et une tâche de traitement de l’information. Un exemple standard d’exploitation serait (une fois la phase de détection des points d’intérêt conclue avec succès) de modéliser les mesures avec des « templates » Gaussiens [CRR02] ou une régression linéaire [SLP05], d’extraire de l’information par maximum de vraisemblance et de la traiter via une approche « divide-and-conquer », éventuellement complétée par une phase d’énumération [PSG16]. L’analyse peut ensuite être résumée par un « graphe de sécurité » (dont un exemple est donné en Figure 2) évaluant la probabilité de succès de l’attaque en fonction du nombre de mesures et du temps de calcul. De nombreuses variantes ont été introduites dans la littérature afin d’exploiter efficacement l’information contenue dans des distributions statistiques multivariées et d’ordre élevé, telles que communément requises pour l’évaluation d’implémentations protégées.

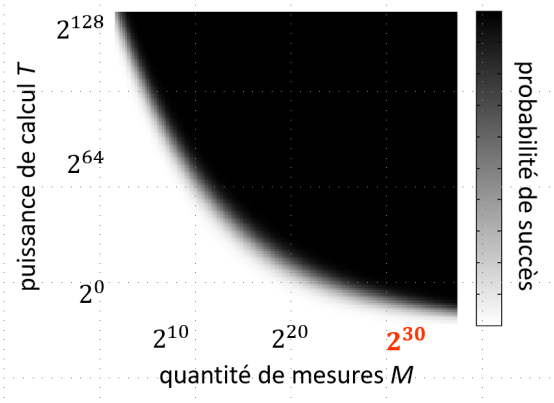

Figure 2. Exemple de graphe de sécurité.

Enfin, l'évaluation par preuve laisse la même liberté dans le choix des outils et métriques d'analyse, mais demande un plus grand niveau de formalisme : l'extrapolation de l'analyse d'une implémentation « à sécurité réduite » se basant typiquement sur les travaux théoriques d'analyse des contremesures. Un exemple éclairant est celui de la contremesure par masquage [C+99]. Elle a pour but de modifier l'implémentation de telle manière que l'ensemble de ses calculs soient rendus aléatoires via un processus de partage. Concrètement, un partage simple est le partage Booléen : il nécessite de représenter toute valeur intermédiaire x de l'implémentation par t « morceaux » $x_1, x_2, \dots, x_t$ (dont $t-1$ choisis aléatoirement) tels que $x = x_1 \oplus x_2 \oplus \dots \oplus x_t$. L'adversaire est alors forcé de « combiner » l'information de ces t morceaux pour extraire de l'information sur x, une tâche significativement plus complexe que d'attaquer une implémentation non protégée (qui implique notamment l'estimation de moments statistiques d'ordre élevé).

La sécurité d'un schéma de masquage se base sur deux hypothèses fondamentales : d'une part il faut que les mesures de chaque morceau x_i soient suffisamment bruitées, d'autre part il faut qu'elles satisfassent une hypothèse d'indépendance (en gros, que chaque échantillon d'une mesure ne dépende pas de plus d'un morceau de secret). Ces hypothèses peuvent être contredites par différents défauts de conception : manque d'aléa dans les calculs masqués (ce qui contredit l'hypothèse d'indépendance) [C+13], recombinaison de morceaux de secret à cause de défauts physiques de l'implémentation (ce qui contredit également l'hypothèse d'indépendance) [MPG05], ou manque de bruit dans les mesures. L'évaluation de ces hypothèses est donc cruciale pour obtenir des garanties de sécurité. A cet égard, une avancée importante de la recherche a été de démontrer que l'évaluation d'une implémentation masquée peut bénéficier d'une *modélisation théorique travaillant à différents niveaux d'abstraction*, telle qu'illustrée en Figure 3.

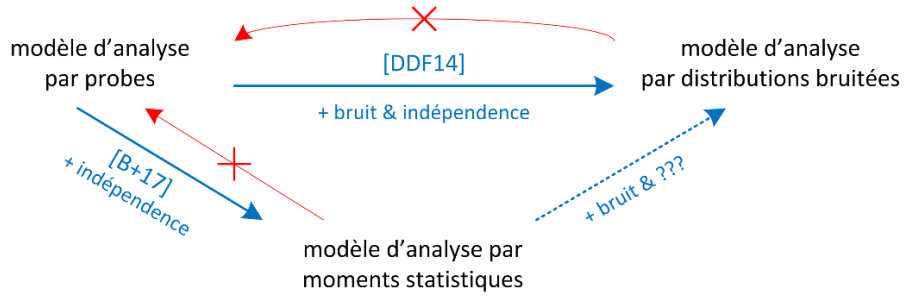

Figure 3. Modèles d'analyse de la contremesure par masquage.

Par exemple, les problèmes d'aléas peuvent être efficacement mis en évidence et résolus dans un modèle d'analyse abstrait « par probes » (où l'adversaire peut observer un nombre borné de valeurs intermédiaires du calcul sécurisé non bruitées) [ISW03,B+16]. Les problèmes de recombinaison dus à des défauts physiques peuvent être efficacement mis en évidence dans un modèle d'analyse qualitatif « par moments statistiques » [B+17], dont l'évaluation concrète peut tirer parti des outils de détection mentionnés précédemment. Enfin, les problèmes de bruit peuvent être mis en évidence dans un modèle d'analyse quantitatif « par distributions bruitées » [PR13], dont l'évaluation concrète peut tirer parti des outils d'attaque mentionnés précédemment. Il a en outre été démontré que ces modèles sont formellement liés (comme illustré par les flèches de la figure) [DDF14], ce qui suggère que de hauts niveaux de sécurité peuvent être garantis en analysant successivement la sécurité d'une implémentation masquée dans ces trois modèles [DFS15]. Il est également possible de résoudre certains problèmes à de plus hauts niveaux d'abstraction que ceux où ils apparaissent. Par exemple, les « implémentations threshold » empêchent certains défauts physiques grâce à une propriété de « non-complétude » analysable dans un modèle par probe [NRS11].

Etape 4 : discussion critique

Globalement, la pertinence d'une évaluation de sécurité physique provient de la *justification des choix et de l'analyse des risques de surévaluation de la sécurité* - et donc des failles mises en évidence autant que de la discussion de celles qui auraient pu échapper à l'évaluateur. Par nature, les évaluations par détection sont les plus susceptibles à ce type d'erreur, vu leur caractère non spécialisé aux implémentations évaluées, et le fait qu'elles n'estiment que la présence d'information sensible, pas la possibilité de l'exploiter. Les analyses expertes par attaques et par preuve ont pour objectif de minimiser ces erreurs. Mais même dans ces cas, des risques importants, inhérents au caractère physique des attaques, subsistent. Certains, comme la nécessité d'un bon matériel de mesure, semblent inévitables (et ne peuvent probablement être contrôlés que via des comparaisons entre différents matériels de mesure). D'autres, comme la nécessité d'un bon modèle prédictif pour les mesures, peuvent être limités plus systématiquement [DSV14] (mais ce type de certification ne permet pas de déterminer le modèle optimal, uniquement de se convaincre que l'évaluateur en est suffisamment proche ou pas). L'étude des techniques d'analyse « au pire cas », exploitant toute l'information rendue disponible par une implémentation cryptographique, reste un problème ouvert et n'a à ce jour pas de réponse unique et indépendante des implémentations. Il en va de même pour l'automatisation de certaines parties de ces analyses [B+15]. Dans ce contexte, il faut à nouveau observer que la divulgation des détails de conception d'une implémentation est généralement bénéfique pour minimiser les erreurs d'évaluation (car elle permet une meilleure modélisation – et donc une meilleure compréhension des risques résiduels inhérents au problème des attaques physiques).[1] A cet égard, une autre question de recherche intéressante est de déterminer dans quelle mesure les attaques que l'on peut réaliser dans un modèle de conception ouvert (où les détails d'implémentation sont connus pas l'adversaire) sont également possibles dans un modèle fermé, par exemple en utilisant des techniques d'apprentissage automatique [MPP16]. Au final, on peut encore noter qu'à long terme, le besoin de transparence sur les détails de conception des implémentations cryptographiques gagnerait à s'étendre aux évaluations elles-mêmes : la possibilité d'évaluer les évaluateurs (en rendant leurs expertises les plus ouvertes et reproductibles possibles) permettrait en effet de rapprocher les analyses de sécurité physique de la cryptanalyse classique, où la somme des expertises conjuguées permet d'améliorer la sécurité des algorithmes.

Remerciements

François-Xavier Standaert est maître de recherche FNRS-F.R.S. Ce travail a été financé en partie par les projets ERC 724725 (SWORD), H2020 REASSURE et FEDER USERMedia (convention 501907-379156). L'auteur remercie François Durvaux, François Koeune, Liran Lerman, Philippe Teuwen et Vincent Verneuil pour leur relecture attentive de versions préliminaires de cet article et leurs commentaires utiles.

Références

[B+15] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub: Verified Proofs of Higher-Order Masking. EUROCRYPT (1) 2015: 457-485. **[B+16]** Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, Rébecca Zucchini*: Strong Non-Interference and Type-Directed Higher-Order Masking.* ACM Conference on Computer and Communications Security 2016: 116-129. **[B+17]** Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, Pierre-Yves Strub: *Parallel Implementations of Masking Schemes and the Bounded Moment Leakage Model.* EUROCRYPT (1) 2017: 535-566. **[BBK08]** Elad Barkan, Eli Biham, Nathan Keller : Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. J. Cryptology 21(3): 392-429 (2008). **[C+99]** Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, Pankaj Rohatgi: *Towards Sound Approaches to Counteract Power-Analysis Attacks.* CRYPTO 1999: 398-412. **[C+13]** Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, Thomas Roche: *Higher-Order Side Channel Security and Mask Refreshing*. FSE 2013: 410-424. **[CRR02]** Suresh Chari, Josyula R. Rao, Pankaj Rohatgi: *Template Attacks*. CHES 2002: 13-28. **[DDF14]** Alexandre Duc, Stefan Dziembowski, Sebastian Faust: *Unifying Leakage Models: From Probing Attacks to Noisy Leakage*. EUROCRYPT 2014: 423-440. **[DFS15]** Alexandre Duc, Sebastian Faust, François-Xavier Standaert: *Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device.* EUROCRYPT (1) 2015: 401-429. **[DSV14]** François Durvaux, François-Xavier Standaert, Nicolas Veyrat-Charvillon: How to Certify the Leakage of a Chip? EUROCRYPT 2014: 459-476. **[DS16]** François Durvaux, François-Xavier Standaert: *From Improved Leakage Detection to the Detection of Points of Interests in Leakage Traces*. EUROCRYPT (1) 2016: 240-262. **[G+08]** Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers,

[1] Travailler dans un modèle de conception ouvert est en outre parfois nécessaire, par exemple pour l'analyse par preuve d'implémentations masquées à très hauts niveaux de sécurité – voir par exemple [GS18].

Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, Bart Jacobs: *Dismantling MIFARE Classic*. ESORICS 2008: 97-114. **[GS18]** Vincent Grosso, François-Xavier Standaert: *Masking Proofs Are Tight and How to Exploit it in Security Evaluations.* EUROCRYPT (2) 2018: 385-412. **[ISW03]** Yuval Ishai, Amit Sahai, David A. Wagner: *Private Circuits: Securing Hardware against Probing Attacks*. CRYPTO 2003: 463-481 **[KJJ99]** Paul C. Kocher, Joshua Jaffe, Benjamin Jun: *Differential Power Analysis*. CRYPTO 1999: 388-397**. [MPG05]** Stefan Mangard, Thomas Popp, Berndt M. Gammel: *Side-Channel Leakage of Masked CMOS Gates*. CT-RSA 2005: 351-365. **[MPP16]** Houssem Maghrebi, Thibault Portigliatti, Emmanuel Prouff: *Breaking Cryptographic Implementations Using Deep Learning Techniques.* SPACE 2016: 3-26. **[NRS11]** Svetla Nikova, Vincent Rijmen, Martin Schläffer: *Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches*. J. Cryptology 24(2): 292-321 (2011). **[PSG16]** Romain Poussier, François-Xavier Standaert, Vincent Grosso: *Simple Key Enumeration (and Rank Estimation) Using Histograms: An Integrated Approach*. CHES 2016: 61-81. **[PR13]** Emmanuel Prouff, Matthieu Rivain: *Masking against Side-Channel Attacks: A Formal Security Proof*. EUROCRYPT 2013: 142-159. **[SLP05]** Werner Schindler, Kerstin Lemke, Christof Paar: *A Stochastic Model for Differential Side Channel Cryptanalysis*. CHES 2005: 30-46. **[SM16]** Tobias Schneider, Amir Moradi: *Leakage assessment methodology - Extended version.* J. Cryptographic Engineering 6(2): 85-99 (2016). **[WOS14]** Carolyn Whitnall, Elisabeth Oswald, François-Xavier Standaert: *The Myth of Generic DPA...and the Magic of Learning*. CT-RSA 2014: 183-205.