Amélioration d'attaques par canaux auxiliaires sur la cryptographie asymétrique

Margaux Dugardin

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2017-ENST-0035

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Electronique et Communications »

présentée et soutenue publiquement par

Margaux DUGARDIN

le 11 juillet 2017

Amélioration d'attaques par canaux auxiliaires sur la cryptographie asymétrique

Directeur de thèse : **Sylvain GUILLEY**, Professeur associé, TELECOM PARISTECH
Co-encadrement de la thèse : **Jean-Luc DANGER**, Directeur d'Etudes, TELECOM PARISTECH

Jury

M. Pierre-Alain FOUQUE, Professeur, UNIVERSITÉ RENNES 1	Rapporteur
M. David NACCACHE, Professeur, ÉCOLE NORMALE SUPÉRIEURE	Rapporteur
Mme Lejla BATINA, Professeur, RADBOUD UNIVERSITY	Examinateur
M. Marc JOYE, Fellow NXP, HDR , NXP	Examinateur
M. Emmanuel PROUFF, Ingénieur de recherche, HDR, ANSSI	Examinateur
M. Werner SCHINDLER, Professeur, BSI	Examinateur
M. François DASSANCE, Evaluateur sécurité, THALES COMMUNICATIONS & SECURITY	Invité
M. Rémy DAUDIGNY, Directeur du CESTI, THALES COMMUNICATIONS & SECURITY	Invité

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

THÈSE

Remerciements

Mes travaux de recherches ont été possible grâce au CESTI Thales de Toulouse et le laboratoire COMELEC de Télécom ParisTech. Mes trois années de recherches se sont déroulées entre Paris et Toulouse, avec une préférence pour la vie Toulousaine. Les déplacements réguliers entre le laboratoire de recherche de Telecom ParisTech et le laboratoire d'analyse du CESTI de Thales ont contribué à la réussite de cette thèse.

J'aimerais remercier chaleureusement toutes les personnes qui ont contribuées au bon déroulement cette thèse, notamment mon directeur de thèse , Sylvain Guilley. Ce dernier a été disponible généralement tôt le matin ou plutôt en soirée/nuit pour m'aiguiller dans mes travaux de recherches, ainsi que dans l'aide à la rédaction de mes divers travaux. Je ne peux oublier les nuits qui raccourcissaient au fur et à mesure que la date de soumission de CHES se rapprochait. Je tiens également à remercier Jean-Luc Danger qui a suivi mes recherches et l'avancement de l'écriture de ce manuscrit. Jean-Luc poussait énormément pour l'écriture et la présentation de mes travaux en conférence, ce fut un atout supplémentaire pour la motivation.

Ma thèse n'aurait pas suivi cette direction sans Jean-Christophe Courrège, mon encadrant chez Thales durant ma première année de thèse. Jean-Christophe a toujours trouvé du temps pour m'aider à avancer et je le remercie pour m'avoir ouvert les portes du laboratoire tardivement afin de me permet de persévérer dans la découverte des failles de sécurité. Je le remercie aussi d'avoir continué à suivre mes travaux de loin après son départ.

Je remercie aussi énormément François Dassance d'avoir accepté de m'encadrer (me supporter) après le départ de Jean-Christophe. François a permis que la fin de ma thèse se passe sereinement en m'écoutant parler des heures sans comprendre un mot de ce que je disais surtout lors de mes problèmes de calcul avec mes multiples intégrales. Je le remercie également d'avoir passé du temps pour me relire et su améliorer la qualité et la compréhension de mes travaux.

Je remercie Pierre-Alain Fouque et David Naccache d'avoir accepté de rapporter cette thèse. Ils ont contribué à l'amélioration de la rédaction de ce manuscrit. Je remercie également les autres membres de mon jury, qui ont accepté de faire le déplacement alors que certain était en vacances. Merci Pierre-Alain Fouque et Emmanuel Prouff d'avoir aussi été présent lors de ma soutenance de mi-thèse. Vos conseils et encouragements m'ont aidé à avancer et que la fin de cette thèse se déroule avec succès. I wish to thank Lejla Batina and Werner Schindler for your attendance during the Phd defense and for the researching collaboration. Merci à Marc Joye pour ses questions très intéressantes, j'ai apprécié les remarques qui ont amélioré la présentation de mes résultats le jour J.

J'aimerais remercier toutes les personnes qui ont participé à la vie de COMELEC durant ces 3 années. Un merci particulier à Zakaria Najm d'avoir mené des campagnes

d'acquisitions et cherché avec moi pendant 6 mois les motifs de multiplications. Nos sessions de brainstorming sur la provenance des fuites dans les multiplications me reviennent encore. Merci de m'avoir appris à utiliser un oscilloscope, un ordinateur sous linux en ligne de commande, les bases du langage Python, etc. Merci à Olivier Rioul pour l'aide sur les mathématiques. Merci à Guillaume Duc et Laurent Sauvage pour leurs multiples explications sur les attaques par canaux auxiliaires et fautes. Merci à Pablo Rauzy et Martin Moreau pour leurs contributions et travaux sur la contre-mesure sur les attaques par faute. Merci à Michaël Timbert d'avoir été une oreille attentive et un compagnon de sevrage de café. J'ai beaucoup aimé ta vision informatique des choses confrontée à ma vision mathématiques, quand tu voyais un XOR, je voyais une addition modulaire par deux. Je me demande encore comment faisions-nous pour nous comprendre. Merci à Nicolas Bruneau de m'avoir dit que j'avais une cape de super-héros et que rien ne pouvait m'atteindre. Je le remercie aussi car il était de très bon conseil pour la rédaction et la préparation de la soutenance de thèse, qu'il venait lui même de vivre. Merci aux personnes travaillant à Secure-IC Paris de m'avoir supporté le midi alors que je voulais manger uniquement à la crêperie.

I would like to thank Louiza Papachristodoulou. I liked working with her. The collaboration was funny, because I spoke English-French and German in the same sentence to make me understand.

Je tiens à remercier Alain Wislez, Nathalie Feyt, Rémy Daudigny, Rodolphe Favrel, Jean-Christophe Courrège d'avoir permis que ma thèse se déroule au sein du CESTI dans de bonnes conditions. Merci à toute l'équipe du CESTI pour leur accueil chaleureux, si je commence à lister les personnes, j'ai peur d'en oublier. J'ai apprécié les moments à la machine à café, mais aussi les midis à la piscine, les quelques After-Work, sortie canoë, soirée jeux et journées de mariage passés ensemble. Merci pour toute votre gentillesse témoignée lors de mon départ.

La vie au CESTI n'aurait pas été la même sans les filles: Anne-Sophie et Carine, mais aussi Oriane, Vanessa, Marion et Eve. Un merci particulier à Anne-Sophie et Carine de m'avoir encadrée bénévolement et poussée à bien rédiger pour que cela soit compréhensible du plus grand nombre. Merci d'avoir su apporter un regard critique sur mes présentations pour qu'elles s'améliorent.

Ma thèse a commencé grâce à la persévérance de mon maître de stage Renaud Dubois, je le remercie énormément. Je tiens aussi à remercier Aurore Guillevic qui m'a fait apprécier les mathématiques sur courbes elliptiques et Sonia Belaid qui m'a expliqué succinctement les attaques par canaux auxiliaires, elle m'a donné envie de me lancer dans cette cryptanalyse différente. Je ne peux oublier aussi les personnes de mon Master qui ont été présentes durant cette thèse, comme Julia Chaulet et Romain Poussier.

Je tiens à remercier ma famille et mes amis qui ne m'ont pas beaucoup vu durant ses 3 ans. Merci à eux pour leur soutien. Ils m'ont poussé à ne jamais abandonner. Un grand merci à Julia, Lucas, Eric et Céline de m'avoir hébergé au début de mon aventure parisienne. Je remercie aussi Jérémy qui a accepté la distance et les allers-retours hebdomadaires Paris-Toulouse. Merci à lui d'avoir organisé tous nos week-ends, il me suffit d'une main pour compter le nombre de week-end durant ces 3 ans où je n'ai pas dû aller dans une gare ou un aéroport.

Résumé

Depuis les années 90, les attaques par canaux auxiliaires ont remis en cause le niveau de sécurité des algorithmes cryptographiques sur des composants embarqués. En effet, tout composant électronique produit des émanations physiques, telles que le rayonnement électromagnétique, la consommation de courant ou encore le temps d'exécution du calcul. Or il se trouve que ces émanations portent de l'information sur l'évolution de l'état interne. On parle donc de canal auxiliaire, car celui-ci permet à un attaquant avisé de retrouver des secrets cachés dans le composant par l'analyse de la « fuite » involontaire.

Cette thèse présente d'une part deux nouvelles attaques ciblant la multiplication modulaire permettant d'attaquer des algorithmes cryptographiques protégés et d'autre part une démonstration formelle du niveau de sécurité d'une contre-mesure. La première attaque vise la multiplication scalaire sur les courbes elliptiques implémentée de façon régulière avec un masquage du scalaire. Cette attaque utilise une unique acquisition sur le composant visé et quelques acquisitions sur un composant similaire pour retrouver le scalaire entier. Une fuite horizontale durant la multiplication de grands nombres a été découverte et permet la détection et la correction d'erreurs afin de retrouver tous les bits du scalaire. La seconde attaque exploite une fuite due à la soustraction conditionnelle finale dans la multiplication modulaire de Montgomery. Une étude statistique de ces soustractions permet de remonter à l'enchainement des multiplications ce qui met en échec un algorithme régulier dont les données d'entrée sont inconnues et masquées. Pour finir, nous avons prouvé formellement le niveau de sécurité de la contre-mesure contre les attaques par fautes du premier ordre nommée extension modulaire appliquée aux courbes elliptiques.

Abstract

Since the 1990s, side channel attacks have challenged the security level of cryptographic algorithms on embedded devices. Indeed, each electronic component produces physical emanations, such as the electromagnetic radiation, the power consumption or the execution time. Besides, these emanations reveal some information on the internal state of the computation. A wise attacker can retrieve secret data in the embedded device using the analyzes of the involuntary "leakage", that is side channel attacks. This thesis focuses on the security evaluation of asymmetric cryptographic algorithm such as RSA and ECC. In these algorithms, the main leakages are observed on the modular multiplication.

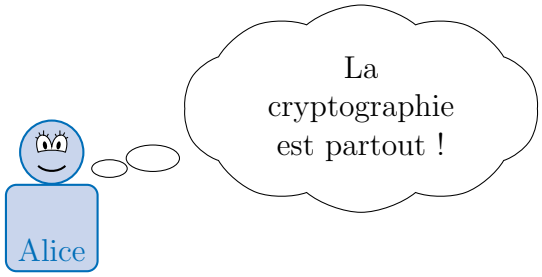
This thesis presents two attacks targeting the modular multiplication in protected algorithms, and a formal demonstration of security level of a countermeasure named modular extension. A first attack is against scalar multiplication on elliptic curve implemented with a regular algorithm and scalar blinding. This attack uses a unique acquisition on the targeted device and few acquisitions on another similar device to retrieve the whole scalar. A horizontal leakage during the modular multiplication over large numbers allows to detect and correct easily an error bit in the scalar. A second attack exploits the final subtraction at the end of Montgomery modular multiplication. By studying the dependency of consecutive multiplications, we can exploit the information of presence or absence of final subtraction in order to defeat two protections: regular algorithm and blinding input values. Finally, we prove formally the security level of modular extension against first order fault attacks applied on elliptic curves cryptography.

Introduction générale en français

Alice aux pays de la cryptographie

Laissez-moi vous raconter une journée du quotidien d'Alice. Pour aller travailler, Alice prend les transports en commun à l'aide son passe électronique. En arrivant au boulot, elle ouvre la porte de son bureau avec son badge d'identification. Après avoir dit bonjour à ses collègues, elle s'installe à son bureau et s'identifie/authentifie sur la session de son ordinateur pour avoir accès au réseau de son entreprise. Ainsi, elle prend connaissance de ses courriels chiffrés pour en garantir la confidentialité. Au midi, elle envoie un message à Bob, afin qu'il la rejoigne pour manger. Elle prend son plateau et paye à la caisse à l'aide de son badge de cantine. A la fin de sa journée de boulot, elle décide d'aller à la piscine en utilisant sa carte de loisir contenant 20h de natation. Avant de rentrer, elle passe chercher ses médicaments en utilisant sa carte vitale à la pharmacie. En rentrant, elle prend du pain à la boulangerie en utilisant le paiement sans contact de sa carte bancaire. Arrivée à la maison, elle décide d'acheter les billets de train du prochain week-end en ligne en utilisant une carte virtuelle générée sur le site de sa banque. Ensuite, elle s'installe sur le canapé pour regarder un film à la demande sur sa TV connectée en mangeant son repas. A la fin de la journée, Alice se demande : où et quand la cryptographie a été présente dans sa journée ?

La réponse est dans toutes les actions quotidiennes décrites précédemment. D'ailleurs dans la plupart des actions Alice utilise de la cryptographie embarquée. En effet, quand Alice a utilisé une carte telle que son passe de transport, son badge d'entreprise, etc. Ses cartes ont permis l'authentification et l'identification d'Alice ou sont une preuve d'achat par Alice. La transaction d'e-commerce effectuée possède aussi une authentification d'Alice ainsi que la confidentialité et la non-répudiation de l'achat. Quand Alice reçoit des courriels ou envoie un message à Bob, alors les messages peuvent être chiffrés et déchiffrés pour assurer la confidentialité et l'intégrité des messages. Dans la majorité des cas, les vidéos à la demande sont chiffrées en utilisant du chiffrement d'émission. Alice se dit que la cryptographie a une place importante dans son quotidien.



Contexte

Dans un monde où l'information se dématérialise de plus en plus, la sécurité informatique a pris une place importante dans la vie de tous les jours. La cryptographie a pour but d'apporter les garanties les plus fortes possibles sur la sécurité de l'information. Pour cela, les algorithmes cryptographiques sont dits « sûrs » quand ils sont démontrés par des preuves mathématiques. La plus ancienne cryptographie est la cryptographie symétrique ou à clé secrète. Elle permet de chiffrer et déchiffrer un message à l'aide d'une clé commune partagée entre les personnes voulant communiquer. Son principal avantage est qu'elle est très efficace, mais elle nécessite l'échange au préalable d'un secret commun. A l'opposé, la cryptographie asymétrique datant du milieu du 20è siècle, permet un échange sécurisé sans partage de secret commun, mais les calculs nécessitent plus de mémoire et de temps de calcul. La recherche effectuée pour cette thèse se concentre uniquement sur la cryptographie asymétrique, en particulier le cryptosystème basé sur RSA et la cryptographie sur les courbes elliptiques. Cependant la mise en œuvre de ces algorithmes sur un support physique, tels que ordinateur, smartphone, carte à puce, provoque une vulnérabilité en terme de sécurité. En effet, un support matériel possède des propriétés physiques. Il consomme du courant électrique. Il produit des émissions électromagnétiques ou photoniques. La durée de l'exécution d'un calcul sur le support matériel peut être différente. Chacun de ses facteurs non cités de manière exhaustive permet un certain type d'attaque remettant en cause l'évaluation du niveau de sécurité de la cryptographie. Ces dernières font partie de la cryptanalyse et se nomment « attaques par canaux auxiliaires » (SCA de l'anglais Side-Channel Attacks). Ces attaques utilisent l'information que l'on peut déduire de la consommation de courant, du temps d'exécution, de l'émanation électromagnétique du circuit, etc. Un attaquant avisé peut retrouver des secrets cachés dans le composant par l'analyse de la « fuite » involontaire. Plus précisément, il existe deux façons d'extraire de l'information : soit par une analyse de la fuite observée, c'est une attaque dite passive; soit par une injection de faute lors d'un calcul, c'est une attaque dite active. L'attaque passive est une observation d'un ou plusieurs canaux auxiliaires durant le calcul et ne modifie pas l'exécution de celui-ci. L'attaque active est une perturbation volontaire du bon déroulement de l'exécution du calcul en ajoutant une faute ciblée. Les premières preuves de concept d'attaques datent d'une vingtaine années. Depuis, nous assistons à une explosion du nombre d'attaques potentielles, plus ou moins dépendantes des choix d'implémentation et/ou des contremesures algorithmiques mises en œuvre. La principale conséquence est que l'évaluation de la sécurité devient de plus en plus complexe.

Principe général de la cryptographie asymétrique

En 1976, W.Diffie et M. Hellman [DH76] propose une nouvelle façon de communications sécurisés entre deux participants (par exemple Alice et Bob). Alice veut envoyer un message $\mathcal{M}$ à Bob, mais le canal de communication utilisé peut être écouté. Alice et Bob doivent donc partager un secret en commun afin de pouvoir chiffrer et déchiffrer leur conversation. La cryptographie symétrique (à clé secrète) étant plus rapide que la cryptographie asymétrique. Alice et Bob vont utiliser le protocole de Diffie-Hellman présenté sur la figure 1 pour s'échanger un secret commun. Ensuite ils utiliseront la cryptographie symétrique en utilisant ce secret commun pour communiquer. Alice et

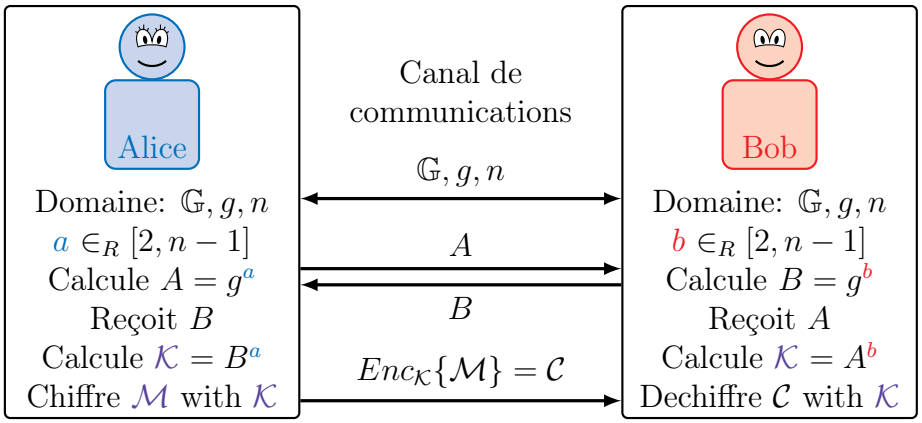


Figure 1: Protocole standard pour chiffrer un message avec un secret commun partagé par Diffie-Hellman

Bob se mettent d'accord sur le domaine cryptographique sur lequel ils veulent travailler, composé d'un groupe $\mathbb{G}$ avec g comme générateur d'un sous-groupe de $\mathbb{G}$ et n le nombre d'éléments du sous-groupe. Alice choisit un nombre aléatoire a entre $[2, n-1]$, calcule $g^a = A$ dans le groupe $\mathbb{G}$ et envoie la valeur A à Bob. Bob choisit un nombre aléatoire b entre $[2, n-1]$, calcule $g^b = B$ dans le groupe $\mathbb{G}$ et envoie la valeur B à Alice. Alice et Bob calcule leur secret commun $\mathcal{K}$ en utilisant B et a pour Alice et en utilisant A et b pour Bob. Alice peut donc envoyer son message chiffré avec le secret commun $\mathcal{K}$ appelé en anglais ciphertext. Bob déchiffre le texte chiffré en utilisant le secret commun afin de retrouver le message clair d'Alice. Un attaquant ayant observé tous les échanges entre Alice et Bob possède le domaine cryptographique avec le générateur g et les deux éléments du sous-groupe $A = g^a$ et $B = g^b$. Avec uniquement ses informations, il est impossible pour un attaquant de trouver le secret commun $\mathcal{K}$. Ce problème s'appelle le problème du logarithme discret, qui est un problème difficile à résoudre en temps raisonnable, lorsque les paramètres assurant un niveau de sécurité sont choisis rigoureusement.

Les principaux objectifs de la cryptographie sont la confidentialité et l'intégrité des messages, ainsi que l'authentification d'un utilisateur et la signature d'une donnée. La cryptographie asymétrique est aussi appelée la cryptographie à clé publique. Chaque utilisateur génère un couple de clé privée/publique. La clé publique est diffusée à l'ensemble des utilisateurs tandis que la clé secrète est conservée par son propriétaire. La clé publique sert à chiffrer un message ou vérifier l'authenticité d'une signature. La clé secrète sert à déchiffrer un message ou signer un message. Il existe plusieurs cryptosystèmes asymétriques dont deux qui sont très utilisés et connus :

1. Le cryptosystème basé sur la factorisation du produits de grands nombres premiers est nommé RSA, du nom de ses trois inventeurs : Ronald Rivest, Adi Shamir and Leonard Adleman. RSA datent du brevet publié par le MIT (Massachusetts Institute Of Technology) en 1983 [RSA83]. Il est très utilisé dans le domaine du commerce électronique et plus généralement dans l'échange de donnée de manière confidentielle sur internet.

2. La cryptographie basée sur les courbes elliptiques notée ECC (de l'anglais Elliptic

Curve Cryptography) est le deuxième cryptosystème le plus utilisé. Les courbes elliptiques ont d'abord été introduites par H.W. Lenstra dans la cryptologie pour casser la factorisation du produit de grands nombres premiers. Ensuite Miller [Mil85] et Koblitz [Kob87] ont décidé d'utiliser le groupe formé sur une courbe elliptique comme domaine cryptographique.

Que ce soit RSA ou la cryptographie sur les courbes elliptiques, ces cryptosystèmes reposent sur l'implémentation de l'arithmétique modulaire. Mes recherches se sont spécialement intéressées aux vulnérabilités exploitables lors de la mise en œuvre logicielle de la multiplication modulaire. Le chapitre 2 détaille les connaissances mathématiques nécessaires pour la compréhension de la thèse et plus précisément, l'implémentation de la multiplication modulaire est détaillée.

L'ensemble des recherches de cette thèse concerne l'évaluation de la sécurité d'algorithmes cryptographiques généralement implémentés dans un circuit embarqué, en particulier, la sécurité de la cryptographie asymétrique contre les attaques par canaux auxiliaires. Les principaux canaux auxiliaires utilisés sont le temps d'exécution, la consommation de courant et l'émanation électromagnétique. La consommation de courant et l'émanation électromagnétique nécessitent une sonde et un oscilloscope pour être mesuré. Ces mesures s'appellent des acquisitions. Le temps d'exécution d'un algorithme asymétrique est élevé pour assurer un certain niveau de sécurité, cela entraîne un coût supplémentaire sur la taille mémoire des acquisitions par rapport à des acquisitions en cryptographie symétrique. La qualité d'une acquisition dépend du taux d'échantillonage sur l'oscilloscope, de la position et de la qualité de la sonde. Les trois paramètres peuvent être optimisés pour améliorer la qualité de l'information contenue dans le signal. L'augmentation du taux d'échantillonage de l'oscilloscope augmente aussi la mémoire nécessaire des acquisitions. L'augmentation du nombre d'acquisitions peut améliorer l'analyse, en réduisant le bruit du signal et en augmentant l'information contenue dans le signal. Le nombre d'acquisitions doit être réduit pour limiter le temps global de l'attaque. En effet, l'utilisation d'un grand nombre d'acquisitions de grand taille augmente le temps d'acquisition, le pré-traitement (alignement, filtrage, etc.) et l'analyse de celles-ci, ainsi que la puissance de calcul nécessaire. Dans le contexte industriel, l'évaluation d'un circuit embarqué prend en compte le temps et le matériel nécessaire au bon déroulement des attaques.

Introduction aux attaques par canaux auxiliaires

Les attaques par canaux auxiliaires se divisent en deux parties, les attaques par observation présentées en premier et les attaques par perturbation volontaire présentées dans un second temps. Dans le chapitre 3, les différentes attaques sont présentées.

En 1999, Paul C. Kocher, Joshua Jaffe et Benjamin Jun [KJJ99] proposent une nouvelle cryptanalyse à base de canaux auxiliaires applicable sur la cryptographie symétrique et asymétrique. Un canal auxiliaire est une fuite d'information durant l'exécution d'un calcul, comme par exemple l'analyse de la consommation de courant, le rayonnement électromagnétique, le temps de calcul global, etc. En combinant l'un ou plusieurs d'entre eux il est possible de retrouver les données secrètes manipulées. Dans le domaine des attaques par canaux auxiliaires, différentes méthodes d'analyse existent. Les attaques simples par analyse de canaux auxiliaires (SSCA de l'anglais Simple Side Channel Analysis) nécessi-

tent l'acquisition d'une unique trace durant l'exécution du calcul ciblé. Pour se protéger contre les attaques dites simples, il est possible de rendre le temps de calcul régulier avec des algorithmes effectuant les mêmes opérations afin de rendre indépendant le calcul des valeurs secrètes. Une autre solution est d'utiliser des opérations atomiques, celles-ci sont indifférentiable en une trace. Cette méthode est associée à l'unification des opérations basiques. Pour le RSA, il faut que l'opération de carré et de multiplication sont la même. Pour les courbes elliptiques, il faut que le doublement et l'addition soient unifiés.

Les attaques différentielles (DSCA de l'anglais Differential Side Channel Analysis) exploitent la dépendance entre les données manipulées et l'acquisition du canal auxiliaire (la consommation de courant, le rayonnement électromagnétique). Nous pouvons classer ces attaques en deux types dépendant de l'attaquant: les attaques non supervisées et les attaques par modèle (de l'anglais "Template Attacks"). Les attaques non supervisées font partie des attaques verticales, elles utilisent un modèle de fuite ainsi qu'un distingueur. La différence de la moyenne de différents groupes est la première méthode présentée par Kocher [KJJ99] Cette attaque nécessite en général un grand nombre de traces ($\simeq 10^4$ ou $\simeq 10^6$). Une amélioration de cette technique s'appelle l'analyse par corrélation (CSCA de l'anglais Correlation Side Channel Analysis) présentée dans [BCO04]. La CSCA permet d'éviter les « faux pics » émis lors de la différence des moyennes, et donc de réduire le nombre de traces. Les attaques par modèle sont des méthodes théoriques très efficaces, car elles modélisent le bruit émis par un composant. L'attaquant doit avoir à sa disposition, en plus du composant attaqué, un composant paramétrable, c'est-à-dire un produit pour lequel nous pouvons choisir les messages d'entrée et les clés utilisées. Rechberger et Oswald [RO04] présentent la première attaque Template sur l'exécution d'un RC4 sur un micro-contrôleur 8-bits. Le travail de De Mulder et al. [MBO+05] montre la première attaque Template sur une implémentation de courbes elliptiques sur un FPGA. Les auteurs sont capables de retrouver un bit de clé en utilisant 1000 traces de rayonnement électromagnétique et la corrélation de Pearson comme distingueur. Pour se protéger contre ces attaques, différentes contre-mesures existent. Les données sont masquées à l'aide de nombres aléatoires. La séparation en groupes ayant les mêmes valeurs manipulées est alors rendu plus difficile.

Nous allons nous intéresser plus particulièrement aux attaques sur la cryptographie asymétrique. Dans [CFR10], Courrège, Feix et Rousselet proposent une attaque en une seule trace sur le RSA à message choisi. Leur analyse se base sur une fuite de consommation de courant, des multiplieurs manipulant des mots de poids de Hamming nul. Une protection est de rendre aléatoire le message afin d'éviter la fuite. Fouque et Valette [FV03] présente la première attaque par collision sur le doublement dans la cryptographie sur courbes elliptiques. La collision s'effectue lorsque le composant manipule les mêmes valeurs intermédiaires lors des calculs sur un point P, puis sur le point $2P$. Cette collision est difficile à obtenir avec des calculs intermédiaires dans une forme différente que celle des entrées.

Des attaques théoriques intéressantes pour l'étude de la cryptographie asymétrique appelées *attaques horizontales* existent [CFG+12, BJPW13, BJP+15]. Ces attaques horizontales permettent en très peu voire une seule acquisition de retrouver le secret recherché. Elles sont très efficaces sur un composant où les contre-mesures sur le scalaire et sur l'entrée sont effectuées, mais difficile à réaliser. En 1996, Kocher [Koc96] propose la première attaque utilisant le temps global d'un calcul d'exponentiation modulaire du

RSA. Ces attaques sont appelées des Timing Attacks. Brumley et Tuveri [BT11] propose une Timing Attack sur l'algorithme Montgomery Ladder implémenté dans OpenSSL. Les divers travaux de Schindler [SW03, Sch15] combinent des attaques de consommation de courant et Timing Attack pour retrouver les secrets dans la cryptographie sur RSA. Les fuites exploitées sont centrées sur l'extra-réduction présente dans la réduction modulaire sous forme de Montgomery. Nous avons étudié plus en détails ces fuites et la manière de les exploiter avec différentes protection mise en place.

Les attaques par perturbation volontaire sont aussi nommées attaques par injection de faute. Comme les attaques par observation, il en existe plusieurs types. Les attaques par "safe-error" consistent en l'injection de faute sur un calcul fantôme. Les calculs fantômes sont introduits en général pour rendre régulier un algorithme en utilisant des opérations inutiles dépend de la valeur secrète. Si une faute est injectée sur un calcul inutile alors le résultat de la sortie ne sera pas altéré, tandis que si le calcul était utile. Le résultat sera modifié. Avec une série de faute ciblée, l'attaquant peut ainsi retrouver la valeur secrète. D'autres attaques par faute permettent d'altérer les entrées nécessaires au calcul. Elles permettent d'extraire de l'information qui pourraient être exploitable. Pour finir les attaques les plus puissantes sont les attaques par analyse différentielle entre plusieurs injection de faute et le déroulement normal. Les principales protections sont calculatoires ou algorithmiques. Shamir propose d'utiliser une protection appelée l'extension modulaire. L'idée est de faire deux calculs et de les comparer. Le premier calcul est sur une extension du corps fini par un nombre aléatoire "petit" et le second est défini par le "petit" module. En réduisant le résultat du premier calcul par le petit module, une égalité entre les deux calculs est possible si aucune faute a été injectée durant le calcul. L'application de cette contre-mesure directe aux courbes elliptiques n'est pas forcément trivial (voir chapitre 7).

Travaux réalisés

Mes recherches ont mis l'accent sur les attaques par canaux auxiliaires contre la cryptographie asymétrique spécifiquement RSA et ECC. Le cryptosystème RSA repose sur l'exponentiation modulaire tandis que ECC sur la multiplication scalaire sur les courbes elliptiques. Ces opérations implémentées sans protection sont vulnérables aux attaques utilisant une « observation simple » du temps de l'exécution. Les protections classiques sur RSA et ECC sont similaires, car la mise en œuvre de l'algorithme d'exponentiation modulaire et l'algorithme de multiplication scalaire sur les courbes elliptiques présentent plusieurs similitudes. Historiquement, les premières protections ont été développées pour RSA. Par la suite, elles furent adaptées à ECC.

Dans ce contexte, nous montrons que les attaques par canaux auxiliaires avec un nombre limité d'acquisitions peuvent être suffisante pour casser les protections classiques. La principale fuite exploitée dans cette thèse est la différence de temps d'exécution dans l'arithmétique modulaire, spécialement dans la multiplication modulaire. En outre, l'une des protections contre l'exploitation d'attaques sur la multiplication modulaire est « l'extension modulaire ». Cette protection est généralement utilisée contre les attaques par injection de fautes sur RSA. L'adaptation de cette protection pour ECC n'est pas triviale et certaines publications sont incorrectes. En effet, le niveau de sécurité de la protection n'a pas été évalué formellement et cette protection est inefficace dans beaucoup de cas.

Dans un premier temps, mes travaux ont donné lieu à une attaque horizontale sur les courbes elliptiques publiée à COSADE 2016. Cette attaque correspond à une attaque par modèle (en anglais Template attack) sur un algorithme de multiplication scalaire sur les courbes elliptiques. L'algorithme est rendu régulier pour protéger contre les attaques par « simple observation » et le scalaire est aléatoire dans le cadre de ECDSA ou protégé par masquage. L'attaque utilise une unique acquisition sur le circuit attaqué et un nombre limité d'acquisitions pour la réalisation finale de l'attaque. Une condition nécessaire au bon déroulement de l'attaque est la connaissance du point d'entrée. Nous avons exploité la découverte d'une fuite temporelle de l'implémentation d'une multiplication de deux grands nombres d'une bibliothèque de calculs.

Dans un second temps, j'ai étudié une autre méthode de multiplication modulaire : la multiplication de Montgomery, beaucoup utilisée dans la cryptographie sur RSA. Cette deuxième attaque met en défaut la régularité de l'algorithme comme la première ainsi que la protection concernant le masquage et la non connaissance de l'entrée de l'algorithme. Cette nouvelle attaque exploite une dépendance dans l'exécution consécutive d'opération. Elle a été présentée à CHES 2016, nécessitant toujours d'un nombre d'acquisitions limités. Avec les retours de la conférence CHES, une amélioration de l'attaque a été trouvée, réduisant par deux le nombre d'acquisition nécessaire et sera prochainement publiée dans

un journal.

Pour finir, j'ai participé à la correction de la contre-mesure « extension modulaire » contre les attaques par faute sur les courbes elliptiques. Les attaques par fautes sont beaucoup utilisées dans la cryptographie asymétrique, car elles nécessitent généralement d'un nombre limité d'exécution pour réussir. La correction de la contre-mesure et la preuve formelle de sécurité ont été publiées à PROOFS 2016.

Attaque par modèle en ligne

Les attaques par modèle (de l'anglais « Template Attack ») s'effectuent généralement en deux phases : une phase hors-ligne de construction d'acquisition qui forment des traces de modèles (building phase) et une phase en ligne de correspondance (matching phase). La principale nouveauté de cette attaque est que la construction des modèles s'effectue après avoir réalisé l'acquisition sur le produit ciblé nécessaire pour la phase de correspondance. Cette attaque notée OTA (de l'anglais Online-Template Attack) a été présentée à Indocrypt en 2014 par Batina et al. [BCP+14]. Le papier présenté à COSADE 2016 [DPN+16] est le résultat d'une collaboration entre Telecom ParisTech et l'université Radboud de Nijmegen dans le cadre de COST actions (http://www.cost.eu/COST_Actions).

Dans le chapitre 4, nous présentons l'attaque par modèle en ligne, qui est une attaque contre la multiplication scalaire sur les courbes elliptiques. Cette attaque repose sur la collision de données manipulées lors de l'opération du doublement de points sur les courbes elliptiques. L'algorithme de multiplication scalaire sur les courbes elliptiques attaqué est régulier, c'est-à -dire qu'il n'y aucune différence d'implémentation entre le traitement d'un bit de scalaire égale à « 1 » par rapport à un bit de scalaire qui vaudrait « 0 ». Le scalaire utilisé lors de la multiplication scalaire peut être masqué par différentes techniques ou le rendre aléatoire comme dans le protocole de signature ECDSA . Une unique acquisition est nécessaire afin de retrouver le scalaire de celle-ci. Pour les acquisitions des traces de modèle uniquement le premier doublement est nécessaire. L'utilisation d'un autre circuit où l'on maîtrise totalement le contrôle n'est pas nécessaire.

Les trois principales conditions nécessaires de l'attaque sont :

- L'attaquant doit connaître le point d'entrée de la multiplication scalaire ;
- Il doit connaître l'implémentation de la multiplication et être capable de calculer les valeurs intermédiaires de chaque itération en émettant une hypothèse sur le bit du scalaire attaqué ;
- Il peut choisir les points d'entrée de la multiplication scalaire sur une circuit ressemblant au circuit attaqué.

Pour cette attaque, les bits sont retrouvés un par un. Nous pouvons les trouver dans l'ordre de la lecture du scalaire par l'algorithme (ici de gauche à droite — du bit de poids fort au bit de poids faible). Le déroulement de l'attaque OTA est le suivant :

1. En premier, l'attaquant obtient une acquisition de l'exécution de la multiplication scalaire avec le point d'entrée P sur le composant dont il veut connaître le scalaire, cette acquisition est la « trace ciblée ».

2. Il affecte à la valeur m_l la valeur de 1, car par définition d'un bit de poids fort il est égal à 1.

3. Pour chaque de bits de scalaire k_i en commençant par i qui vaut l, nous avons deux hypothèses « 0 » ou « 1 »

   (a) L'attaquant calcule les deux hypothèses de points intermédiaires $Q_0 = [2m_i]P$ et $Q_1 = [2m_i + 1]P$ en utilisant la loi de groupe utilisée dans le composant ciblée.

   (b) L'attaquant fait les deux acquisitions de la multiplication scalaire avec les deux points d'entrée $Q_0 = [2m_i]P$ et $Q_1 = [2m_i + 1]P$.

   (c) L'attaquant récupère le premier doublement de l'acquisition avec le point d'entrée $Q_0 = [2m_i]P$ nommé $T0$.

   (d) L'attaquant récupère le premier doublement de l'acquisition avec le point d'entrée $Q_1 = [2m_i + 1]P$ nommé $T1$.

   (e) L'attaquant découpe le doublement suivant le bit attaqué de la « trace ciblée », il est noté $ECDBL_{i-1}$.

   (f) L'attaquant compare $ECDBL_{i-1}$ avec $T0$ puis $T1$, la meilleure comparaison permet de retrouver le bit du scalaire k_i.

   (g) L'attaquant retourne à l'étape 3a afin de trouver le bit suivant k_{i-1} avec m_{i-1} qui vaut $2m_i + k_i$.

A la fin de cette procédure, l'attaquant à retrouver tous les bits un à un excepté le bit de poids faible, qu'il peut déterminer à l'aide de la sortie de l'algorithme en essayant les deux hypothèses.

La principale différence entre notre amélioration est l'attaque originale est que nous décidons d'acquérir une trace par hypothèse de bit de scalaire. Dans l'attaque originale, les auteurs ont décidé d'établir un seuil afin de déterminer si la trace de modèle ressemble suffisamment à la « trace ciblée ». Premièrement, ce seuil est très difficile à obtenir lors de l'acquisition de traces qui ont beaucoup de bruit. Le bruit lors d'une prise de mesure peut être dû aux matériels de mesure, mais aussi à d'autres composants ayant une autre activité que le calcul ciblé, le bruit de mesure peut-être particulièrement handicapant lors de l'acquisition de traces de rayonnement électromagnétique. La partie expérimentale de l'attaque a été menée sur l'implémentation logicielle des courbes standardisées par le NIST (`P-256`) et la BSI (`brainpoolP256r1`) sur une micro-contrôleur ARM : Cortex M4, manipulant de registre de 32 bits. Nous avons utilisé la bibliothèque logicielle PolarSSL v1.3.7 pour la loi de groupe des courbes elliptiques et de l'arithmétique modulaire. Lors de la partie expérimentale, nous avons observé une fuite horizontale involontaire due à une propagation de retenue interne lors de la multiplication de grands nombres de taille 256 bits chacun faites en 8 multiplications de 32 par 256 bits. Cette fuite temporelle existe toujours dans la nouvelle version de PolarSSL nommée mbedTLS v2.2.0, mais aussi dans OpenSSL v1.0.2. Pour détecter la fuite horizontale, il suffit d'aligner/de synchroniser le début de deux multiplications, comme le montre la figure 2, nous observons un dés-alignement de la fin des multiplications.
Cette fuite horizontale permet de détecter facilement si la trace de modèle pour une hypothèse de bit du scalaire correspond à la « trace ciblée », ou si c'est l'autre hypothèse

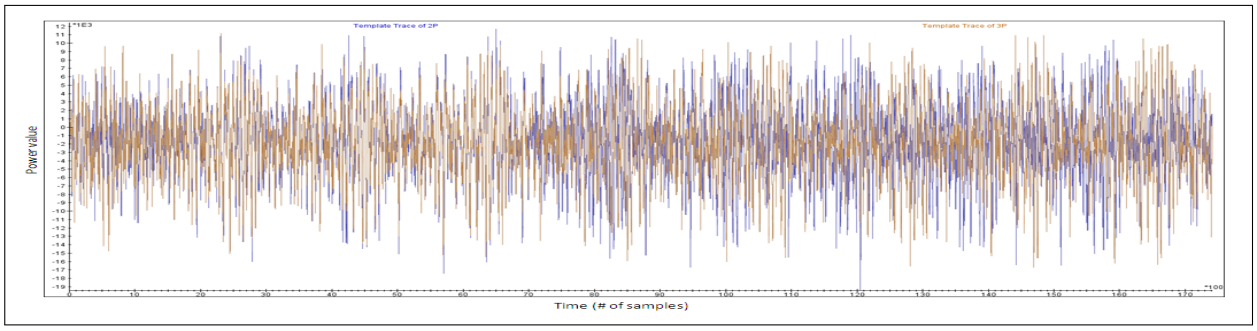


Figure 2: Dés-alignement de deux multiplications de grands nombres dû à la propagation interne de retenue lors de celle-ci

qui correspond mieux. Lorsque les deux hypothèses possèdent la même propagation de retenue (au maximum 14% des cas), la fuite horizontale n'est pas exploitable, mais la meilleure corrélation entre la trace ciblée et les deux traces modèles permet de décider. Le cas précédemment cité correspond à la fuite verticale due aux données manipulées différentes.

Deuxièmement, les auteurs de l'attaque originale n'ont pas considéré le taux de succès pour retrouver tous les bits d'un scalaire. Si un bit retrouvé par l'attaque était faux, cela rend impossible la continuité de l'attaque. Le travail fourni a été l'amélioration de cette attaque en améliorant le taux de succès pour un bit lors de la fuite verticale et en ajoutant une méthode de détection et correction des erreurs possibles. En utilisant un nombre limité d'acquisition, nous pouvons améliorer le taux de succès d'un bit. En effet à l'aide de la moyenne des traces de modèles, nous diminuons le bruit de mesure de ces dernières et améliorons le taux de succès d'un bit. Les résultats sont de 69% de taux de succès pour un bit avec 1 trace de modèle et de 99.80% pour 100 traces de modèle moyennées. Notons que nous ne pouvons pas diminuer le bruit de mesure sur la « trace ciblée », car une contre-mesure inefficace contre notre attaque est le masquage ou la randomisation du scalaire. Notre nouvelle méthode permet aussi détecter et corriger les erreurs dans les bits du scalaire retrouvés, en utilisant l'avantage de la fuite horizontale due à la propagation de retenue. Cela permet d'améliorer le taux de succès globale de l'attaque pour retrouver tous les bits d'un scalaire.

Pour résumer, cette attaque présentée est une « attaque horizontale » utilisant une unique acquisition manipulant une donnée sensible. Mais la condition nécessaire est la connaissance du point d'entrée ainsi que la méthode afin de calculer de manière déterministe les valeurs intermédiaires utilisées à chaque itération pour chaque hypothèse. Comme dans l'article originale, la contre-mesure de masquage du point d'entrée de l'algorithme de multiplication scalaire permet de protéger contre cette attaque. Lors des recherches suivants, nous décidons d'étudier les attaques par canaux auxiliaires lorsque le point d'entrée pour les courbes elliptiques ou le message pour RSA est inconnu, masqué ou le rendre aléatoire.

Analyse du lien entre plusieurs extra-réductions lors d'opérations consécutives

Il existe différentes manières de mettre en œuvre l'arithmétique modulaire utilisée dans la cryptographie. L'optimisation de la multiplication modulaire est très importante pour la cryptographie asymétrique. Dans la section 2.4, les méthodes de réduction après la multiplication les plus courantes sont détaillées. Dans mes travaux concernant les extra-réductions, je me suis plus particulièrement intéressée à la multiplication modulaire de Montgomery. Les extra-réductions dans la multiplication modulaire de Montgomery sont la soustraction à la fin de la réduction pour obtenir un entier compris entre $[0, p[$ (avec p la valeur du module). Les chapitres 5 et 6 expliquent plus en détail l'exploitation des extra-réductions.

L'étude des extra-réductions de Montgomery dans les attaques par canaux auxiliaires datent de 2001 par Walter & Thomson dans [WT01], mais aussi de nombreux travaux sont réalisés par Werner Schindler et ses co-auteurs [SKQ01, Sch02, SW03, ASK05, AS08, Sch15]. Les travaux de Schindler cités ci-après [SKQ01, SW03, ASK05, AS08, Sch15] exploitent les extra-réductions de la multiplication de Montgomery avec des messages choisis pour l'entrée de la multiplication scalaire. Ces attaques sont intéressantes, mais le masquage de l'entrée de la multiplication scalaire est une contre-mesure classique et de plus très efficace contre ces attaques. Walter & Thomson [WT01] et Schindler [Sch02] quant à eux montrent qu'il existe une différence entre le nombre d'extra-réductions de Montgomery lorsque l'opération est un carré modulaire ou une multiplication modulaire (proposition 3.5). Cela permet de retrouver les bits de l'exposant privé lors de l'exécution d'un algorithme classique tel que le « multiplication et carré » (en anglais « Square-and-Multiply »— algorithmes 2.1-2.2). Pour l'attaque de [WT01, Sch02], un attaquant commence par prendre une série d'acquisition avec des messages aléatoires et différents. Notons que si une contre-mesure de masquage de l'entrée est utilisée pour protéger cela n'empêchera pas cette attaque. Pour chaque opération i, l'attaquant repère pour chaque acquisitions q, s'il y a eu une extra-réduction notée $x_q^i = 1$ ou non notée $x_q^i = 0$. L'attaquant calcule la moyenne de présence des extra-réductions sur toutes les acquisitions pour chaque opérations, cela correspond à l'estimation de la probabilité $\hat{\mathbb{P}}(X_i = 1)$. La probabilité de la présence des extra-réductions d'un carré est de $\frac{p}{3R}$ et la probabilité de la présence des extra-réductions d'une multiplication est $\frac{p}{4R}$ (avec p la valeur du module et $R = 2^{\lceil \log_2(p) \rceil}$ la constante de Montgomery). La figure 4 permet d'illustrer l'attaque de Walter & Thomson et Schindler.

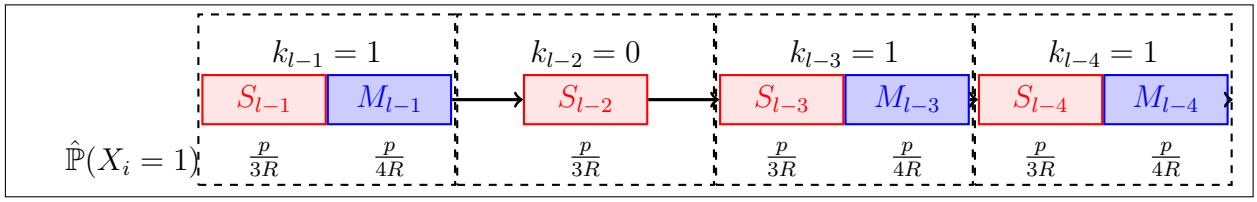


Figure 3: Principe de l'attaque de Walter & Thomson et Schindler

L'utilisation d'un algorithme régulier permet d'éviter cette attaque, en effet comme le montre la figure 4, nous nous pouvons plus différencier un bit égale à « 0 » d'un bit égale à « 1 ».

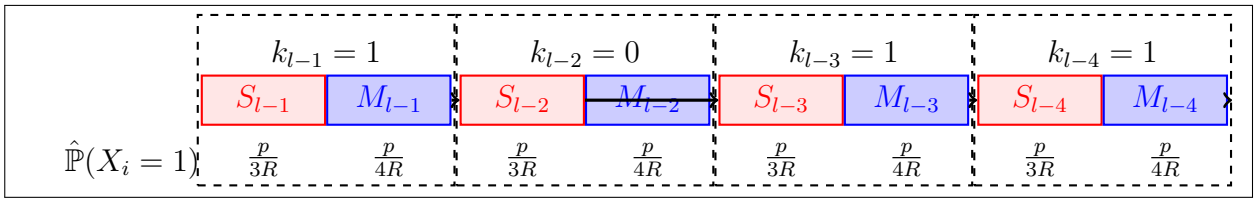


Figure 4: Principe de l'attaque de Walter & Thomson et Schindler sur un algorithme régulier — Echec de l'attaque

Dans les chapitres 5 et 6, notre analyse concernant les extra-réductions de Montgomery est décrite de manière complète. Le chapitre 5 concerne la première partie de ce travail qui a été publié à la conférence internationale CHES 2016. Tandis que le chapitre 6 est l'amélioration de notre analyse en utilisant plus d'information concernant les extra-réductions. Notre première analyse (chapitre 5) a montré qu'il existait une différence exploitable pour la probabilités d'opérations successives lorsqu'un bit est à « 0 » ou un bit égale à « 1 » dans le cadre de l'algorithme régulier « carré-et-multiplication-toujours » (SMA de l'anglais Square-and-Multiply-Always). En effet, la figure 5 permet d'illustrer cette différence.

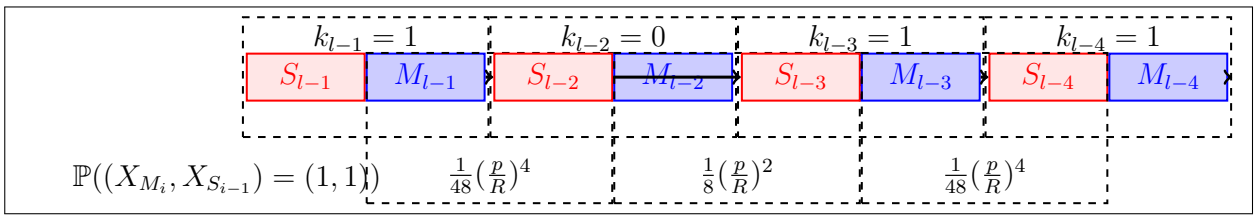


Figure 5: Notre extra-réduction analyse en étudiant une opération de multiplication suivi d'un carré dans un algorithme régulier tel que « Carré-et-multiplication-toujours »

La preuve mathématique du calcul de ces différentes probabilités est décrite section 5.2.4. En utilisant l'algorithme 5.1, cela nous permet de retrouver les $l-1$ premiers bits d'un exposant de l bits juste en affichant le coefficient de Pearson de la loi de probabilités entre les deux opérations, comme le montre la figure 6. Les acquisitions sont réalisées sur un algorithme d'exponentiation modulaire « carré-et-multiplication-toujours » avec lecture du scalaire du bit de poids fort au bit de poids faible. La figure 6 représente l'estimation des coefficients de corrélation de Pearson entre la multiplication et le carré suivant pour les 20 premiers bits d'un exposant choisi aléatoirement avec pour module qui vaut `RSA-1024-p` défini dans la section 3.5.3.

Une nouveauté dans nos travaux concerne aussi la détection des extra-réductions. Premièrement, lorsqu'une opération supplémentaire est appliquée cela se traduit par une consommation de temps différente, c'est le cas dans l'implémentation OpenSSL(voir listing 5.1). Pour identifier l'extra-réduction dans OpenSSL, il suffit de faire une simple analyse par canaux auxiliaires. Pour mettre en évidence la différence de temps, nous avons obtenue des traces de consommations de courant sur un cœur double cortexM0-M4. Nous avons calculer le spectrogramme de ces traces, pour avoir une meilleur visibilité des différentes opérations. Cette technique nous a permis d'identifier la présence et l'absence d'extra-réduction.

Deuxièmement, afin d'éviter cette fuite en temps, l'opération d'extra-réduction peut

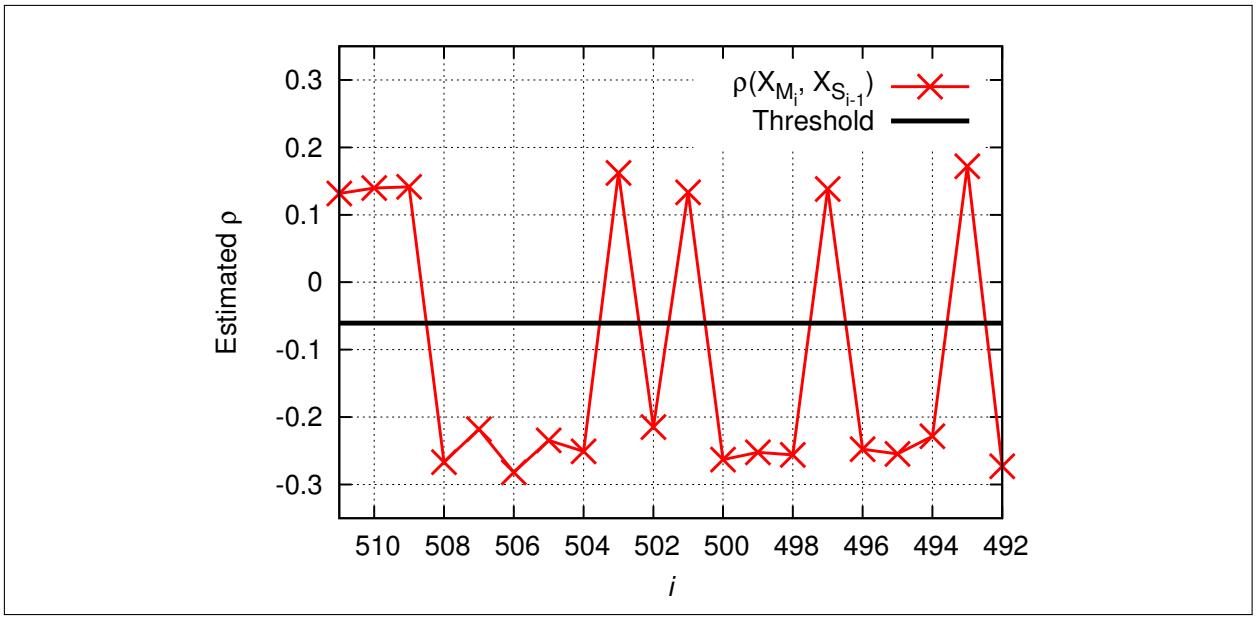


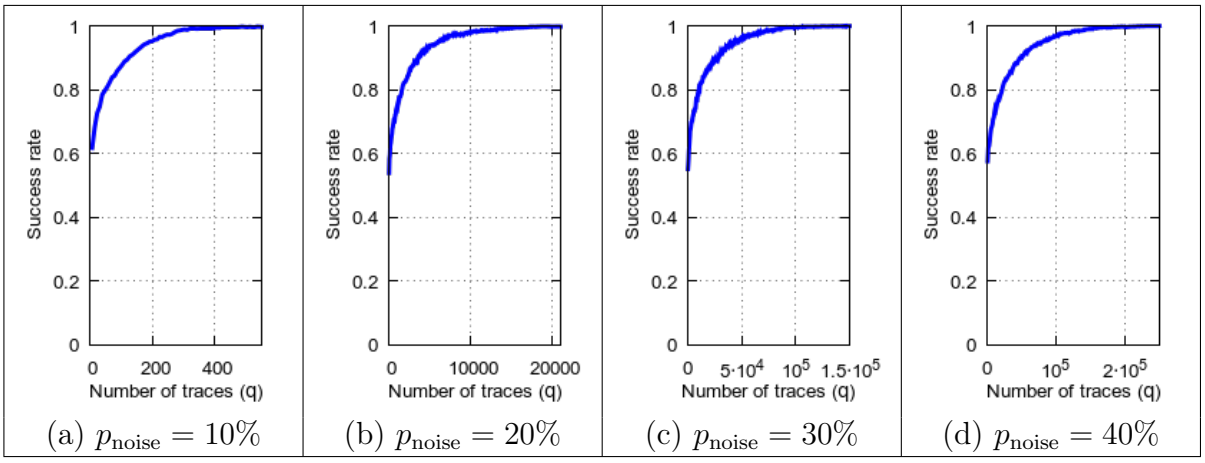
Figure 6: Estimation du coefficient de corrélation de Pearson en utilisant 1000 acquisitions avec des messages inconnus et aléatoires pour les 20 premiers bits de l'exposant.

être compensée, comme le fait un algorithme régulier avec les différents traitements des bits. Dans la bibliothèque mbedTLS, la soustraction finale est composée par une soustraction fantôme (voir listing 5.2). Cette opération fantôme empêche des attaques simples par canaux auxiliaires. Dans le but de différencier les deux opérations (la soustraction utile et la soustraction fantôme), nous allons effectuer une attaque horizontale. Nous allons d'abord construire des lots de traces avec la connaissance si c'est une soustraction utile et si c'est une soustraction fantôme. Sur la trace que nous attaquons nous allons comparer notre trace avec les deux lots et décider lequel est le plus semblable. Le taux de succès de l'attaque horizontale en utilisant deux moyens de comparaison différents — le maximum de corrélation de Pearson et la distance minimal entre les acquisitions — est supérieur à 80%. Cela correspond à un taux de bruit dans la détection des extra-réduction noté p_{noise} à 20%.

Afin de déterminer le taux de succès global de l'attaque utilisant l'analyse de l'extra-réduction, nous avons calcule le taux de succès pour un bit pour différentes valeur de p_{noise}. La figure 7 montre l'évolution des taux de succès pour un bit en fonction du nombre d'acquisition q.

Dans notre partie expérimentale, nous obtenions un bruit p_{noise} inférieur à 20%, il faut 10.000 acquisitions afin d'obtenir un bon taux de succès de notre attaque pour un algorithme « carré-et-multiplications-toujours » en exploitant uniquement la succession de deux opérations.

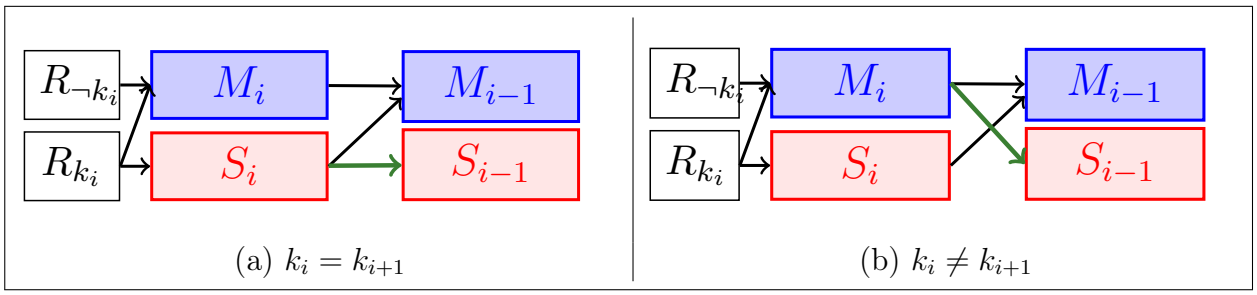
Lors d'une itération de boucle dans un algorithme binaire régulier, il y a toujours deux opérations le carré et la multiplication. En étudiant uniquement deux opérations consécutives, la multiplication et le carré, nous perdons de l'information, c'est pourquoi dans le chapitre 6, nous décrivons une méthode qui optimise notre première analyse. Dans le chapitre 6, nous nous focalisons sur l'algorithme régulier « Échelle de Montgomery » (« Montgomery Ladder » en anglais — algorithme 3.2), mais les mêmes résultats s'appliquent



(a) $p_{\text{noise}} = 10\%$ (b) $p_{\text{noise}} = 20\%$ (c) $p_{\text{noise}} = 30\%$ (d) $p_{\text{noise}} = 40\%$

Figure 7: Évolution du taux de succès pour un bit en fonction du nombre d'acquisition q pour quatre valeurs de bruit p_{noise}

sur l'algorithme « carré-et-multiplications-toujours » aussi.



(a) $k_i = k_{i+1}$ (b) $k_i \neq k_{i+1}$

Figure 8: Dépendance entre les opérations de deux itérations dans un algorithme « Echelle de Montgomery »(algorithme 3.2)

Pour commencer, les liens entre les opérations dans l'algorithme « Échelle de Montgomery » ne sont pas en fonction de la valeur du bits « 0 » et « 1 », mais si le bit étudié k_i est égal ou différent par rapport au bit précédent k_{i+1}. Ce propos est illustré par la figure 8.

Au lieu d'étudier une probabilité à deux variables (chapitre 5) nous allons étudier une probabilité de 4 variables, ou plus si nous voulons étudier 2 bits ou 3 bits en une fois (chapitre 6). Le maximum de vraisemblance sera utilisé comme distingueur à la place du coefficient de corrélation de Pearson. Le lemme 6.3 reprend le calcul de probabilité des extra-réduction pour $2u$ opérations avec $u-1$ le nombre de bits à retrouver. Pour 1 bit à retrouver, nous étudions 4 opérations consécutives. Le résultat du lemme 6.3 est repris dans les tables D.1 et D.2 pour les deux cas. Pour 2 bits à retrouver,nous étudions 6 opérations consécutives illustrées par la figure 9. Le résultat du lemme 6.3 donne quatre cas possibles en fonction des quatre possibilités illustrées par la figure 9. Les probabilités sont données dans les tables D.3-D.4-D.5-D.6.

L'algorithme d'attaque est l'algorithme 6.2. Nous comparons les taux de succès pour 1 bit avec la méthode du chapitre 5 et cette nouvelle méthode dans la figure 10.

Pour un bit le taux de succès croît de 85% à 90% pour 100 acquisitions. Afin de

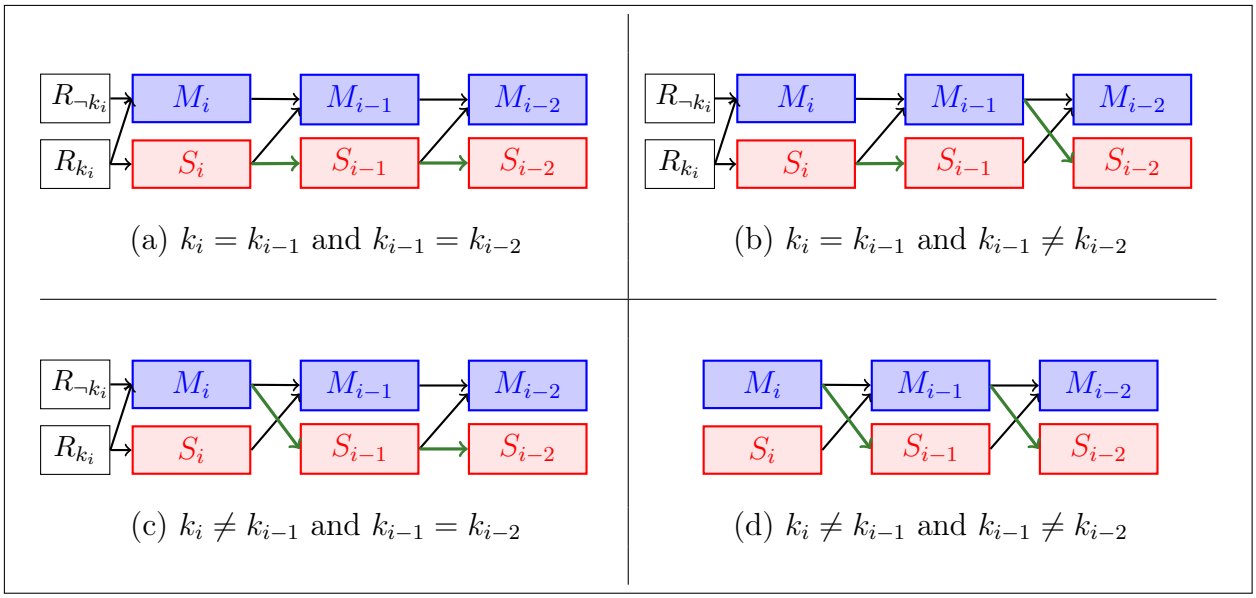


Figure 9: Dépendance entre les opérations de trois itérations dans un algorithme « Échelle de Montgomery »(algorithme 3.2)

mieux visualiser cette amélioration, nous avons calculé le taux de succès pour différentes valeurs de u et différentes probabilités de bruit dans la détection des extra-réductions. La figure 11 représente le taux de succès d'un exposant entier, calculé avec 1000 exposants différents. Ici, nous pouvons observer que notre méthode pour différentes valeurs u permet d'augmenter significativement le taux de réussite par rapport à la méthode décrite dans le chapitre 5. Le nombre d'acquisitions nécessaires pour réussir l'attaque est divisé par un facteur supérieur à 2. Le gain entre les différentes valeurs de u n'est pas significatif. Néanmoins, lorsque $u > 2$, nous pouvons prendre un avantage en essayant de détecter et corriger les bits faux.

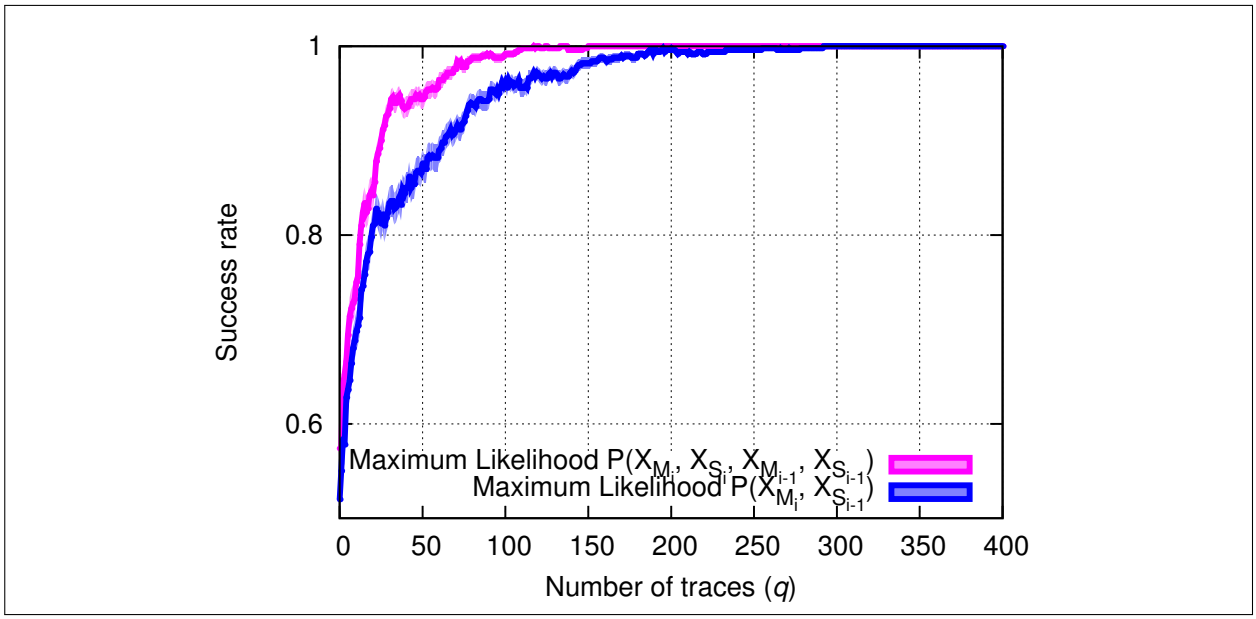
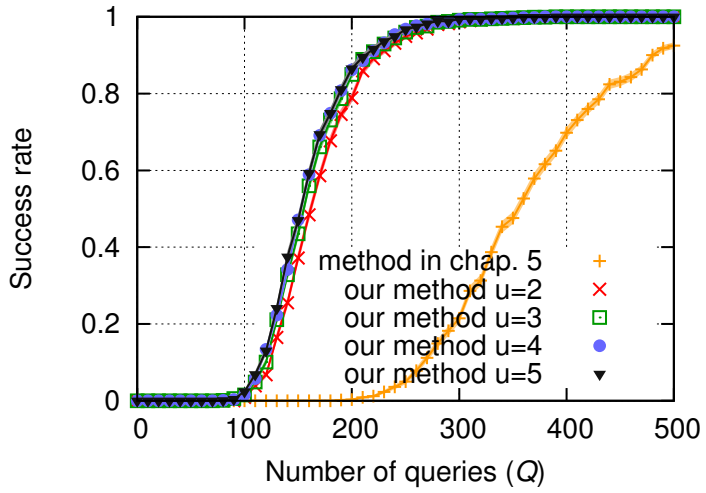


Figure 10: Évolution du taux de succès pour un bit en utilisant 500 expériences entre la méthode du chapitre 5 et l'amélioration décrites dans le chapitre 6 avec $u = 2$ sans erreur dans la détection d'extra-réductions.

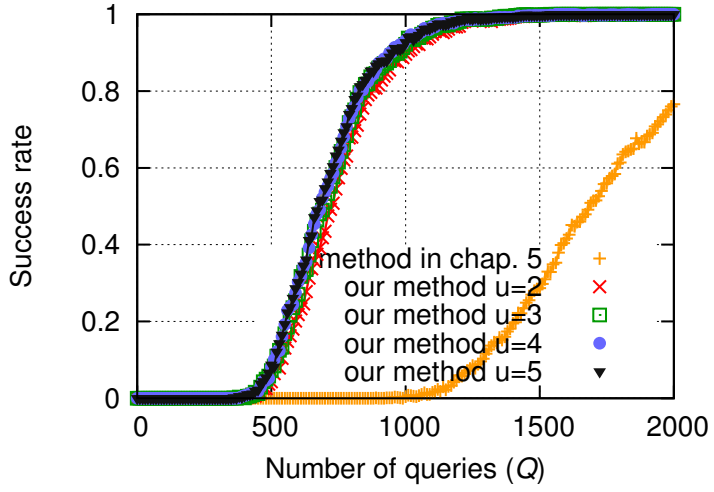
La recherche sur l'analyse des extra-réductions dans les différents scénarios de protection contre les attaques par canaux auxiliaires n'est pas épuisée. En effet, il reste à étudier l'influence d'un masquage de l'exposant sur nos probabilités théoriques. Berzati et al. in [BCG10] montre que selon le masquage certains bits est inneficace. Ce biais dans le masquage pourrait entraîner la réussite d'une nouvelle attaque. Une autre protection est la randomisation du module. Dans la pratique, la taille des nombres aléatoires est limitée. Nous nous demandons si cette protection entraîne une nouvelle attaque ou protège réellement contre les attaques sur les extra-réductions qui dépendent de la valeur du modulo.

Correction d'une contre-mesure contre les attaques par faute sur les courbes elliptiques

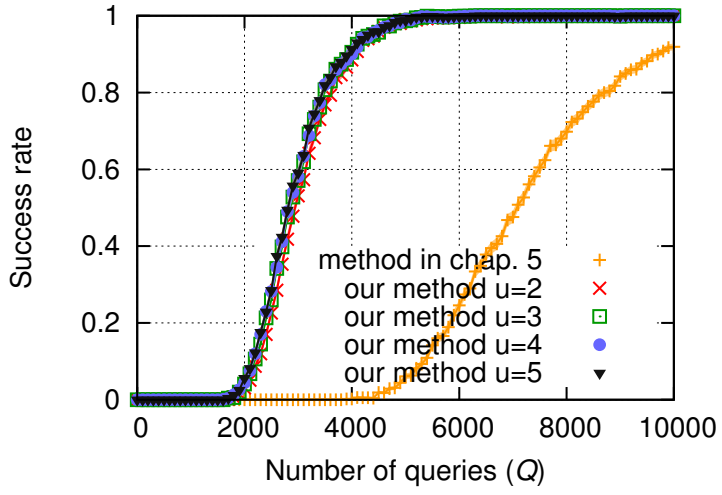
Dans le chapitre 7, nous présentons la correction d'une protection contre les attaques par faute nommée extension modulaire. Cette protection permet de protéger contre les attaques par faute d'ordre 1. La correction expliquée dans le chapitre 7 s'appliquent sur les courbes elliptiques de type Edwards et Twisted Edwards mais aussi sur toutes les courbes dont les formules d'addition et de doublement sont complètes et unifiées. L'étude formelle de cette contre-mesures permet de calculer une probabilité de non détection de faute et d'ainsi d'assurer un certain niveau de sécurité.



(a) $p_{\text{noise}} = 0.00$



(b) $p_{\text{noise}} = 0.10$



(c) $p_{\text{noise}} = 0.20$

Figure 11: Évolution du taux de succès pour tous les bits d'un exposant utilisant 1000 exposants différents pour différentes valeur de p_{noise}.

Publications et présentations en conférences

[DPN+16] Margaux Dugardin, Louiza Papachristodoulou, Zakaria Najm, Lejla Batina, Jean-Luc Danger, and Sylvain Guilley. Dismantling real-world ECC with horizontal and vertical template attacks. In François-Xavier Standaert and Elisabeth Oswald, editors, *Constructive Side-Channel Analysis and Secure Design - 7th International Workshop, COSADE 2016, Graz, Austria, April 14-15, 2016, Revised Selected Papers*, volume 9689 of *Lecture Notes in Computer Science*, pages 88–108. Springer, 2016

[DGD+16] Margaux Dugardin, Sylvain Guilley, Jean-Luc Danger, Zakaria Najm, and Olivier Rioul. Correlated extra-reductions defeat blinded regular exponentiation. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2016

[DGM+16] Margaux Dugardin, Sylvain Guilley, Martin Moreau, Zakaria Najm, and Pablo Rauzy. Using Modular Extension to Provably Protect Edwards Curves Against Fault Attacks. In *PROOFS: Security Proofs for Embedded Systems 2016*, San, United States, August 2016

Publications à venir

JMC17 Margaux Dugardin, Werner Schindler, Sylvain Guilley, Improvement of extra-reduction analysis against blinded regular exponentiation, Journal of Mathematical Cryptography, submitted in 2017

Contents

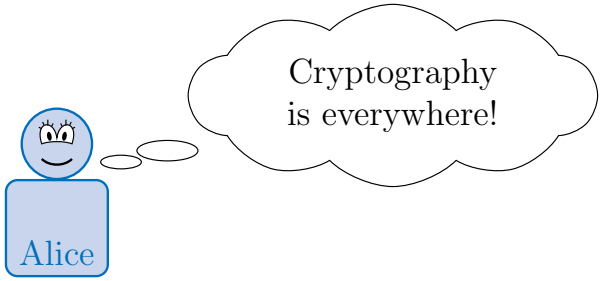
Chapter 1

Introduction

1.1 Alice in cryptoland

I will tell you a typical day for Alice. To go to work, Alice takes public transport using her electronic pass. Upon arriving at work, she opens her office's door with her identification badge. After saying hello to her colleagues, she moves to her office and identifies/authenticates herself on her computer session to have access to her company's network and reads her emails encrypted in order to guarantee their confidentiality. At noon, she sends a message to Bob, so that he can join her to eat. She takes her meal tray and pays at the cash desk with her canteen badge. At the end of her working day, she goes to the pool using her hobbies' card containing 20 hours of swimming. After, she picks up her medications using her vital card at the pharmacy. Then she takes bread at the bakery using the contact-less payment of her credit card. Arriving at home, she decides to buy the train tickets for the next weekend online using an e-card generated on her bank's website. Then she settles on the sofa to watch a movie on demand on her connected TV while eating her meal. At the end of the day, Alice thinks where and when cryptography is present in my day?

The answer is all these daily actions, described previously. In most daily actions, Alice uses embedding devices with cryptography, when Alice uses a card such as electronic pass, identification badge, etc. All these cards permit to identify and authenticate Alice. Each transactions/actions may be made with these card save and have got a signature to avoid the repudiation by Alice. When Alice receives e-mails and sends a message to Bob, the message was encrypted and decrypted to insure the confidentiality and integrity of the message. In general case, the video was encrypted for the movie on pay-tv using broadcast encryption. Alice thinks that cryptography takes an important place in her daily.



1.2 Context

In a world where information is increasingly paperless, computer security has taken an important place in everyday life. Cryptography's aims cause the strongest guarantees on information security. For its security, the cryptosystems claim to be "sure" when they are demonstrated by mathematical proofs. The oldest cryptography names secret key cryptography or symmetric cryptography. It allows encrypting and decrypting a message using a common key shared between each participant. Its main advantage is that the computation is very efficient, but requires the exchange of a common secret. Since three quarters of the 20th century, the other way is asymmetric cryptography. It allows a secure exchange without sharing common secret, but calculations require more memory and computing time. Asymmetric cryptography addresses different needs such as key exchange and digital signature. RSA, Diffie-Hellman, and ElGamal have been used for decades, and elliptic curve cryptography (ECC) algorithms such as ECDSA [ANS99] are more and more deployed. ECC pairing-based cryptography has recently been accelerated in practice and is thus becoming practical [NNS10]. For example, the construction of "pairing-friendly" elliptic curves is an active subject [GV12]. Homomorphic encryption schemes are getting more practical and are progressively considered viable solutions for some real-world applications requiring strong privacy. All these algorithms use large numbers and take place in mathematical structures such as finite rings and fields, which enables powerful mathematical properties but also facilitates attacks.

My doctoral studies focus only on the asymmetric cryptography such as RSA scheme and cryptography based on elliptic curve. The classical algorithms security level is oblivious to the implementation on the cryptosystem on a physical device such as computer, smart-phone, smart-card, etc. In fact, a concrete device possesses physical properties: it consumes power current; it produces electromagnetic and photonic emanation. Moreover, the execution duration of the computation can vary on the concrete device. Each physical parameter cited previously no exhaustively allows a kind of attack, who implies a new evaluation of the security of these cryptosystems. These attacks are a part of cryptanalysis and are named Side Channel Analysis (SCA). These attacks deduced information from the power consumption, the electromagnetic and photonic emanation, etc. A clever attacker can retrieve secret information by analyzing the involuntary leak. To be more precise, there are two ways to extract information from side channel: either observed a leakage during an execution, named passive attack or disturbed using a fault injection during an execution, named active attack. Passive attacks are an observation of some side channels during a cryptographic computation and do not modify its execution. Active attacks disturb voluntarily the execution of the cryptographic computation. The first proof of concept of these attacks is a 20-year old. Since there are an explosion in the number of attacks published, the discovery of new side channel or application of this side channel analyze. Each applied attack depends of the choice of implementation and/or countermeasures used on the device. The main consequence is the security evaluation of cryptosystems on a device becomes increasingly complicated.

This thesis is about the security evaluation of cryptographic algorithm implemented on embedded device, particularly, the security of asymmetric cryptography against side channel attacks. The main side channels used are execution time, power consumption and electromagnetic emanation. The power consumption and the electromagnetic emanation

require probe and oscilloscope in order to be measured. The measurements are named acquisitions. The execution time of asymmetric algorithm is larger than a symmetric algorithm; it increases the memory for each acquisition. The quality of an acquisition depends on the sampling rate on the oscilloscope, the position probing and the quality of the probe. The three parameters should be optimized to improve the quality of the signal information. The increasing of sampling rate of the oscilloscope increases also the memory size of acquisitions. The increasing of the number of acquisition can improve the analysis by reducing the noise of the signal and increasing the signal information. The number of acquisitions must be reduced to limit the global time of the side channel attack. Indeed, a large number of huge size acquisitions increases the measurement time, the pre-processing time like alignment, filtering, etc. and the analysis time, as well as the computational power. In the industrial context, the evaluation of embedded device takes into account time and specialized equipment required for the attacks.

My researches have focused on the side channel attack against asymmetric cryptography specifically RSA and ECC. The RSA scheme relies on the security of the modular exponentiation, while ECC relies on the elliptic curve scalar multiplication. These main operations implemented without protection are vulnerable against side channel attack using a "simple observation" of the time execution. The classical protection on RSA and ECC are similar, because the implementations of the modular exponentiation algorithm and of the elliptic curve scalar multiplication algorithm have several similarities. Historically, the first protections have been developed to protect RSA implementation and afterward they have been adapted for ECC.

1.3 Contributions

In this context, we show that the side channel attacks with a limited number of acquisitions can be sufficient to break the classical security protection on embedded device. The main leakage exploited in this thesis is the execution time difference in the modular arithmetic, especially in the modular multiplication. Also one of the protections on the exploitation of modular multiplication can be the "modular extension". This protection is generally used against fault attacks on RSA. This adaptation on ECC is not trivial and some publications are incorrect. Indeed, the security level of this protection was not proved formally and this protection is ineffective in most of case on fault attacks on ECC.

My researches are composed on three different publications. First, I was interested in the horizontal side channel attack specifically applied on ECC. The main operation in ECC is the elliptic curve scalar multiplication (ECSM). Developers and designers pay attention to the security of this operation. It exits a lot of side channel attacks against this operation depending of this implementation. So, it exists also protections against these attacks. My first work is an attack against a binary algorithm regular and with scalar randomization. A time leakage in the implementation of multiplication are found and exploited in this work.

Second, I investigate a classical modular multiplication named Montgomery multiplication. This multiplication is used for its efficiency and easy adaptation of multiplication operation in constrained device. At the end of the Montgomery modular multiplication, there is a final subtraction, which is exploited by side channel analyzes. Many works have been carried out against this final subtraction. These works used specially the time leak-

age of the final subtractions. We show that when the constant time is applied on the final subtraction, their exploitation is against possible. The novelty of this attack is adapted for the binary regular algorithm, with unknown input there are no difference between an iteration loop when the bit equals "0" or "1". The number of acquisition at the end of the attack depends on the noise in the acquisitions, but remains limited.

Third, the fault attacks are generally powerful attacks against asymmetric cryptography. In fact, with a fault injection the behavior of the device is perturbed and reveals involuntary information. The protection against RSA can generally easily adapted on ECC, but in the modular extension case is not trivial. Indeed, this protection provokes an information leakage and in some case there is a no protection against fault injection. The adaptation of the modular extension on ECC is fixed for the fault attack and the security level of this protection is evaluated for first order fault attacks.

1.4 Outline

The rest of this thesis is represented by the pyramid scheme (figure 1.1).

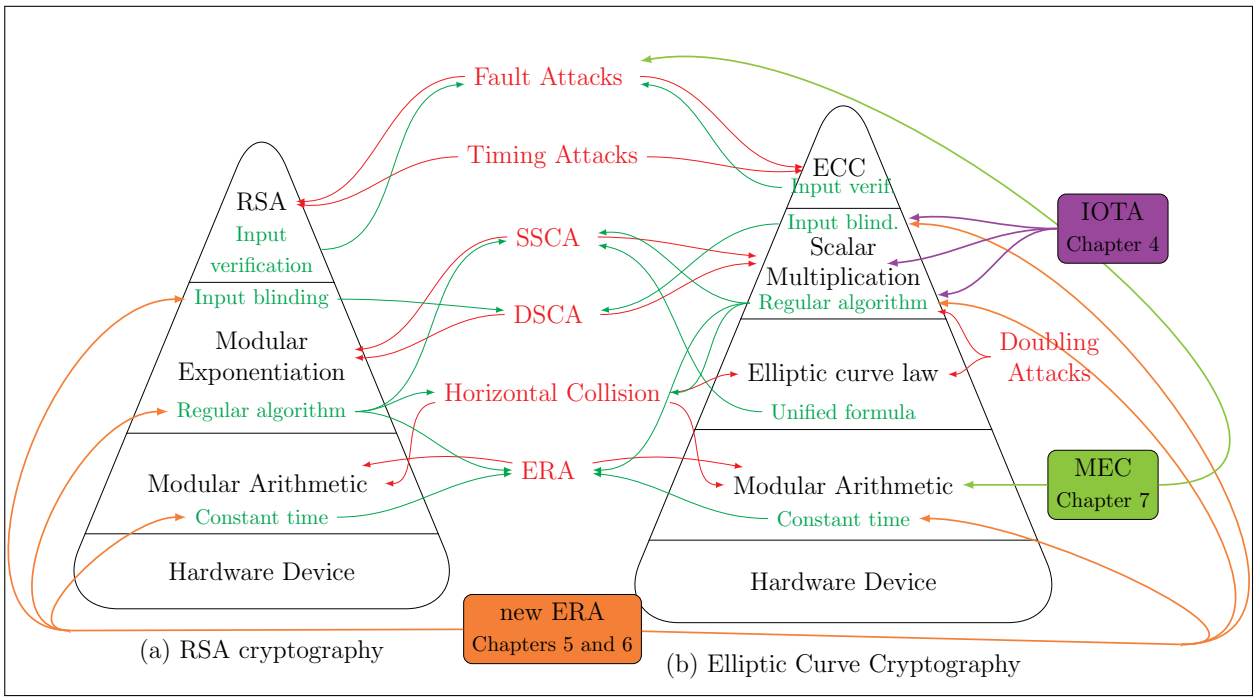


Figure 1.1: Global view of this thesis

The **black piece** of the two pyramids is the mathematical background described in the chapter 2. All the **red arrows** are the classical side channel attacks presented in the chapter 3 with the classical protection represented by the **green arrows**. My contributions are represented by the three rectangles on the pyramid scheme. First the **purple block** corresponds to chapter 4, which is an attack against elliptic curve cryptography with regular algorithm and scalar blinding. Second the chapter 5 corresponds to the **orange block**. It is an attack against regular algorithm with input blinding applied on RSA and ECC schemes. The chapter 6 is an improvement of the previous attack corresponds to the

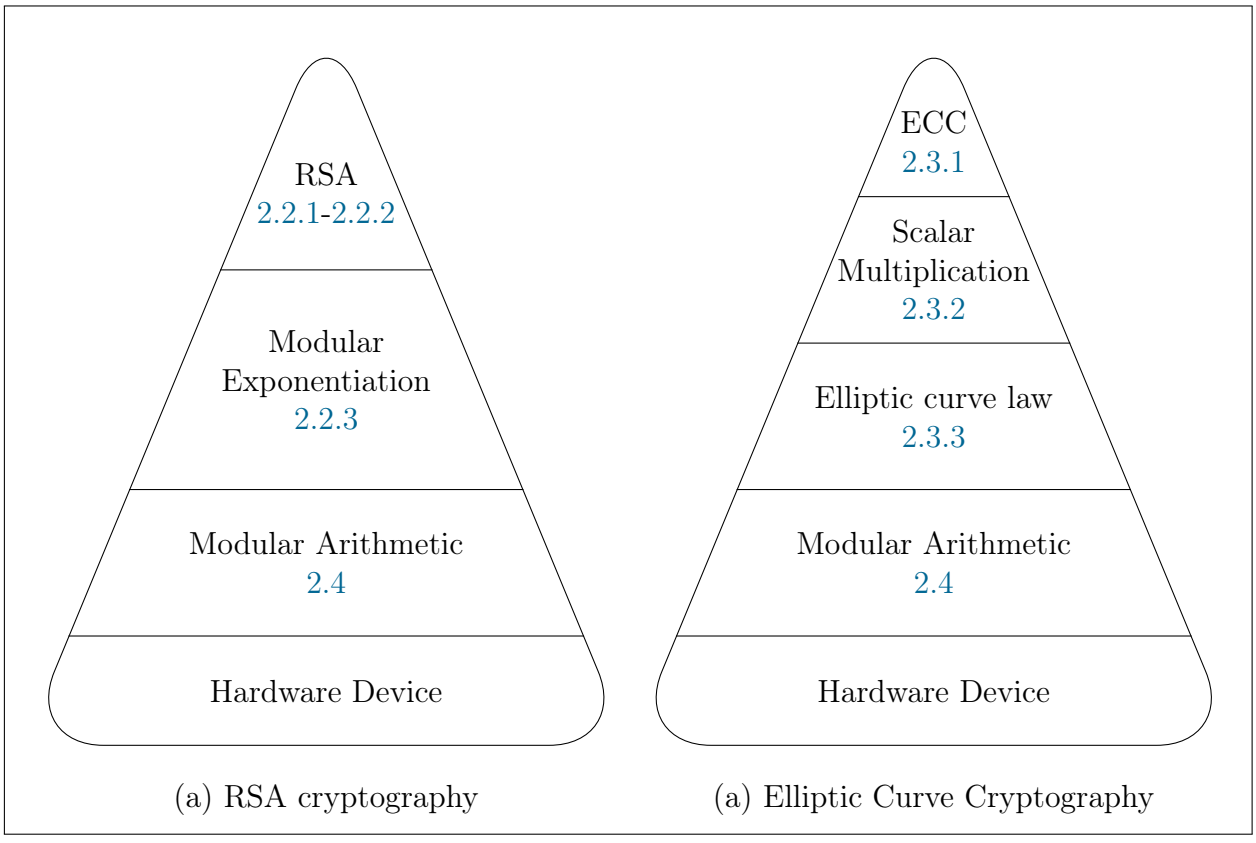
orange block. Finally, the **lime green block** corresponds to the correction of the countermeasures against fault attacks on ECC described in chapter 7.

The conclusion and perspectives are on chapter 8. Informative appendix contain auxiliary information : in chapter A, there are some number theory theorems, the parameters of standardized elliptic curves are remained in chapter B, the algorithms of doubling and adding implemented in PolarSSL are remained in chapter C, and chapter D gives the probabilities law of extra-reduction for consecutive operations. The acronyms are remained in chapter E.

Chapter 2

Asymmetric cryptography and its implementation

In this chapter, we decide to focus on mathematical background of the pyramid scheme. This thesis focuses exclusively on the asymmetric cryptography specially RSA cryptography and Elliptic Curve Cryptography. Firstly, we will introduce the asymmetric cryptography in global view. Secondly, we will define the mathematical background of the RSA scheme (figure 2.1(a)). Thirdly, some definitions concerning the elliptic curve group are evoked (figure 2.1(b)). Finally, we detailed different implementation of the modular multiplication, which is used a lot in RSA and in ECC.



(a) RSA cryptography (a) Elliptic Curve Cryptography

Figure 2.1: Mathematical background of the pyramid scheme presented in this chapter

Contents

2.1 Introduction

In 1976, W. Diffie and M. Hellman [DH76] proposed a new kind of secure communications between two participants (e.g. Alice and Bob). Alice wants to send a message $\mathcal{M}$ to Bob, but the communication channel is not sure. Alice and Bob need to share a common secret in order to encrypt and decrypt their conversation. Symmetric (secret key) cryptography is faster than asymmetric cryptography. Alice and Bob will use the Diffie-Hellman protocol presented in figure 2.2 to exchange a common secret. Then they will use symmetric cryptography using this common secret to communicate. Alice and Bob agree on a cryptography domain composed by a group $\mathbb{G}$, g a generator of the subgroup of $\mathbb{G}$ and n the number of elements in the subgroup generated by g. Alice chooses a random a in $[2, n-1]$, computes $g^a = A$ in the group $\mathbb{G}$ and sends the value A to Bob. Bob chooses a random b in $[2, n-1]$, computes $g^b = B$ in the group $\mathbb{G}$ and sends the value B to Alice. Alice and Bob compute the common secret $\mathcal{K}$ using B and a for Alice and using A and b for Bob. Alice can send the message encrypted with the secret $\mathcal{K}$ named ciphertext. Bob can recover the message by decrypting of the ciphertext with the secret $\mathcal{K}$. An attacker can observe the communication channel. He learns the cryptography domain with the generator g, and the $A = g^a$ and $B = g^b$ values. However he cannot find the common secret generated $\mathcal{K}$ and cannot decrypt the ciphertext $\mathcal{C}$. The security of the asymmetric cryptosystem is based on this "hard" problems named discrete logarithm problem and computational Diffie-Hellman problem. After Diffie-Hellman, other asymmetric protocols are invented such as ElGamal, DSA, ECDSA,etc.

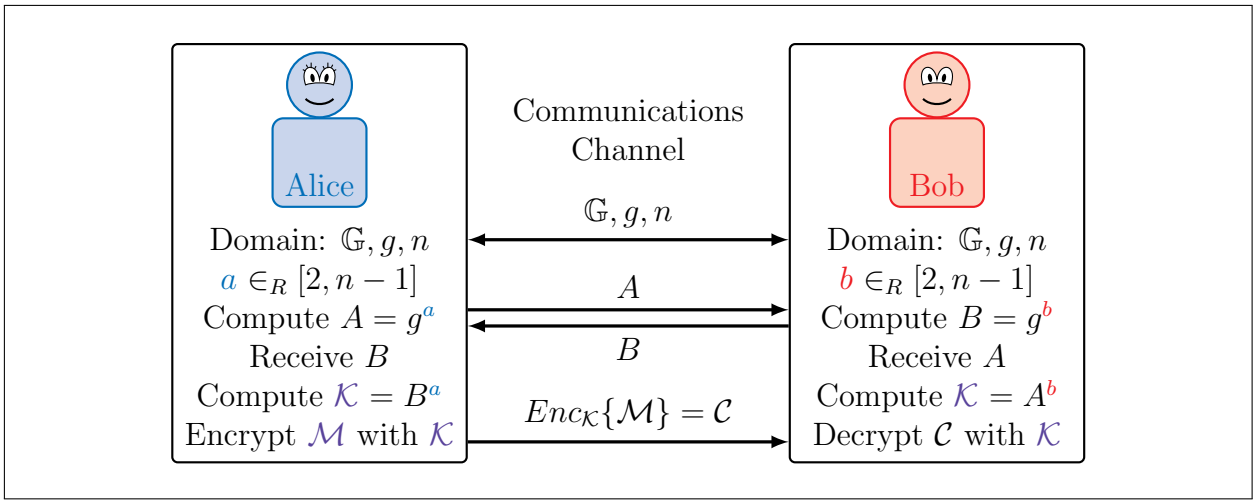


Figure 2.2: Standard protocol to encrypt a message with a common shared secret by Diffie-Hellman

The asymmetric cryptography is also called the public key cryptography (PKC). The main goals of the asymmetric cryptography are confidentiality, data integrity, entity authentication, and signature of data. In fact, the public key cryptography uses one public key to encrypt a message (plaintext) or verify the validity of a signature and one secret key to decrypt a message or sign a message. The two keys were generated for each participant. There are two very useful asymmetric family of algorithms: RSA scheme and Elliptic Curve Cryptography.

1. The first asymmetric cryptography is based on the factorization of product of large prime numbers, named RSA. The name of RSA cryptography becomes from the inventors named Ronald Rivest, Adi Shamir and Leonard Adleman. The RSA cryptographic communications system is a method of asymmetric cryptography widely used in the e-commerce, and more generally to exchange confidential data on internet. In USA, RSA has been patented by the MIT (Massachusetts Institute of Technology) in 1983 [RSA83].

2. The second asymmetric cryptography is based on the group elliptic curve. The elliptic curve was introduced in cryptography in 1985 by Miller [Mil85] and Koblitz [Kob87]. The elliptic curve was defined over a finite field or a field extension. In this thesis, we focus on the elliptic curve cryptography (ECC) over large prime numbers for finite field characteristic.

2.2 Mathematics over RSA scheme

The security of the RSA cryptosystem is based on the hard problem of the factorization of product of two large prime numbers. In fact, the difficulty of factorization of product of two large prime numbers increases with the growth of length of two prime numbers.

2.2.1 RSA keys

In RSA cryptosystem, the decryption key is personal and secret, but it differs to the encryption key, which is broadcasted with other users. This couple of keys is named private/public keys.

Definition 2.1 (Keys in RSA) *The private key is composed by two large prime numbers p and q and an integer d co-prime with $(p-1)(q-1)$. The public key is composed by $n = p \times q$ and e an integer such as $ed \bmod (p-1)(q-1) = 1$. The integer d is named the private exponent, the integer e is the public exponent and n is the public modulus.*

Remark The data of private key are individual and confidential. The data of public key must be share for all users. Computing n and e with the private key is easy using the extended Euclidean algorithm defined in appendix (see algorithm A.1). But compute p, q and d with the public key is a hard problem.

2.2.2 RSA encryption

A data encryption/decryption permits to guarantee the confidentiality of the data. A digital signature allows guaranteeing the integrity of a message, the authentication of message's sender and the non-repudiation of a message. The signature algorithm permits of the message's sender to sign a message. The verification algorithm enables any user who knows the public key of the sender to verify integrity and authenticity of message's signature.

Definition 2.2 (RSA operations) *Let (e, n) and (d, n) a valid key pair of RSA scheme. The modular exponentiation of the integer m using the public exponent e and the public modulus n is the encryption operation noted $m^e \bmod n$. The modular exponentiation of the integer c using the private exponent d and the public modulus n is the decryption operation noted $c^d \bmod n$. The signature s of the representative message m is $s = m^d \bmod n$ using the private exponent d. The verification of the signature is the equality between m and $s^e \bmod n$ using the public exponent e.*

Remark For the efficiency of modular exponentiation using the public exponent e, this value can be small, in most of used cases $e = 3$ or $e = 2^{16} + 1 = 65537$. Warning for $e = 3$, there exists Hastad Attack [Has88]: using fixed public exponent $e = 3$, the same message sent for 3 different users can be retrieved easily. For efficiency of modular exponentiation using the private exponent d, two small private exponents $d_p = d \bmod p$ and $d_q = d \bmod q$ are computed. The modular exponentiation is composed of two modular exponentiation operations: one with d_p, another with d_q and a recombination with Chinese Remainder Theorem method A.3. This method allows approximately to reduced by a factor 4 the time computation.

Proposition 2.3 (RSA encryption/decryption) *Let (p, q, d) be a RSA private key and (e, n) its associated public key. Let m be an integer between $[1, n]$ and $c = m^e \bmod n$.*

$$c^d \bmod n = m^{ed} \bmod n = m \ . \tag{2.1}$$

Proof A proof distinguishing two cases is described.

Case 1. The message m is co-prime with n. The public exponent e is the inverse of d modulo $(p-1)(q-1)$ by definition 2.1, which means there is an integer k such as $ed = 1 + k(p-1)(q-1)$. So, we have:

$$c^d \bmod n = m^{ed} \bmod n = m^{1+k(p-1)(q-1)} \bmod n = m \times (m^{(p-1)(q-1)})^k \bmod n. \tag{2.2}$$

We can conclude using the arithmetic Euler's theorem A.2, because we have $(m^{(p-1)(q-1)})^k \bmod n = 1$ and $m < n$.

Case 2. The message m is not co-prime with n. The message m is a multiple of p or m is a multiple of q.

(a) The message m is a multiple of p, $m \bmod p = 0$ and $m^{ed} \bmod p = 0$. $m < n$ so m and q are co-prime. The little Fermat theorem A.1 implies $m^{ed} \bmod q = m$.

(b) The message m is a multiple of q, $m \bmod q = 0$ and $m^{ed} \bmod q = 0$. $m < n$ so m and p are co-prime. The little Fermat theorem A.1 implies $m^{ed} \bmod p = m$.

In all cases, we have m and m^{ed} solutions of the following system of equations:

$$\begin{cases} x = m \bmod p \\ x = m \bmod q \end{cases}. \tag{2.3}$$

Using the Chinese Remainder Theorem A.3, $m = m^{ed} \bmod pq$.

□

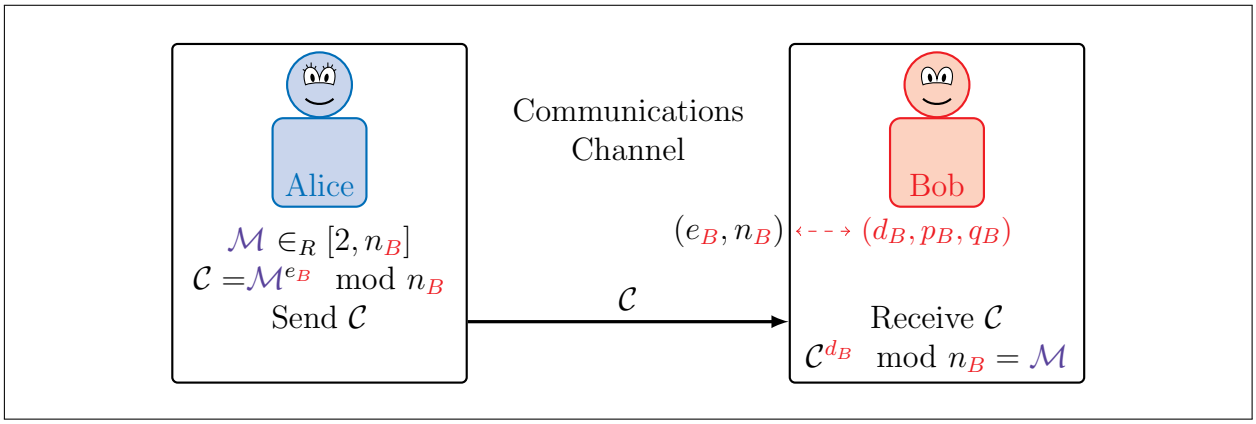


Figure 2.3: RSA scheme to send an encrypted message to one user.

On the figure 2.3, Alice wants to send a message to Bob. Bob generates a RSA keys, and shares to Alice (and others users) his public key (e_B, n_B). Alice's message is $\mathcal{M}$ an integer between $[1, n_B]$. Alice computes the ciphertext $\mathcal{C} = \mathcal{M}^{e_B} \bmod n_B$ using the public key of Bob and sends it to Bob. Bob receives the ciphertext $\mathcal{C}$ and recover the message $\mathcal{M}$ using his private exponent d_B. An attacker knows the public key (e_B, n_B) and the ciphertext $\mathcal{C}$, but he cannot retrieve the message without knowing the private exponent d_B.

2.2.3 RSA implementation

The main operation in RSA in decryption and signature is the modular exponentiation. The developers want to optimize the computation time and make secure this operation, because it manipulates the secret key.

Definition 2.4 (Naive method of modular exponentiation) *The naive method of modular exponentiation is to perform k times a modular multiply of the integer m.*

$$m^k \bmod n = \prod_{i=0}^{k} m \bmod n = (\dots((m \times m \bmod n) \times m \bmod n) \dots \times m \bmod n) \quad . \tag{2.4}$$

The complexity of this method is k modular multiplications. To speed up, the computation of the modular exponentiation, the binary exponent representations can be used.

Definition 2.5 (Binary representations of an integer) *Let k be a positive integer. The binary representation of k is $(k_{l-1}k_{l-2}\dots k_0)_2$ with:*

$$k = \sum_{i=0}^{l-1} k_i 2^i, , k_i \in \{0,1\}, \text{ and } k_{l-1} \neq 0 \quad . \tag{2.5}$$

k_0 was the least significant bit value (lsb). k_{l-1} was the most significant bit value (msb).

Algorithm 2.1: Modular exponentiation: Classical Square and Multiply (Left-to-Right)

Require: $m, k = (k_{l-1}k_{l-2}\dots k_0)_2, p$
Ensure: $m^k \bmod p$
1: $R_0 \leftarrow 1$
2: **for** $i = l-1$ **down to** 0 **do**
3: $R_0 \leftarrow R_0 \times R_0 \bmod p$
4: **if** $k_i = 1$ **then**
5: $R_0 \leftarrow R_0 \times m \bmod p$
6: **end if**
7: **end for**
8: **return** R_0

Algorithm 2.2: Modular exponentiation: Classical Square and Multiply (Right-to-Left)

Require: $m, k = (k_{l-1}k_{l-2}\dots k_0)_2, p$
Ensure: $m^k \bmod p$
1: $R_0 \leftarrow 1, R_1 \leftarrow m$
2: **for** $i = 0$ **to** $l-1$ **do**
3: **if** $k_i = 1$ **then**
4: $R_0 \leftarrow R_0 \times R_1 \bmod p$
5: **end if**
6: $R_1 \leftarrow R_1 \times R_1 \bmod p$
7: **end for**
8: **return** R_0

The complexity of these methods is linear with the size of the exponent k.

2.3 Mathematics over elliptic curve

The use of elliptic curves in cryptography was introduced by H.W. Lenstra, who proposed to break RSA with an integer factorization algorithm based on the arithmetic of elliptic curves. In 1985, Miller [Mil85] and Koblitz [Kob87] presented the asymmetric cryptography based on elliptic curves. The main advantage to use the elliptic curve is the small amount of data space of parameters compared to the RSA.

2.3.1 Definition of elliptic curve domain

Definition 2.6 (Elliptic curve) *Let p be a large prime number or a power of a prime number. On the finite field $\mathbb{F}_p$, an elliptic curve $\mathcal{E}$ in shorted Weierstrass form*[1] *is solutions (x, y) in $\mathbb{F}_p \times \mathbb{F}_p$ satisfying the following equations:*

$$\mathcal{E}(\mathbb{F}_p) : y^2 = x^3 + ax + b \ . \tag{2.6}$$

where the finite field elements in $\mathbb{F}_p$ a and b verify $\Delta(\mathcal{E}(\mathbb{F}_p)) = -16(4a^3 + 27b^2) \neq 0$. Each couple satisfying the shorted Weierstrass equation is a point on elliptic curve. The set of points with an additional point named infinity point $\mathcal{O}_\mathcal{E}$ form a mathematical group.

Remark A couple (x, y) satisfying the equation (2.6) represents the affine coordinates of a point on the elliptic curve $\mathcal{E}$.

Theorem 2.7 (Hasse's theorem) *The number of points on the elliptic curve including the infinity point noted $\#\mathcal{E}(\mathbb{F}_p)$ was*

$$p + 1 - 2\sqrt{p} \geq \#\mathcal{E}(\mathbb{F}_p) \geq p + 1 + 2\sqrt{p} \ . \tag{2.7}$$

The Hasse's theorem permits to conclude that the elliptic curve is finite group.

Definition 2.8 (Generator point) *Let G be a point satisfying the equation (2.6) of the elliptic curve $\mathcal{E}(\mathbb{F}_p)$. The subgroup generated by G is a cyclic group of the curve. Its order is the order of G noted n. The co-factor h is the quotient $\frac{\#\mathcal{E}(\mathbb{F}_p)}{n}$.*

The domain of elliptic curve is defined using the following parameters (p, a, b, G, n, h):

- p is the characteristic of the finite field where the coefficient of the curve are defined,
- a, b are the two parameters of the elliptic curve defined by equation (2.6),
- G is the generator point defined by definition 2.8 In most cases, it is defined by its affine coordinates (x, y),
- n is the order of the subgroup generated by the point G,
- h is the co-factor defined in definition 2.8.

Some recommendations for the security of elliptic curve domain are described:

a) the field characteristic p must be prime and $p = 3 \bmod 4$,

b) the order of subgroup n must be prime greater than 224 bits,

c) $n \neq p$, to avoid the anomal curve,

d) for each r in $\{1, 10^4\}$, $p^r \neq 1 \bmod n$,

e) the decomposition in prime factors of the number of points $\#\mathcal{E}(\mathbb{F}_p)$ must contain a prime number greater than 200 bits.

[1] The Weierstrass form is $\mathcal{W}(\mathbb{F}_p) : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$. The shorted form is $\mathcal{W}(\mathbb{F}_p)$ with $a_1 = a_3 = a_2 = 0$, $a_4 = a$ and $a_6 = b$.

In practical cases, the most used curves are the standardized curves. The recommended curves by the German institute BSI are named `brainpoolPXXXr1` curves with `XXX` the binary length of the modulo [BSI10]. The American curves are recommended by the NIST, which are the `P-XXX` with `XXX` the binary length of the modulo [NIS13]. In our experiment part, we are interested exclusively in the curves on modulo of 256 bits length, `brainpoolP256r1` and `P-256`, their parameters are reminded in appendix B.

Like RSA, after the choice of elliptic curve domain, each participant creates a couple of keys is named private/public keys.

Definition 2.9 (ECC keys) *Let (p, a, b, G, n, h) be the elliptic curve domain. The private key is composed of an integer between $[1, n-1]$ noted d, named private scalar. The public key is composed of a public point $P = [d]G$, the result of the elliptic curve scalar multiplication of the generator point by the private scalar.*

Key agreement. The Diffie-Hellman protocol illustrated by the figure 2.2 can be adapted using the elliptic curve domain. Alice and Bob agree on a point P on the elliptic curve domain. Alice (respectively Bob) chooses a random a (resp. b) and computes $\mathcal{A} = [a]P$ (resp. $\mathcal{B} = [b]P$) and shares $\mathcal{A}$ to Bob (resp. $\mathcal{B}$ to Alice). Alice computes $\mathcal{K} = [a]\mathcal{B}$ and Bob computes $\mathcal{K} = [b]\mathcal{A}$, by associativity and commutativity they share the same secret key $\mathcal{K}$. For more details, see ECDH protocol in [ANS96].

Digital signature. There are many[2] signature protocols using the elliptic curve cryptography domain. The most popular standardized by ANSI is the ECDSA [ANS99]. The algorithm 2.3 presents the method of ECDSA signature (r, s) for a message $\mathcal{M}$ by a user having generated a key pair (d, P) on the elliptic curve domain (p, a, b, G, n, h). Any user knowing the public point P and the message $\mathcal{M}$ can verify the authenticity of the ECDSA signature (r, s) using the computation of the elliptic curve point Q as:

$$Q = \left[\frac{Hash(\mathcal{M})}{r} \bmod n\right] G + \left[\frac{s}{r} \bmod n\right] P \ . \tag{2.8}$$

The signature is *valid* if the x-coordinates of Q in affine representation reduced by modulo n equals r, and corrupts otherwise.

2.3.2 Elliptic curve scalar multiplication

The main operation in ECDH [ANS96] and ECDSA [ANS99] was the elliptic curve scalar multiplication. The developers want to optimize the time consumption and make this operation secure, because it manipulates the secret key or data permits to retrieve the secret key [3].

[2] For more details see the standard ISO/IEC 14888-3:2016

[3] In ECDSA, the secret key d can be retrieved using the knowledge of scalar value k and the ECDSA signature (r, s) of the message $\mathcal{M}$ by using this formula $d = \frac{sk - Hash(\mathcal{M})}{r} \bmod n$

Algorithm 2.3: Signature Algorithm ECDSA:

Different representations

Require: (q, a, b, G, n, h) an elliptic curve domain, d private scalar, $\mathcal{M}$ message.
Ensure: ECDSA signature (r, s) of $\mathcal{M}$. La signature ECDSA

1: $k \leftarrow RNG(\{1, 2, \cdots, n-1\})$
2: $Q \leftarrow [k]G$
3: $r \leftarrow x_Q \bmod n$ ▷ x_Q is x-coordinate of point Q in affine representation.
4: **if** $r = 0$ **then**
5: goto 1
6: **end if**
7: $k_{inv} \leftarrow k^{-1} \bmod n$
8: $s \leftarrow k_{inv} \cdot (r \cdot d + Hash(M)) \bmod n$
9: **if** $s = 0$ **then**
10: goto 1
11: **end if**
12: return (r, s)

Definition 2.10 (Elliptic curve scalar multiplication (ECSM)) *The elliptic curve scalar multiplication of a point P by an integer k named scalar is k times the additions of the point P. The ECSM is noted:*

$$[k]P = \underbrace{P +_{\mathcal{E}} \ldots +_{\mathcal{E}} P}_{k \text{ times}} . \tag{2.9}$$

where $+_{\mathcal{E}}$ is the group law on elliptic curve defined in definition 2.11.

The complexity of this method is k elliptic curve operations. Like RSA, in order to optimize the time of the computation of the scalar multiplication, the binary representation of the scalar (see definition 2.5) can be used. The two following algorithms correspond to algorithm 2.1 and algorithm 2.2 in RSA scheme.

Algorithm 2.4: ECSM: Classical Double and Add (Left-to-Right)

Require: $\mathcal{E}(\mathbb{F}_p), P, k = (k_{l-1}k_{l-2}\ldots k_0)_2$
Ensure: $[k]P$

1: $R_0 \leftarrow \mathcal{O}_{\mathcal{E}}$
2: **for** $i = l-1$ **down to** 0 **do**
3: $R_0 \leftarrow R_0 +_{\mathcal{E}} R_0$
4: **if** $k_i = 1$ **then**
5: $R_0 \leftarrow R_0 +_{\mathcal{E}} P$
6: **end if**
7: **end for**
8: **return** R_0

Algorithm 2.5: ECSM: Classical Double and Add (Right-to-Left)

Require: $\mathcal{E}(\mathbb{F}_p), P, k = (k_{l-1}k_{l-2}\ldots k_0)_2$
Ensure: $[k]P$

1: $R_0 \leftarrow \mathcal{O}_{\mathcal{E}}, R_1 \leftarrow \mathcal{O}_{\mathcal{E}}$
2: **for** $i = 0$ **to** $l-1$ **do**
3: **if** $k_i = 1$ **then**
4: $R_0 \leftarrow R_0 +_{\mathcal{E}} P$
5: **end if**
6: $R_1 \leftarrow R_1 +_{\mathcal{E}} R_1$
7: **end for**
8: **return** R_0

The complexity of these methods is linear with the size of the scalar k.

2.3.3 Group law: Addition and doubling operations

Definition 2.11 (Elliptic curve group operations) *The operation on the elliptic curve points group is denoted $+_{\mathcal{E}}$, this operation proposes different types of computation operations according to the input points. Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two elliptic curve points $\mathcal{E}$. We define the opposite of the point P by $-_{\mathcal{E}}P = (x, -y)$. The operation between P and Q over the elliptic curve respects the following properties:*

- *If $P = \mathcal{O}_{\mathcal{E}}$, then $P +_{\mathcal{E}} Q = Q$.*
- *If $P = -_{\mathcal{E}}Q$, then $P +_{\mathcal{E}} Q = \mathcal{O}_{\mathcal{E}}$.*
- *If $P = Q$, then $P +_{\mathcal{E}} Q = ECDBL(P)$ is named the doubling operation.*
- *Otherwise, then $P +_{\mathcal{E}} Q = ECADD(P, Q)$ is named the addition operation.*

Using these operation properties, the set of the points over elliptic curve forms an abelian group with the neutral element is $\mathcal{O}_{\mathcal{E}}$.

Remark If the doubling operation and addition have the same formula, then the formula is *unified*. If there are no specific formula for the infinity point input and opposite points, then the formula is *complete*.

Some particular elliptic curves can be defined using another form than the shorted Weierstrass form defined by equation (2.6). The group operation uses different formula, to be more efficient or use complete and unified formula. The Montgomery form permits to be more efficient in Montgomery Ladder algorithm. The unified formula for addition and doubling are more efficient on Edwards, Hessian, and Jacobi curves. In [BL, Ver12], the different formulas of addition and doubling operation depending of the curve form are detailed. It is important to note that a point on an elliptic curve may have different representations: affine, projective, jacobian, etc. Firstly, we focus on standardized curves in shorted Weierstrass form by the BSI the `brainpoolP256r1` and the NIST `P-256` in affine and jacobian representation. Secondly, the Edwards and Twisted Edward are detailed for the complete formula of addition in affine and projective representation. To compare the efficiency of the elliptic curve operations, we define theses notations corresponding to the cost of the modular arithmetic over the finite field $\mathbb{F}_p$:

- I_p is the cost of the modular inversion corresponds to 100 modular multiplications.
- M_p is the cost of the modular multiplication.
- S_p is the cost of the modular square.
- A_p is the cost of the modular addition/subtraction.
- C_p^z is the cost of the modular multiplication by element $z \in \mathbb{F}_p$.

Affine coordinates in shorted Weierstrass form. The affine representation is the classical representations to define a point an elliptic curve. The affine representation is a couple (x, y) which verifies the equation (2.6). This point representation is unique.

Definition 2.12 (Elliptic curve operations using affine representation) *Let $P = (x_1, y_1)$ be a point in affine representation and $Q = (x_2, y_2)$ be another point such as P and Q satisfying the elliptic curve equation (2.6), $P \neq \mathcal{O}_\mathcal{E}$, $Q \neq \mathcal{O}_\mathcal{E}$, and $P \neq \pm_\mathcal{E} Q$. We define $R = (x_3, y_3)$, the result of the following elliptic curve operation on $\mathcal{E}(\mathbb{F}_p)$.*

$$R = ECADD(P, Q): \qquad \begin{cases} \lambda = \dfrac{y_2 - y_1}{x_2 - x_1} \\ x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_3 - x_1) - y_1 \end{cases} . \tag{2.10}$$

Cost: $1I_p + 1M_p + 1S_p + 5A_p$.

$$R = ECDBL(P): \qquad \begin{cases} \lambda = \dfrac{3x_1^2 + a}{2y_1} \\ x_3 = \lambda^2 - 2x_1 \\ y_3 = \lambda(x_3 - x_1) - y_1 \end{cases} . \tag{2.11}$$

Cost: $1I_p + 1M_p + 2S_p + 4A_p$.

Remark A point has only one representation in affine coordinates. The first advantage is the comparison between two points. This representation is generic, even if the computations were made in a different representation for efficient or security reasons, it is always possible to return to the affine form for compatibility reasons. This is why we will always use the affine representation for the input and output of our implemented algorithms. The main drawback in affine computation is the modular inversion, which is an expensive operation, generally the cost corresponds to 100 modular multiplications.

Jacobian coordinates in shorted Weierstrass form. In order to optimize the elliptic curve operations, some different representations of points can be used. One of them named the jacobian representations is defined by a triple (X, Y, Z) in $\mathbb{F}_p^3$.

Definition 2.13 (Jacobian coordinates on shorted Weierstrass form) *Let (X, Y, Z) be a triple in $\mathbb{F}_p^3$. The jacobian coordinates (X, Y, Z) of a point over an elliptic curve satisfied the following equation:*

$$\mathcal{E}(\mathbb{F}_p) : Y^2 Z = X^3 + aXZ^2 + bZ^3 \ . \tag{2.12}$$

where the finite field elements a and b are the parameters of the elliptic curve over $\mathbb{F}_p$ defined by definition 2.6

Proposition 2.14 (Multi-coordinate of same point in jacobian representation) *An elliptic curve point are several jacobian representations.*

$$\forall \mu \in \mathbb{F}_p, \mu \neq 0, (\mu^2 X, \mu^3 Y, \mu Z) = (X, Y, Z) \ . \tag{2.13}$$

Proposition 2.15 (Morphism jacobian to affine coordinates) *A point in jacobian representation (X, Y, Z) with $Z \neq 0$ corresponds to a point in affine representation (x, y) with the following map:*

$$\begin{aligned} \{jacobian,\ Z \neq 0\} &\to \{affine\} \\ (X, Y, Z) &\mapsto (x, y) = \left(\frac{X}{Z^2}, \frac{Y}{Z^3}\right) \ . \end{aligned} \tag{2.14}$$

Remark This map is surjective but not injective, because the jacobian representation is not unique (see proposition 2.14). This map is named the *normalization* operation.

Proposition 2.16 (Morphism affine to jacobian coordinates) *A point in affine coordinates (x, y) can be represented by a point in jacobian coordinates with Z-coordinates equals 1. The map is:*

$$\begin{aligned} \{affine\} &\rightarrow \{jacobian\} \\ (x, y) &\mapsto (X, Y, Z) = (x, y, 1) \ . \end{aligned} \tag{2.15}$$

Remark This map is injective but not surjective.

Definition 2.17 (Elliptic curve operations using jacobian representation) *Let $P = (X_1, Y_1, Z_1)$ be a point in jacobian representation and $Q = (X_2, Y_2, Z_2)$ be another point such as P and Q satisfying the elliptic curve equation (2.12) with Z-coordinates do not equal to zero, and $P \neq \pm_{\mathcal{E}} Q$. We define $R = (X_3, Y_3, Z_3)$, the result of the following elliptic curve operation on $\mathcal{E}(\mathbb{F}_p)$*

$$R = ECADD(P, Q):$$

$$\begin{cases} X_3 = 4(Y_2 Z_1^3 - Y_1 Z_3)^2 - 2(X_1 Z_2^2 - X_2 Z_1^2)^3 - 4X_1 Z_2^2 (X_1 Z_2^2 - X_2 Z_1^2)^2 \\ Y_3 = 2(Y_2 Z_1^3 - Y_1 Z_2^3)(2X_1 Z_2^2 (X_1 Z_2^2 - X_2 Z_1^2)^2 - X_3) - 4Y_1 Z_2^3 (X_1 Z_2^2 - X_2 Z_1^2)^3 \\ Z_3 = ((Z_1 + Z_2)^2 - Z_1^2 - Z_2^2)(X_1 Z_2^2 - X_2 Z_1^2) \end{cases} . \tag{2.16}$$

Cost[4]*: $11M_p + 5S_p + 9A_p$.*

$$R = ECDBL(P):$$

$$\begin{cases} X_3 = (3X_1^3 + aZ_1^4)^2 - 4((X_1 + Y_1^2)^2 - Y_1^4 - X_1^2 \\ Y_3 = (6((X_1 + Y_1^2)^2 - Y_1^4 - X_1^2) - (3X_1^3 + aZ_1^4)^2)(3X_1^3 + aZ_1^4) - 8Y_1^4 \\ Z_3 = (Y_1 + Z_1)^2 - Y_1^2 - Z_1^2 \end{cases} . \tag{2.17}$$

Cost[5]*: $1M_p + 8S_p + 1C_p^a + 10A_p$.*

In some cases, the two representations (affine/jacobian) can be used in order to improve the elliptic curve computation.

Definition 2.18 (Elliptic curve addition using jacobian/affine representation) *Let $P = (X_1, Y_1, Z_1)$ be a point in jacobian representation and $Q = (x_2, y_2)$ be another point in affine representation such as $P \neq \pm_{\mathcal{E}} Q$. We define $R = (X_3, Y_3, Z_3)$, the result of the following elliptic curve addition on $\mathcal{E}(\mathbb{F}_p)$ named mixed-addition:*

$$R = ECADD(P, Q):$$

$$\begin{cases} X_3 = (Z_1^3 y_2 - Y_1)^2 - 2X_1 (Z_1^2 x_2 - X_1)^2 - (Z_1^2 x_2 - X_1)^3 \\ Y_3 = X_1 (Z_1^2 x_2 - X_1)^2 (Z_1^3 y_2 - Y_1) - Y_1 (Z_1^2 x_2 - X_1)^3 \\ Z_3 = Z_1 (Z_1^2 x_2 - X_1) \end{cases} . \tag{2.18}$$

Cost[6]*: $7M_p + 4S_p + 9A_p$.*

[4] In order to have this efficiency, use "add-2007-bl" on [BL]
[5] In order to have this efficiency, use "dbl-2007-bl" on [BL]
[6] In order to have this efficiency, use "madd-2007-bl" on [BL]

Edwards curve group law. In mathematics, the Edwards curves are a family of elliptic curves studied by Harold M. Edwards in 2007 [Edw07]. Technically, an Edwards curve is not elliptic, because it has singularities; but resolving those singularities produces an elliptic curve. The concept of elliptic curves over finite fields is widely used in elliptic curve cryptography. Applications of Edwards curves to cryptography were developed by Bernstein and Lange: they pointed out several advantages of the Edwards form in comparison to the more well-known shorted Weierstrass form.

Definition 2.19 (Edwards curves) *On the finite field $\mathbb{F}_p$ with p a prime number, an elliptic curve in Edwards form has parameters c, d in the finite field $\mathbb{F}_p$ and coordinates (u, v) satisfying the following equation:*

$$Ed(\mathbb{F}_p) : u^2 + v^2 = c^2(1 + du^2v^2), \text{ with } cd(1 - c^4d) \neq 0 \ . \tag{2.19}$$

The main advantage to use the Edwards curves is that addition formulas are *unified*, meaning that there are no tests to verify if the two input points are equal, opposite or different.

Definition 2.20 (Edwards unified addition on affine coordinates) *Let $P = (u_1, v_1)$ be a point in affine representation and $Q = (u_2, v_2)$ be another point such as P and Q satisfying the Edwards curve equation (2.19). We define $R = (u_3, v_3)$, the result of the unified addition on $Ed(\mathbb{F}_p)$, $R = P +_{\mathcal{E}} Q$ defined by the following system of equations:*

$$\begin{cases} u_3 = \dfrac{u_1v_2 + v_1u_2}{c(1 + du_1u_2v_1v_2)} \\ v_3 = \dfrac{v_1v_2 - u_1u_2}{c(1 - du_1u_2v_1v_2)} \end{cases} . \tag{2.20}$$

Cost: $2I_p + 5M_p + 1C_p^c + 1C_p^d + 4A_p$.
The affine negation formula is as expected: $-_{\mathcal{E}}(u_1, v_1) = (-u_1, v_1)$.
The neutral element $\mathcal{O}_{\mathcal{E}}$ of the curve is the point $(0, c)$.

Remark Contrary to shorted Weierstrass curves, the neutral element $\mathcal{O}_{\mathcal{E}}$ is not special (there is no abstract "*point at infinity*"), but verifies the curve equation. The point $(0, -c)$ has order 2. The points $(c, 0)$ and $(-c, 0)$ have order 4.

Proposition 2.21 *Addition law on Edwards curves is* complete *if d is a non-square in $\mathbb{F}_p$.*

Proof See the proof by Bernstein and Lange in [BL07].

Remark This means that the addition formula is valid for all points, with no exception like neutral element. That is one of the advantages of Edwards curves over shorted Weierstrass curves in which the addition law is *not complete*: a *complete* addition law provides some resistance to side channel attacks.

To be more efficient, we use the unified projective coordinates to the addition law. Like Jacobian representation, the projective coordinate is composed by a triple (U, V, W).

Definition 2.22 (Projective coordinates on Edwards form) *Let (U, V, W) be a triple in $\mathbb{F}_p^3$. The projective coordinates (U, V, W) of a point over an Edwards curve satisfied the following equation:*

$$Ed(\mathbb{F}_p) : U^2W^2 + V^2W^2 = c^2(W^4 + dU^2V^2) \ . \tag{2.21}$$

where the finite field elements c and d are the parameters of the elliptic curve over $\mathbb{F}_p$ defined by definition 2.19

Proposition 2.23 (Multi-coordinate of same point in projective representation) *An elliptic curve point are several projective representations.*

$$\forall \mu \in \mathbb{F}_p, \mu \neq 0, (\mu U, \mu V, \mu W) = (U, V, W) \ . \tag{2.22}$$

Proposition 2.24 (Morphism projective to affine coordinates) *A point in projective representation (U, V, W) with $W \neq 0$ corresponds to a point in affine representation (u, v) with the following map:*

$$\begin{aligned} \{projective,\ W \neq 0\} &\to \{affine\} \\ (U, V, W) &\mapsto (u, v) = \left(\frac{U}{W}, \frac{V}{W}\right) \ . \end{aligned} \tag{2.23}$$

Remark This map is surjective but not injective, because the projective representation is not unique (see proposition 2.23). This map is named the *normalization* operation like jacobian representation.

Proposition 2.25 (Morphism affine to projective coordinates) *A point in affine coordinates (u, v) can be represented by a point in projective coordinates with W-coordinates equals 1. The map is:*

$$\begin{aligned} \{affine\} &\to \{projective\} \\ (u, v) &\mapsto (U, V, W) = (u, v, 1) \ . \end{aligned} \tag{2.24}$$

Remark This map is injective but not surjective.

Definition 2.26 (Edwards unified addition on projective representation) *Let $P = (U_1, V_1, W_1)$ be a point in projective representation and $Q = (U_2, V_2, W_2)$ be another point with such as P and Q satisfying the elliptic curve equation (2.21). We define $R = (U_3, V_3, W_3)$, the result of the unified addition on $Ed(\mathbb{F}_p)$, $R = P +_{\mathcal{E}} Q$ defined by the following system of equations:*

$$\begin{cases} U_3 = W_1W_2(U_1V_2 + U_2V_1) \\ V_3 = W_1W_2(W_1^2W_2^2 + dU_1U_2V_1V_2)(V_1V_2 - U_1U_2) \\ W_3 = c(W_1^4W_2^4 - d^2U_1^2U_2^2V_1^2V_2^2) \end{cases} \ . \tag{2.25}$$

Cost[7] *: $10M_p + 1S_p + 1C_p^c + 1C_p^d + 7A_p$.*

[7] In order to have this efficiency, use "add-2007-bl-2" on [BL] or on [BL07, Sec. 4, page 9].

Twisted Edwards curve group law. Twisted Edwards curves are a generalization of Edwards curves [BBJ+08].

Definition 2.27 (Twisted Edwards curves) *Let p a prime number. On the finite field $\mathbb{F}_p$, an elliptic curve in twisted Edwards form has parameters a, d in the finite field $\mathbb{F}_p$ and coordinates (u, v) satisfying the following equation:*

$$TEd(\mathbb{F}_p) : au^2 + v^2 = 1 + du^2v^2, \text{ with } ad(a-d) \neq 0 \ . \tag{2.26}$$

Like Edwards curves, the addition formulas are unified.

Definition 2.28 (Twisted Edwards unified addition on affine coordinates) *Let $P = (u_1, v_1)$ be a point in affine representation and $Q = (u_2, v_2)$ be another point such as P and Q satisfying the Edwards curve equation (2.26). We define $R = (u_3, v_3)$, the result of the unified addition on $TEd(\mathbb{F}_p)$, $R = P +_{\mathcal{E}} Q$ defined by the following system of equations:*

$$\begin{cases} u_3 = \dfrac{u_1v_2 + v_1u_2}{1 + du_1u_2v_1v_2} \\ v_3 = \dfrac{v_1v_2 - au_1u_2}{1 - du_1u_2v_1v_2} \end{cases} . \tag{2.27}$$

Cost: $2I_p + 5M_p + 1C_p^a + 1C_p^d + 4A_p$.
Affine negation formula is natural: $-_{\mathcal{E}}(u_1, v_1) = (-u_1, v_1)$.
The neutral element is $(0, 1)$.

Proposition 2.29 *Addition law on Twisted Edwards curves is* complete *if a is a square and d is a non-square*

Proof See the proof by Bernstein and Lange in [BL07].

To be more efficient, we use the unified projective coordinates to the addition law.

Definition 2.30 (Projective coordinates on Twisted Edwards form) *Let (U, V, W) be a triple in $\mathbb{F}_p^3$. The projective coordinates (U, V, W) of a point over a Twisted Edwards curve satisfy the following equation:*

$$TEd(\mathbb{F}_p) : aU^2W^2 + V^2W^2 = W^4 + dU^2V^2 \ . \tag{2.28}$$

where the finite field elements a and d are the parameters of the elliptic curve over $\mathbb{F}_p$ defined by definition 2.27

Definition 2.31 (Twisted Edwards unified addition on projective representation) *Let $P = (U_1, V_1, W_1)$ be a point in projective representation and $Q = (U_2, V_2, W_2)$ be another point with such as P and Q satisfying the elliptic curve equation (2.28). We define $R = (U_3, V_3, W_3)$, the result of the unified addition on $TEd(\mathbb{F}_p)$, $R = P +_{\mathcal{E}} Q$ defined by the following system of equations:*

$$\begin{cases} U_3 = W_1W_2(U_1V_2 + U_2V_1) \\ V_3 = W_1W_2(W_1^2W_2^2 + dU_1U_2V_1V_2)(V_1V_2 - aU_1U_2) \\ W_3 = W_1^4W_2^4 - d^2U_1^2U_2^2V_1^2V_2^2 \end{cases} . \tag{2.29}$$

Cost[8] *: $10M_p + 1S_p + 1C_p^a + 1C_p^d + 7A_p$.*

[8] In order to have this efficiency, use "add-2008-bbjlp" on [BL].

Example Today no curve in the form of Twisted Edwards is standardized by ANSSI, BSI or NIST, however a well-known curve is often used, that of Bernstein named `Curve25519`. On the finite field $\mathbb{F}_p$ with $p = 2^{255} - 19$, the elliptic curve `Curve25519`. The `Curve25519` is a Montgomery curve [9] , where very efficient computations can be carried out using only the X and Z coordinates in jacobian or projective. The `Curve25519` is defined by the equation $y^2 = x^3 + 48662x^2 + x$ is bi-rationally equivalent to the twisted Edwards Curves `Ed25519` defined by equation:

$$-u^2 + v^2 = 1 - \frac{121665}{121666}u^2v^2 \quad . \tag{2.30}$$

This equivalence is given by:

$$\begin{cases} u = \dfrac{x}{y}\sqrt{-48664} \\ v = \dfrac{x-1}{x+1} \end{cases} \quad \text{or} \quad \begin{cases} x = \dfrac{1+v}{1-v} \\ y = \dfrac{1+v}{(1-v)u}\sqrt{-48664} \end{cases} \quad . \tag{2.31}$$

2.4 Modular Multiplications Implementation

Whether it is for the modular exponentiation in RSA scheme or for elliptic curve scalar multiplication, as well as additions and doubling operations, modular arithmetic takes place a main role. In this section, modular arithmetic concerning exclusively modular multiplication is detailed. Given two integers $\mathcal{A}$ and $\mathcal{B}$, the classical modular multiplication $\mathcal{A} \times \mathcal{B} \bmod p$ computes the multiplication $\mathcal{A} \times \mathcal{B}$ followed by the modular reduction by p. To make a modular multiplication, there are several classical methods described in [MvOV96]. The modular multiplication is composed by the multiplication of large numbers and some reduction methods [MvOV96, algorithm 14.28].

1. Remainder of Euclidean division [MvOV96, algorithm 14.20]
2. Montgomery reduction [MvOV96, algorithm 14.32]
3. Barrett reduction [MvOV96, algorithm 14.42]
4. Reduction methods for moduli of special form [MvOV96, Sec. 14.3.4].

2.4.1 Multiplication over large numbers

We call "large numbers" a integer who cannot store over a integer value on a computer, hence the terminology of multi-precision multiplication. In this section, n denotes the number of computer integer length in other terms the number of words used on computer, beware that the symbol n is not the product of pq in RSA context or the order of the generator in ECC context.

[9] The Weierstrass form is $\mathcal{W}(\mathbb{F}_p) : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$. The Montgomery form is $\mathcal{W}(\mathbb{F}_p)$ with $a_1 = a_3 = a_4 = 0$, $a_2 \neq 0$ and $a_6 \neq 0$.

Definition 2.32 (Multiplication of two large numbers [MvOV96, algorithm 14.28]) *Let $N \gg n$. Let $\mathcal{A}$ and $\mathcal{B}$ be two N-bit elements. Then, $\mathcal{A}$ (resp. $\mathcal{B}$) can be written as n words $\mathcal{A}_i$ for all $i \in \{0, 1, \ldots, n-1\}$ (resp. $\mathcal{B}_i$) of $b = \lceil \frac{N}{n} \rceil$-bits. $\mathcal{A}_0$ is the least significant word (LSW) of $\mathcal{A}$ and $\mathcal{A}_{n-1}$ is the most significant word (MSW) of $\mathcal{A}$. Let $\mathcal{X}$ be the result of the multiplication $\mathcal{A} \times \mathcal{B}$ before reduction; $\mathcal{X}$ can be represented by $2n$ b-bit words $(\mathcal{X}_{2n-1}\mathcal{X}_{2n-2} \ldots \mathcal{X}_0)_b$.*

For our experiment (see chapter 4), the elliptic curve cryptography in the library mbedTLS is targeted. The elliptic curves targeted is defined on the finite field is 256-bit long. The micro-controller targeted is Cortex-M4 with 32-bit registers for data operations. Therefore, one field element corresponds to 8 words of 32-bits. In the arithmetic implemented for the elliptic curve cryptography in mbedTLS, a multiplication between two elements in the finite field is computed as described in algorithm 2.6. The result of the multiplication is stored in a 512-bit element, called *"multiplication-before-reduction"*; then the result is reduced modulo p (the characteristic of the finite field).

Algorithm 2.6: Multiplication of two large numbers in mbedTLS

Require: $\mathcal{A}$ and $(\mathcal{B}_7 \ldots \mathcal{B}_0)_{32}$ two 256-bit integers.
Ensure: $\mathcal{X} = \mathcal{A} \times \mathcal{B}$
1: $\mathcal{X} \leftarrow 0$
2: **for** i from 7 down to 0 **do**
3: $(c, \mathcal{X}_{i+7}, \mathcal{X}_{i+6}, \ldots, \mathcal{X}_i) \leftarrow (\mathcal{X}_{i+7}, \ldots, \mathcal{X}_i) + \mathcal{A} \times \mathcal{B}_i$
4: $j \leftarrow i + 8$
5: **repeat**
6: $(c, \mathcal{X}_j) \leftarrow \mathcal{X}_j + c$
7: $j \leftarrow j + 1$
8: **until** $c \neq 0$
9: **end for**
10: return $\mathcal{X}$

The algorithm 2.6 shows how multiplication is performed specially in elliptic curve mathematics in library mbedTLS. The result $\mathcal{A} \times \mathcal{B}_i$ is stored in eight 32-bit words and there is a potential carry c, which needs to be stored separately (see step 3). This potential carry creates a timing overflow when $c \neq 0$.

2.4.2 Euclidean division

Using the algorithm 2.6, we have $\mathcal{X} = \mathcal{A} \times \mathcal{B}$, and after this computation the reduction by p using Euclidean division is made. The Euclidean division is the "school" method to divide an integer $\mathcal{X}$ named dividend by another integer p named divisor. The Euclidean division produced two integers a quotient $\mathcal{Q}$ and a remainder $\mathcal{R}$ such that $\mathcal{X} = \mathcal{Q} \times p + \mathcal{R}$. This remainder $\mathcal{R}$ was also the result of the reduction of $\mathcal{X}$ by the modulo p.

The algorithm 2.7 shows how a reduction by a modulo p is performed for the element $\mathcal{X}$. The value of the $\mathcal{Q}$ is found word by word. The most significant word is computed on the lines 3-6, and the other word in the for-loop beginning at line7. The stop condition in line 13 does not take into account the whole of the $\mathcal{Q}_{i-n-2} \times p$ value but only on 3

Algorithm 2.7: Euclidean division in mbedTLS[MvOV96, algorithm 14.20]

Require: an integer $\mathcal{X} = (\mathcal{X}_{2n-1}, \ldots, \mathcal{X}_0)_b$ and a divisor $p = (p_{n-1}, \ldots, p_0)_b$
Ensure: a quotient $\mathcal{Q} = (\mathcal{Q}_{n-1}, \ldots, \mathcal{Q}_0)_b$, a remainder $\mathcal{R} = (\mathcal{R}_{n-1}, \ldots, \mathcal{R}_0)_b$

$\mathcal{R} \leftarrow \mathcal{X}$
$\mathcal{Q} \leftarrow 0$
while $\mathcal{R} \geq b^{n-1} \times p$ **do**
  $\mathcal{Q}_{n-1} \leftarrow \mathcal{Q}_{n-1} + 1$
  $\mathcal{R} \leftarrow \mathcal{R} - p \times b^{n-1}$
end while
for i from $2n-1$ to n **do**
  if $\mathcal{R}_i = p_{n-1}$ **then**
    $\mathcal{Q}_{i-n-2} \leftarrow b - 1$
  else
    $\mathcal{Q}_{i-n-2} \leftarrow (\mathcal{R}_i \times b + \mathcal{R}_{i-1}) \div p_{n-1}$
  end if
  while $\mathcal{Q}_{i-n-2}(p_{n-1} \times b + p_{n-2}) > \mathcal{R}_i \times b^2 + \mathcal{R}_{i-1} \times b + \mathcal{R}_{i-2}$ **do**
    $\mathcal{Q}_{i-n-2} \leftarrow \mathcal{Q}_{i-n-2} - 1$
    $\mathcal{R} \leftarrow \mathcal{R} - \mathcal{Q}_{i-n-2} \times p \times b^{i-n-2}$
  end while
  if $\mathcal{R} < 0$ **then**
    $\mathcal{R} \leftarrow \mathcal{R} + p \times b^{i-n-2}$
    $\mathcal{Q}_{i-n-2} \leftarrow \mathcal{Q}_{i-n-2} - 1$
  end if
end for
return $\mathcal{Q}, \mathcal{R}$

words, it is possible that the value of $\mathcal{Q}_{i-n-2}$ is overestimated. A second test in line 17 is required to correct this overestimation.

The euclidean reduction method is specially used in elliptic curve mathematics using `brainpoolP256r1` curve in library PolarSSL-mbedTLS.

2.4.3 Montgomery reduction

The Montgomery reduction is an efficient algorithm to compute the modular multiplications introduced in 1985 by Peter L. Montgomery [Mon85]. This method is very efficient when a huge number of modular multiplications with the same modulo have to be performed. Thus, it is particularly adjusted to the modular exponentiation or scalar multiplication computations. Given two integers $\mathcal{A}$ and $\mathcal{B}$, the classical modular multiplication $\mathcal{A} \times \mathcal{B} \bmod p$ computes the multiplication $\mathcal{A} \times \mathcal{B}$ followed by the modular reduction by p. Montgomery Modular Multiplication (MMM) transforms $\mathcal{A}$ and $\mathcal{B}$ into special representations known as their Montgomery forms.

Definition 2.33 (Montgomery Transformation [Mon85]) *For any prime modulus p, the Montgomery form of $\mathcal{A} \in \mathbb{F}_p$ is $\phi(\mathcal{A}) = \mathcal{A} \times R \bmod p$ for some constant R greater than and co-prime with p.*

In order to ease the computation, R is usually chosen as the smallest power of b greater than p, that is $R = b^{\lceil \log_b(p) \rceil}$. In our experiment, R is the smallest power of 2 greater than p, that is $R = 2^{\lceil \log_2(p) \rceil}$. Using the Montgomery form of integers, modular multiplications used in modular exponentiation computation can be carried out using the Montgomery Modular Multiplication (MMM):

Proposition 2.34 (Correction of MMM [MvOV96, Sec. 14.3.2]) *If the Modular Multiplication is $\mathcal{C} = \mathcal{A} \times \mathcal{B} \bmod p$, then the Montgomery Modular Multiplication is:*

$$\phi(\mathcal{C}) = \phi(\mathcal{A})\phi(\mathcal{B})R^{-1} \bmod p \ . \tag{2.32}$$

Proof The proof comes from [Mon85], but is briefly recalled here for completeness:

$$\begin{aligned}
\phi(\mathcal{A})\phi(\mathcal{B})R^{-1} \bmod p &= \mathcal{A} \times R \times \mathcal{B} \times R \times R^{-1} \bmod p && \text{by definition 2.33,} \\
&= \mathcal{A} \times \mathcal{B} \times R \times (R \times R^{-1}) \bmod p && \text{by commutativity,} \\
&= (\mathcal{A} \times \mathcal{B}) \times R \bmod p && RR^{-1} = 1 \bmod p, \\
&= \phi(\mathcal{A} \times \mathcal{B}) = \phi(\mathcal{C}) && \text{by definition 2.33.}
\end{aligned}$$

□

The MMM can be implemented in two steps:

(i) compute $D = \phi(\mathcal{A}) \times \phi(\mathcal{B})$ using the algorithm 2.6, then

(ii) reduce D using Montgomery reduction defined by algorithm 2.8, which returns $\phi(\mathcal{C})$.

In algorithm 2.8, the pair (R^{-1}, v) is such that $RR^{-1} - vp = 1$. The modular reduction (respectively division $\div$) by R is the bit-wise AND (resp. right-shift) of the binary representation.

Algorithm 2.8: Montgomery Reduction [MvOV96, algorithm 14.32]

Require: $D = \phi(\mathcal{A}) \times \phi(\mathcal{B}), R, R^{-1}, v, p$ such as $RR^{-1} - vp = 1$
Ensure: $\phi(\mathcal{C}) = \phi(\mathcal{A}) \times \phi(\mathcal{B}) \times R^{-1} \bmod p$
1: $m \leftarrow (D \bmod R) \times v \bmod R$
2: $U \leftarrow (D + m \times p) \div R$ ▷ Invariant: $0 \leq U < 2p$
3: **if** $U \geq p$ **then**
4: $\quad C \leftarrow U - p$ ▷ Extra-reduction
5: **else**
6: $\quad C \leftarrow U$
7: **end if**
8: **return** C

Definition 2.35 (Extra-Reduction [WT01]) *In algorithm 2.8, when the intermediate value U is greater than p, a subtraction named eXtra-reduction occurs so as to have a result C of the Montgomery multiplication between 0 and $p-1$. We set $X = 1$ in the presence of the eXtra-reduction, and $X = 0$ in its absence.*

Most software implementations of modular arithmetic for large numbers (such as OpenSSL and mbedTLS) use the MMM, where there is a final conditional extra-reduction.

2.4.4 Barrett Reduction

The Barrett Reduction permits to improve the time of the modular reduction, with a pre-computation of an integer noted μ. The modular multiplication using the Barrett reduction is the computation of $\mathcal{X} = \mathcal{A} \times \mathcal{B}$ using the algorithm 2.6 and after the Barrett reduction described in the algorithm 2.9. As the Montgomery reduction, this method is very useful when the number of modular reductions is huge. The pre-computation of the integer μ can be computed once for each modulus and stored in the embedded device. Typically, μ equals $\left\lfloor \frac{b^{2n}}{p} \right\rfloor$, with b the word-size of the processor and n the length of the modulus in b-word.

The divisions $\div$ in lines 1-3 (respectively the modular reduction *mod* in lines 4-5) by b^{n-1} is the right-shift (resp. left-shift) of the b representation.

2.4.5 Special moduli reduction

When the modulo has a special form, the modular reduction can be performed using only shift and addition. In [MvOV96, Sec. 14.3.4], the case with a modulo $p = b^n - r$ with "small" r was explained. The main use case is when the NIST curves are employed or the Bernstein curve `Ed25519`. This trick can be used to perform the modular reduction when a NIST or Bernstein curves are chosen. In the RSA scheme, this method cannot be used with random moduli.

Algorithm 2.9: Barrett Reduction [MvOV96, algorithm 14.42]

Require: $\mathcal{X} = (\mathcal{X}_{2n-1}, \ldots, \mathcal{X}_0)_b, \mu = \left\lfloor \frac{b^{2n}}{p} \right\rfloor, p = (p_{n-1}, \ldots, p_0)_b$
Ensure: $\mathcal{C} = \mathcal{X} \bmod p$
1: $Q_1 \leftarrow \mathcal{X} \div b^{n-1}$
2: $Q_2 \leftarrow Q_1 \times \mu$
3: $Q_3 \leftarrow Q_2 \div b^{n-1}$
4: $R_1 \leftarrow \mathcal{X} \bmod b^{n-1}$
5: $R_2 \leftarrow Q_3 \times p \bmod b^{n-1}$
6: $\mathcal{C} \leftarrow R_1 - R_2$
7: **if** $\mathcal{C} < 0$ **then**
8: $\quad \mathcal{C} \leftarrow \mathcal{C} + b^{n-1}$
9: **end if**
10: **while** $\mathcal{C} \geq p$ **do**
11: $\quad \mathcal{C} \leftarrow \mathcal{C} - p$
12: **end while**
13: **return** $\mathcal{C}$

2.5 Conclusion

In this chapter, we have defined the mathematical background necessary to my researches. These definitions and properties define the form of two pyramid schemes (figure 2.1(a)-(b)). The top of the pyramids is the cryptography protocol, signature, authentication, secret key exchange, the public and secret keys generated, and the cryptography domain for the elliptic curve. The sensitive operation in RSA is the modular exponentiation and in ECC is the scalar multiplication. In ECC, the group law composed by the addition and doubling operation is required, but this group law is based on modular arithmetic. The base of two pyramid schemes is the modular arithmetic and the hardware device. The device defines how to implement the modular arithmetic with the multiplication described previously and the modular reduction in the multiplication operation.

—

Chapter 3

Side Channel Attacks on asymmetric cryptography

Since 1996, side channel attacks have challenged the security level of cryptography device on embedded device. This chapter focuses on the state-of-the-art of side channel and fault attacks against asymmetric cryptography. There exists a large number of publications on this subject; we have made a selection to present the previous works, who contributed in our research process. On the figure 3.1, we can observe in red color, the side channel and fault attacks, which can applied to fail the security level of each pyramid parts. The green color represents the protections, which contributed to protect of each part of pyramid scheme.

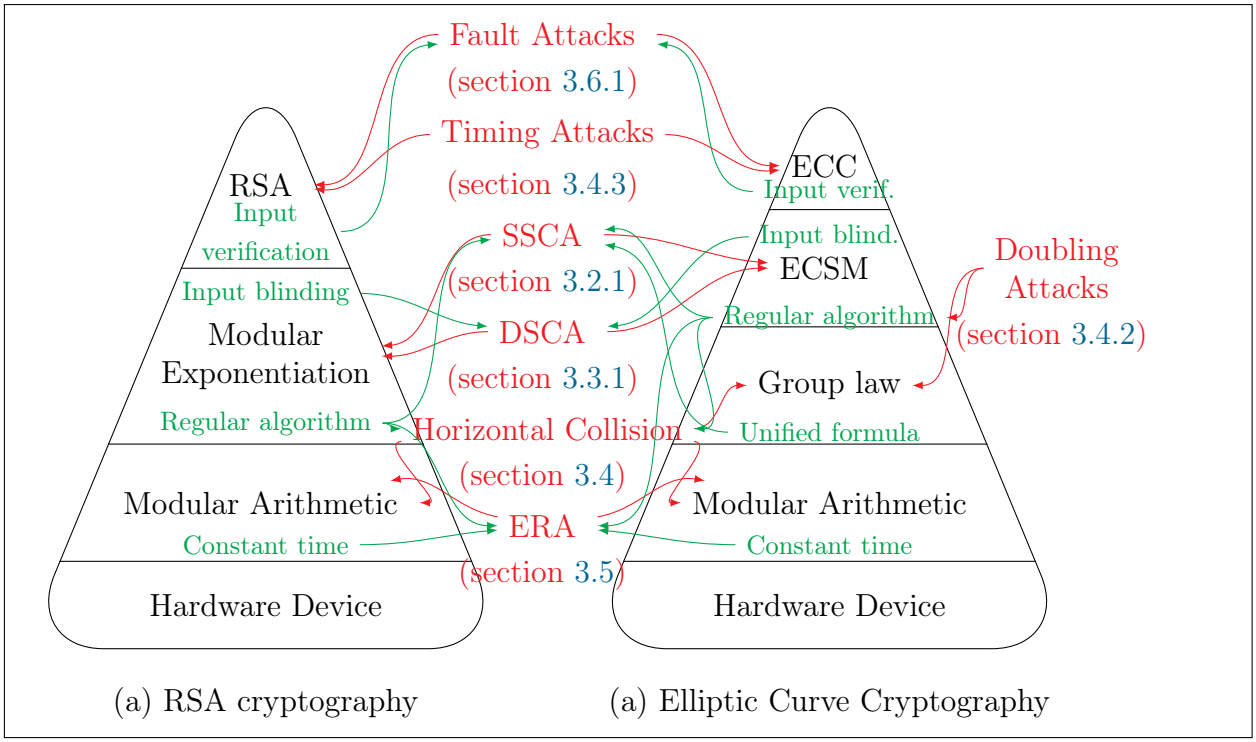


Figure 3.1: Global view of state-of-the-art of attacks and protections required for this thesis

Contents

3.1 Introduction

Cryptography makes it possible to secure sensitive data. The mathematical proofs of cryptography guarantee a high security level of sensitive data. Yet in real world the security levels are oblivious to the implementation on the cryptosystem on a physical device such as computer, smart-phone, smart-card, etc. In fact, a concrete device possesses physical properties: It consumes power current, it produces electromagnetic, photonic emanation and the computation takes time. These physical properties can be observed or modified using specialized equipment. There exist two kinds of attacks: the passive attacks and the active attacks.

passive attacks. Passive attacks are an observation of the computation execution. They can differentiate according to the type of physical property of the component from which it takes advantage. There exist three main side channels observations. Time: the device takes time to make a critical computation; a wise attacker can take advantage of time difference observations. Power: The electrical component is composed by transistor to construct logical gates. Each transistor works like an interrupter. Its power consumption depends on its state: open, close or switching values. Each consumption of these states is few different and this bias can be exploited. In fact a wise attacker can retrieve information of the internal state of the computation. EM: Attacks using electromagnetic emissions of a component are based on the fact that low current charges which are in motion produce a magnetic field which itself produces an electric field.

The first step on a side channel attack is the acquisition step. The current consumption as well as the electromagnetic radiation of a device can be observed using probe and digitalized by an oscilloscope. Figure 3.2 represents a classical side channel lab.

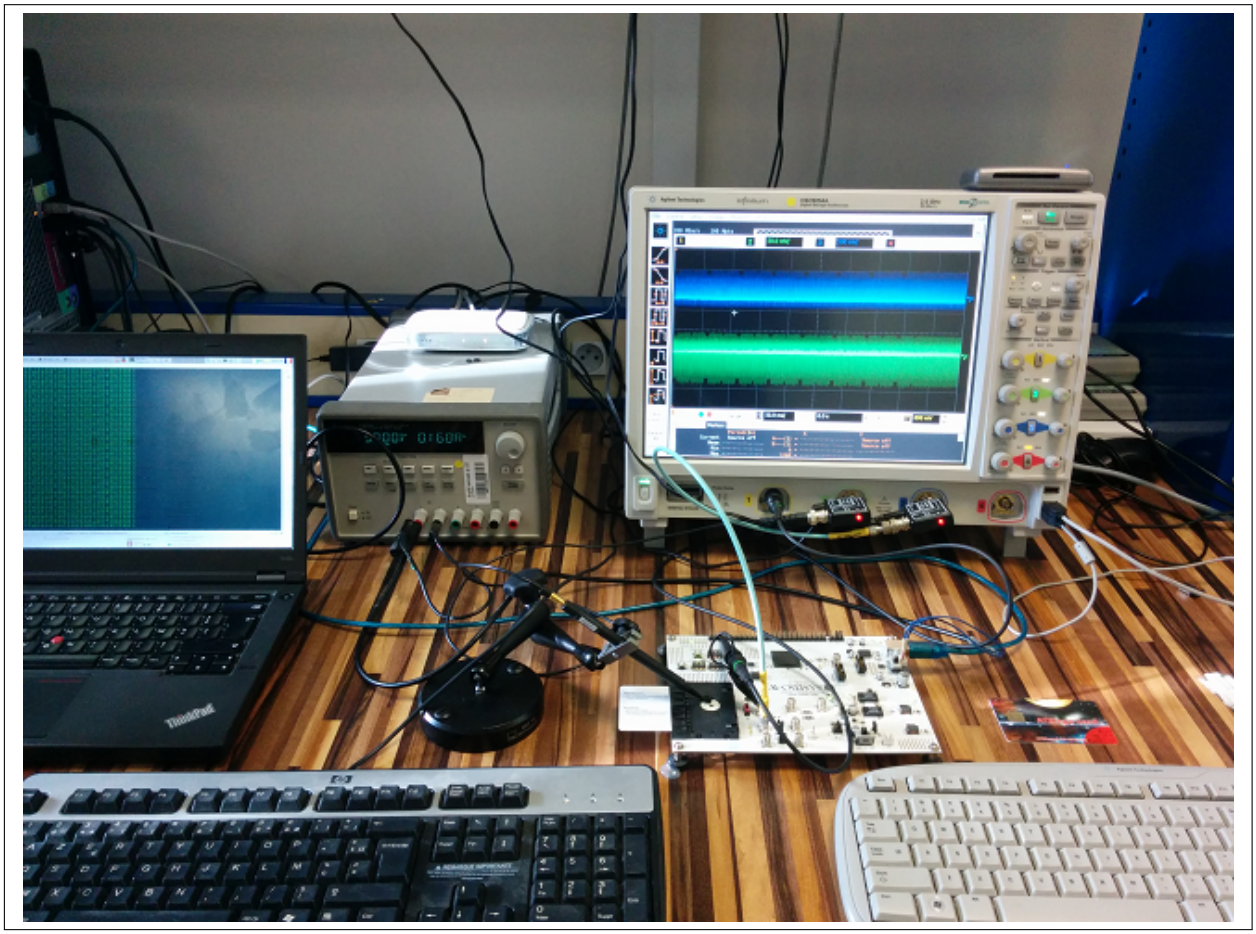


Figure 3.2: Specialized equipment for a side channel acquisition set up

The computer is required to control the oscilloscope acquisition and the communication with the device. An external power supply enables to stabilize the intensity and voltage with more precision. A current probe allows to acquire the global power consumption of the device. An EM probe serves to observe the electromagnetic emanation.

The second step is the pre-processing of the acquisition traces. All method of pre-

processing is not detailed in this thesis. But for example it can be filtering method or transform the signal in spectral domain. The pre-processing of the acquisition set is also the alignment of the acquisitions. In most of cases, the signal information between two acquisitions must be synchronized to make an analysis.

The third step is the analysis of the acquisition. There different kind of analysis described in this chapter.

active attacks. Actives attacks modify the device by its comportment of the secure computation or its packaging protections. There exist three kind of active attacks: invasive, semi-invasive and non-invasive. In this thesis, we focus only on fault injection attacks. Generally they require a preparation to eliminate the packaging protections to make a fault injection. A fault can be realized using an electromagnetic injection or laser injection. Fault attacks modify the comportment or a value during the critical computation. So, the fault attacks are semi-invasive attack when they do not destroy the component. A side channel observation can be made to find the position and the time samples to inject fault. So, the classical equipment for side channel attacks is required: oscilloscope, current probe or EM probe. But other specialized equipment is required for the fault injection: a laser or an EM injection probe.

The rest of this chapter is organized as follows: In section 3.2, we present the simple side channel attacks against modular exponentiation and elliptic curve scalar multiplication using one acquisition and the main protections against these simple attacks. The second classical attacks are a vertical analysis using more acquisitions and name the differential side channel attacks presented in section 3.3. My first attack presented in this thesis is against elliptic curve scalar multiplication with a regular and blinding exponent. This attack is an horizontal attack, in section 3.4 we presented the state-of-the-art for introducing this attack. My second work presented in this thesis is a new analysis on the extra-reduction operation in Montgomery Multiplication. In the section 3.5, all the previous work on extra-reduction analyzed are classified in three categories. To finish, in section 3.6 presents a quick state-of-the-art of fault attacks and its protections in order to introduce my thesis work on the protection on elliptic curve against fault attacks.

3.2 Simple Side Channel Analysis

In 1999, Paul C. Kocher, Joshua Jaffe and Benjamin Jun [KJJ99] propose a new cryptanalysis exploited the side channel. Their first idea is using the power consumption to retrieve secret data. This analysis names the Simple Power Analysis (SPA). While in 2001, Karine Gandolfi, Christophe Mourtel and Francis Olivier [GMO01] propose to exploit the electromagnetic emanation. This analysis names Simple ElectroMagnetic Analysis (SEMA). More generally, these attacks are Simple Side Channel Analysis (SSCA). The cryptography implementation are vulnerable to SSCA, if the implementation uses conditional branch in function of the secret bit and if acquisition shows different patterns in function of secret values. This attack can be applied at different level of the computation.

3.2.1 Attack

For the modular exponentiation in RSA and the elliptic curves scalar multiplication in ECC, the classical simple side channel attack is found a difference between a square and a multiplication in RSA and in a doubling and adding operation in ECC. Generally, this attack can be applied when any protection are used, e.g. using directly the algorithms 2.1-2.5-2.4-2.2. These algorithms are non-regular, because their execution depends on the bit value of exponent or scalar. If the bit value is "0", then the execution flow includes one operation (square or doubling). If the bit value is "1", then the execution flow includes two operations (square and multiply or doubling and adding).

Using only one acquisition, an attacker can observed the sequence of operations like presented in figure 3.3 for RSA.

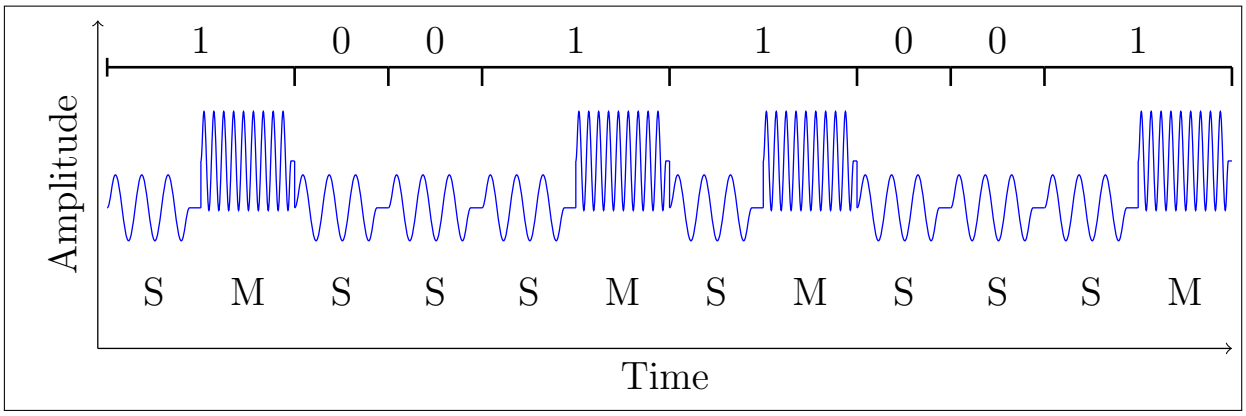


Figure 3.3: Principle of simple side channel analysis on RSA

To realize a simple attack on an acquisition, the first step is to be able to cut operation. The second step is finding the difference between each operation. The main differences are the time and the signal amplitude. In particular case on ECC, the sequence of modular arithmetic operations can be distinguishable.

On the figure 3.3, we observe two patterns. We have two smallest consecutive patterns and never two consecutive biggest patterns. So the identification of the two patterns is: the smallest amplitude pattern is the square operation denoted by an "S", and the other pattern is the multiply operation denoted by an "M". Using the sequence of square and multiply operation we retrieve each bit value of exponent.

Example We take an acquisition using an EM probe on cortex-M4. The implementation was a classical double-and-add L2R algorithm 2.4 using the affine operation described in PolarSSL for the elliptic curves groups law. The particularities of this implementation are:

- there is an modular inversion on each elliptic curve operations, and
- the number of modular multiplication in doubling and adding operation are different.

Figure 3.4 is the acquisition on elliptic curve scalar multiplication on the `brainpoolP256r1` .

Each elliptic curve operations are splitting using the modular inversion presented on each operation. In PolarSSL, the number of modular multiplications in adding operation

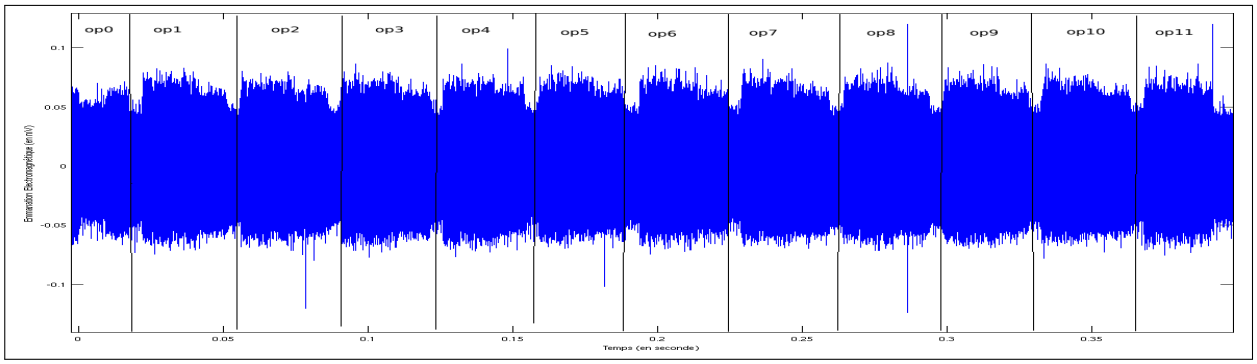


Figure 3.4: EM acquisition on elliptic curve scalar multiplication execution on `brainpoolP256r1` with affine coordinates

in affine coordinate are 10 and in doubling in affine coordinate, there are 11 modular multiplications. The modular multiplication operation is recognized by a specific pattern encircled in figure 3.5.

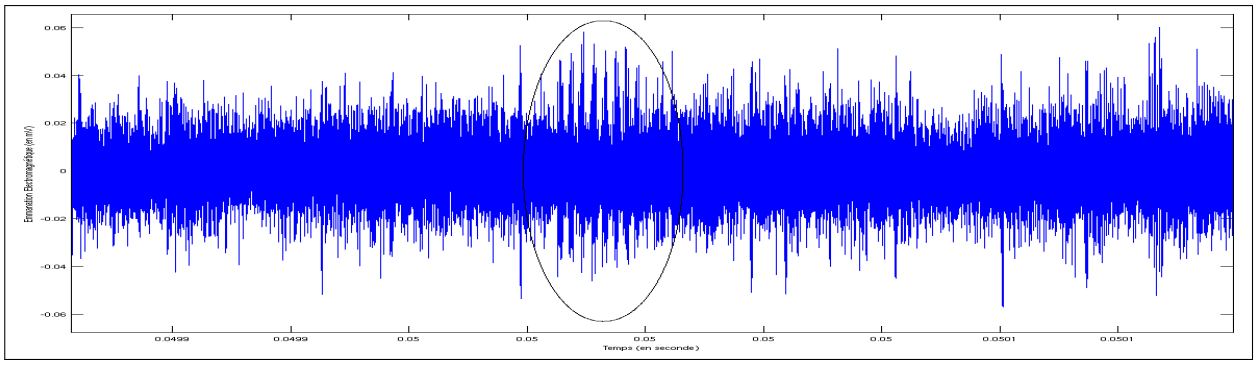


Figure 3.5: Zoom on EM acquisition on elliptic curve operation to spot the modular multiplication operation

By counting the number of multiplications in EM acquisitions, the doubling operation and adding operation are recognized. The scalar is retrieved with the writing of the operation sequences as suggest by figure 3.3.

Remark In order to amplify the difference between two operations, an attacker can chose specific inputs like described in [AT03, CFR10]. In [AT03], Toru Akishita and Tsuyoshi Takagi propose to exploit the leakage observed using chosen input point on elliptic curve. Their attack consists to use a intermediate point value with a coordinate equals to zero, to observe the power leakage of the operation employing zero value. In [CFR10], Jean-Christophe Courrège, Benoit Feix and Mylène Roussellet propose an attack with a chosen message on RSA exploited the same leakage presented in [AT03]. Their analysis is based on a power leakage of the device using words which have zero hamming weight.

3.2.2 Extra-information on simple side channel techniques

There exist some mathematical techniques to identify the difference between two operations when the human eyes are not sufficient.

With a learning phase, an attacker can construct two sets of acquisitions with the operation knowledge. Either using the public operation with the public keys known, or using the initialization phase of the computation, or with a full control on the device. These two sets can be composed by one or several acquisitions. The question is to which acquisition sets looks the most with the unknown acquisition operation. Two kinds of techniques are used in my thesis described by the two following definitions.

Definition 3.1 (Minimum of absolute difference) *Let T_a and T_b two sets of Q acquisitions with S time samples for the value a and b. The acquisition $t_x(q,s)$ is the amplitude of the time sample value s of the acquisition q in he set T_x The acquisition $\mathcal{C}$ of S time samples is more likely to belong to the set whose euclidean difference between this acquisition $\mathcal{C}$ and each acquisition of the set is the smallest. In other terms, the acquisition $\mathcal{C}$ has more probability to be in set T_x with $\min_{x\in\{a,b\}}(\sum_{s=1}^{S}\sum_{q=1}^{Q}|\mathcal{C}s - t_x(q,s)|$ with $\mathcal{C}(s)$ the amplitude of the acquisition $\mathcal{C}$ at s time sample value.*

Definition 3.2 (Maximum Correlation techniques) *Let T_a and T_b two sets of Q acquisitions with S time samples for the value a and b. The acquisition $t_x(q,s)$ is the amplitude of the time sample value s of the acquisition q in he set T_x The acquisition $\mathcal{C}$ of S time samples is more likely to belong to the set whose Pearson correlation coefficient between this acquisition $\mathcal{C}$ and the mean acquisition of the set is the biggest. In other terms, the acquisition $\mathcal{C}$ has more probability to be in set T_x with $\max_{x\in\{a,b\}}\rho(\mathcal{C},\mathcal{M}_x)$, with :*

- *the means of each set for one sample s are $\mathcal{M}_x^{(s)} = \sum_{q=1}^{Q}\frac{t_x(q,s)}{Q}$,*
- *the Pearson coefficient of two vector variables X and Y are defined by the formula $\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\mathrm{Var}(X)\mathrm{Var}(Y)}$.*

Other techniques like [TK01, HTM11, LMV+13, HIM+13, PITM14] allow to classified the operation acquisitions with and without a learning phase.

3.2.3 Countermeasures

The main protection against this simple side channel attack is to use a regular algorithm for modular exponentiation and elliptic curves scalar multiplication. Another protection is the non-distinguishable operations.

Definition 3.3 (Binary regular algorithm) *In a regular algorithm, there exists no difference in the execution flow depending on the bit value of the exponent (or scalar).*

The modular exponentiation algorithm can be made in binary regular algorithm. The most popular algorithm is the "square-and-multiply-always" described in the left-to-right version in algorithm 3.1. The version right-to-left exists also but it is not described here. Another popular method is the "Montgomery Ladder" described in algorithm 3.2. A version right-to-left presented by Joye in [Joy03] exists also.

Algorithm 3.1: Square and Multiply Always Left-to-Right

Require: $m, k = (k_l k_{l-1} \dots k_0)_2, p$ $(k_l = 1)$
Ensure: $m^k \bmod p$
1: $R_0 \leftarrow 1$
2: $R_1 \leftarrow m$
3: **for** $i = l - 1$ **downto** 0 **do**
4: $\quad R_1 \leftarrow R_1 \times R_1 \bmod p$ ▷ S_i
5: $\quad R_{k_i} \leftarrow R_1 \times m \bmod p$ ▷ M_i
6: **end for**
7: **return** R_1

Algorithm 3.2: Montgomery Ladder Left-to-Right

Require: $m, k = (k_l k_{l-1} \dots k_0)_2, p$ $(k_l = 1)$
Ensure: $m^k \bmod p$
1: $R_0 \leftarrow m$
2: $R_1 \leftarrow R_0 \times R_0 \bmod p$ ▷ FS
3: **for** $i = l - 1$ **downto** 0 **do**
4: $\quad R_{\neg k_i} \leftarrow R_0 \times R_1 \bmod p$ ▷ M_i
5: $\quad R_{k_i} \leftarrow R_{k_i} \times R_{k_i} \bmod p$ ▷ S_i
6: **end for**
7: **return** R_0

The cost of the algorithms 3.1 and 3.2 is l square operations and l multiply operations. The computational overhead of this countermeasure compared to algorithms 2.1 or 2.2 is $l/2$ multiply operations.

Like the modular exponentiation, the elliptic curve scalar multiplication can be regular algorithm. The double-and-add-always procedure is described in algorithm 3.3 and the Montgomery Ladder algorithm in algorithm 3.4.

Algorithm 3.3: double-and-add-always Left-to-Right

Require: $\mathcal{E}(\mathbb{F}_p), P, k = (k_l k_{l-1} \dots k_0)_2$ $(k_l = 1)$
Ensure: $[k]P$
1: $R_0 \leftarrow \mathcal{O}_\mathcal{E}$
2: $R_1 \leftarrow P$
3: **for** $i = l - 1$ **downto** 0 **do**
4: $\quad R_1 \leftarrow R_1 +_\mathcal{E} R_1$ ▷ $ECDBL_i$
5: $\quad R_{k_i} \leftarrow R_1 +_\mathcal{E} P$ ▷ $ECADD_i$
6: **end for**
7: **return** R_1

Algorithm 3.4: ECSM Montgomery Ladder Left-to-Right

Require: $\mathcal{E}(\mathbb{F}_p), P, k = (k_l k_{l-1} \dots k_0)_2$ $(k_l = 1)$
Ensure: $[k]P$
1: $R_0 \leftarrow P$
2: $R_1 \leftarrow R_0 +_\mathcal{E} R_0$ ▷ $First - ECDBL$
3: **for** $i = l - 1$ **downto** 0 **do**
4: $\quad R_{\neg k_i} \leftarrow R_0 +_\mathcal{E} R_1$ ▷ $ECADD_i$
5: $\quad R_{k_i} \leftarrow R_{k_i} +_\mathcal{E} R_{k_i}$ ▷ $ECDBL_i$
6: **end for**
7: **return** R_0

The cost of the algorithms 3.3 and 3.4 is l elliptic curve adding operations and l elliptic curve doubling operations. The computational overhead of this countermeasure compared to algorithms 2.4 or 2.5 is $l/2$ adding operations.

Another protection is to make two operations non-distinguishable. In RSA, the developer must use the same implementation for a square and a multiplication, he cannot used a trick to compute more efficiently the square computation. In ECC, the developer must use the same formula for doubling and adding noted ECADD-unified. The elliptic curve operations are unified like Edwards and Twisted Edwards curves presented in section 2.3.3. The modular exponentiation and the elliptic curve scalar multiplication algorithms can be adapted for unified formula.

Algorithm 3.5: Multiply-always Left-to-Right

Require: $m, k = (k_l k_{l-1} \dots k_0)_2, p$ $(k_l = 1)$
Ensure: $m^k \bmod p$
1: $R_0 \leftarrow m$
2: $R_1 \leftarrow m$
3: $j \leftarrow l - 2$
4: $b \leftarrow$
5: **while** $j \geq 0$ **do**
6: $\quad R_0 \leftarrow R_0 \times R_b \mod p$
7: $\quad b \leftarrow b \oplus k_j$
8: $\quad j \leftarrow j + k_j - 1$
9: **end while**
10: **Return** R_0

Algorithm 3.6: Add-always Left-to-Right

Require: $\mathcal{E}(\mathbb{F}_p), P, k = (k_l k_{l-1} \dots k_0)_2$ $(k_l = 1)$
Ensure: $[k]P$
1: $R_0 \leftarrow P$
2: $R_1 \leftarrow P$
3: $j \leftarrow l - 2$
4: $b \leftarrow$
5: **while** $j \geq 0$ **do**
6: $\quad R_0 \leftarrow ECADD - unified(R_0, R_b)$
7: $\quad b \leftarrow b \oplus k_j$
8: $\quad j \leftarrow j + k_j - 1$
9: **end while**
10: **Return** R_0

The cost of the algorithms 3.5 and 3.6 depending of the hamming weight of the exponent or scalar k noted $HW(k)$. The cost is $l + HW(k)$ square operation and $l + HW(k)$ multiply operations. The computational overhead of this countermeasure depends of the computational overhead of the multiplication compared to square operation for RSA. For ECC, the computational overhead of this countermeasure is the computational overhead of the unified addition compared to the addition and doubling operations.

3.3 Differential Side Channel Analyses

The differential side channel analyses exploit the dependency between the manipulated data and the side channel observations. The classification of side channel analyses is under standardization at ISO: for instance, *horizontal* vs. *vertical* and *univariate* vs. *multivariate* is defined in figure 2 of ISO 20085-1 [ISO17]. In this section, we present classical vertical attacks including template attacks and in the following section examples of horizontal attacks are detailed.

3.3.1 Non supervised attacks

Firstly, the classical vertical attacks use a leakage model of an intermediate value and a distinguisher. For example, the leakage model can be a value manipulated by the processor, the hamming weight of the value, or the hamming distance between two intermediate values. For the Differential Power Analysis (DPA) presented by Kocher et al. [KJJ99], the distinguisher is the difference of the means of groups. The acquisition set is classified in groups depending of the leakage model of the intermediate value targeted for each bit hypotheses values. The DPA traces for one hypothesis is the differential trace between the mean traces of the two groups. If the groups are well classified, then the mean of each group contains signal information depending of its group and the difference of the two means are not equal to zero. If the groups are not classified, then the mean of each group does not contain signal information and the difference of two means must give zero. Generally, the DPA requires a large numbers of acquisitions ($\simeq 10^4$ or 10^6)

Another classical distinguisher is the coefficient of Pearson correlation [BCO04]. The attack is also named correlation power analysis (CPA). This method shows the linearity between the leakage model of the targeted intermediate value and the signal acquisition. The number of acquisitions require for CPA is less than the number of acquisitions for a DPA.

There exist other distinguishers like Spearman coefficient, mutual information, etc. But these distinguishers are less used in practice.

3.3.2 Template attacks

Secondly, template Attacks belong to yet another kind of attacks and are considered to be the most powerful method from the information-theoretic point of view, since they take advantage of more information available in a side channel observation [CRR02]. The attacker is assumed to have one or limited number of side channel measurements from the target device, i.e. power, EM traces or timing, but he has access to a similar device, on which he can simulate the computations of the target (template building phase). Rechberger and Oswald presented the first practical template attack on RC4 running on an 8-bit micro-controller in [RO04]. The template attacks are made in two steps: the building phase and the matching phase. Using a device with a full control, the attacker constructs a template set of the acquisitions. This set is a dictionary of the acquisitions, constructed for each hypothesis of the secret and each input value. This is the building phase. The matching phase consists on acquiring a set with an input value known and the secret value unknown. We compare the set of acquisitions with the template acquisition set to find the right secret value. For ECC and RSA, the building phase of template is the same as described in MESD attack [MDS99].

The main drawback of template attacks is the assumption concerning the similar device with a full control. In the real world, this assumption can be difficult to realize, the private key are fixed and cannot be chosen by an attacker.

3.3.3 Countermeasures in RSA

The main protection against the differential side channel attacks was the input blinding. For RSA, the inputs of the modular exponentiation are the two integers values message m and private exponent d with the modulo $n = pq$. The public exponent is e.

Classical blinding exponent method

The different ways of scalar randomization are:

1. exponent blinding method: the integer $d \leftarrow d + r\varphi(n)$, where r is a small (e.g., 64-bit) uniformly distributed random number [Koc96, section 10, page 9]. This blinding method is impossible without the knowledge of the two prime factors p and q since $\varphi(n) = (p-1)(q-1) = n + 1 - p - q$ cannot be computed.

2. exponent splitting method: $m^d \mod n \leftarrow m^{(d-r)} \mod n \times m^r \mod n$, where r is a small (e.g., 64 bit) uniformly distributed random number

Message blinding

The method used in order to blind a message m in RSA is :

1. Select a random number r,
2. Compute the modular exponentiation with the public exponent $r^e \mod n$,
3. Multiply m by $r^e \mod n$ to obtain m^*,
4. Use m^* as input of the modular exponentiation with exponent d and the modulo n, the output is c^*,
5. Divide the result of the modular exponentiation c^* by integer r.

3.3.4 Countermeasures in ECC

For ECC, we denote by P the input point of the elliptic curve scalar multiplication ECSM on $\mathcal{E}$, k the input scalar of the ECSM and n is the order of the generator point of $\mathcal{E}$ defined definition 2.8.

Classical blinding scalar method

The different ways of scalar randomization are:

1. $[k]P = [k - r]P + [r]P$, two scalar multiplications are computed $Q = [k - r]P$ and $R = [r]P$;
2. $[k]P = [k \times r^{-1}]([r]P)$, two scalar multiplications are computed $Q = [r]P$ and $R = [k \times r^{-1}]Q$;
3. $[k]P = [k \mod r]P + [\lfloor \frac{k}{r} \rfloor]([r]P)$, three scalar multiplications are computed $Q = [k \mod r]P$, $R = [r]P$ and $S = [\lfloor \frac{k}{r} \rfloor]R$.
4. $[k]P = [k']P$, where n is the order of the curve and $k' = k + nr$ is the randomized scalar. Thus, $[rn]P = \mathcal{O}_{\mathcal{E}}$ the neutral element on $\mathcal{E}$.

Point blinding

For the point blinding, two different ways exist.

1. The most popular randomizes the coordinate, it is not possible for affine coordinate, but this is possible for other point representation e.g. in projective using the proposition 2.23 and in jacobian using the proposition 2.14. The cost of this method is 4 or 5 modular multiplications; it is negligible compared to an ECSM operation.
2. A second method is to use a random point R, compute the ECSM of R by the scalar $-k \mod n$, add this point result to the input P, and use this result as input of the ECSM. In fact we compute $[k]P = [k](P + [-k \mod n]R)$. This method requires two ECSM computations.

3.4 Horizontal Analysis

The execution time of an asymmetric computation is longer than symmetric algorithm. The vertical analysis described in previous section can be difficult to realize, due to the very large number of acquisitions required, to the alignment step and the computation power to process. The horizontal analysis is more efficient to attack asymmetric cryptography. In this section, we presents only two kinds of attacks that we will use in the chapter 4 and the timing attack.

3.4.1 Horizontal attacks

Horizontal attack [CFG+12, BJPW13, BJP+15] allows to attack modular exponentiation or elliptic curve scalar multiplication with only one acquisition, when the formula is unified. There are no difference of implementation between a square and a multiply in RSA and between adding and doubling in ECC. When adding and doubling are unified, there are no difference between the implementation of an addition of two difference point and an addition of the same point. The addition is composed of modular operations like modular multiplications. So, in order to make an horizontal attack on ECC using unified formula or on RSA, an attacker wants to find a difference between a square and a multiplication operations.

As described in section 2.4, the modular multiplication need a multiplication over large numbers. This operation is composed by several smaller multiplications on the processor named "little multiplication" (see line 3 in algorithm 2.6). For example, by observing these "little multiplication" when $\mathcal{A}_i \times \mathcal{B}_j$ is manipulated and $\mathcal{A}_j \times \mathcal{B}_i$ is manipulated for $i \neq j$, we can find if $\mathcal{A} = \mathcal{B}$ or not. Another example we can observe only the "little multiplication" when $\mathcal{A}_i \times \mathcal{B}_i$ for each i. In fact the leakage of the power consumption can be dependent of the hamming weight of the register value. The hamming weight of the square is different than the hamming weight of a multiplication, we are able to find if $\mathcal{A} = \mathcal{B}$ or not.

Remark *Advantages.* The main advantages are only one acquisition is required and all the input blinding protection methods described in section 3.3.3 are ineffective against these attacks.

Drawbacks. In practice, theses attacks are very strong to realize due to re-synchronization of each "little multiplication". Theses attacks can be applied only using the algorithm "multiply-always" (algorithm 3.5) for RSA and "add-always" (algorithm 3.6) for ECC. The protection by using a regular algorithm (see definition 3.3) allows to protection against these attacks.

3.4.2 Collision on Doubling operation

The "Doubling attack" described by Pierre-Alain Fouque and Fédéric Valette [FV03] is a collision side channel analysis on the doubling operation. This attack permits to defeat a regular ECSM algorithm (definition 3.3) like "double and add always" (algorithm 3.3). An attacker does not know if the computation execution is a real addition or a dummy addition. More precisely, an attacker must be able to compare two doubling operations $ECDBL([m_1]P)$ and $ECDBL([m_2]P)$ and decides if $m_1 = m_2$ or not.

An attacker takes one acquisition with the input point P and another acquisition with the input point $2P$. On the both acquisitions, the doubling operation is targeted at each iteration i (for the iteration loop of i, we have $l-1 \leq i > 0$ where i is decremented after each iteration). If a collision between the doubling operation i on the acquisition with P is made with the doubling $i+1$ on the acquisition with $2P$, then the bit value k_i equals "0", otherwise $k_i = 1$.

Remark This attack can be applied on RSA scheme. The method is used the message m and its squaring value m^2 and made the collision with the square operation. *Advantages.* The main advantage of these attacks is the number of acquisition only two. This attack works on regular algorithm (see definition 3.3). In ECDH protocol choose the input point of the algorithm is possible.

Drawbacks. In most use case, the input point of the ECSM algorithm is in affine coordinate and the computational intermediate values are in jacobian or projective, in order to avoid the modular inversion. To achieve a collision between the acquisition of P and that of $2P$, we must have the same intermediate values. If the intermediate values are in projective or jacobian, generally the Z-coordinate does not equal 1. The affine representation is unique, but for projective and jacobian coordinates this is not the case. If the input value is necessarily in affine coordinate, this attack cannot be possible. Otherwise a protection against this attack is the input point randomization (see point blinding in section 3.3.4).

3.4.3 Timing execution attacks

Timing execution attacks exploit the time difference of an execution with one input compared to another input. The timing variation can be in local operation and observe using a EM probe or current probe or the global timing can be studied.

The first timing attack against RSA is presented by Kocher [Koc96]. This attack is applied on RSA algorithm without protection and with a known modulus. Using the CRT method or one protection input protection this attack could not work.

The main protection against time attack is to use a constant time implementation. The constant time protection can be applied at each level of the pyramid scheme. The main consequence is the alignment of different acquisitions can be easier.

3.5 Extra-Reductions Analysis

The extra-reduction is the final subtraction at the end of the Montgomery reduction as defined in definition 2.35. The integer p is the modulo of the modular multiplication. The integer R is the Montgomery constant defined by $R = 2^{\lceil \log_2(p) \rceil}$. This section reviews known results about Montgomery extra-reductions. In this thesis we focus on the Montgomery Modular Multiplication(MMM), but there exist extra-reduction in Barrett reduction (see line 11 in algorithm 2.9) and or extra-operation in extended Euclid algorithm (see line 17 in algorithm 2.7). The result of extra-reduction analyses can be applied on other modular reduction algorithm.

The first work on Montgomery extra-reduction analyses is published in 2001 by Walter & Thomson in [WT01]. A huge number of extra-reduction analyses is made by Werner

Schindler and his co-autors [SKQ01, Sch02, SW03, ASK05, AS08, Sch15]. We have try to classified theses previous works in three categories depending on the protection used to protect initially the algorithm.

3.5.1 ERA 1: attacks on RSA without protections

The Schindler and his coauthors [SKQ01, SW03, ASK05, AS08] exploit the extra-reductions of the Montgomery multiplication with messages chosen for the input of the modular exponentiation. These works analyze simply the timing of RSA implementations.

The main property used in these work is the following:

Proposition 3.4 (Probability of Extra-Reduction (part 1/2) [Sch00, Lemma 1]) *Assuming uniform distribution of operands, the probability of an extra-reduction in a multiply by a constant c is noted X_c is:*

$$\mathbb{P}(X_c = 1) = \frac{c}{2R} \tag{3.1}$$

Let c be a constant chosen by the attacker. It can be the ciphertext input into RSA, in decryption mode. Here, we consider that the attacker targets the timing of a multiplication with c. The duration T of this multiplication can take only two values:

- T_0, the duration of a multiplication without an extra-reduction,
- $T_0 + T_X$ of a multiplication with an extra-reduction.

Now, the other operand of the multiplication might be random, hence the duration is probabilistic. Its mean duration is:

$$\begin{aligned}\mathbb{E}(T) &= T_0 + \mathbb{P}(X = 1) \times T_X \\ &= T_0 + \frac{c}{2R} \times T_X. \quad \text{(see proposition 3.4)}\end{aligned}$$

Hence it is an affine relationship. However, when $c > p$, the ciphertext is reduced before being multiplied, hence a "rupture" in the curve, as shown in figure 3.6 and observed by Schindler in [Sch00]. Actually, the complete expression over $[0, p)$ of the timing T is:

$$T = T_0 + \frac{c \bmod p}{2R} \times T_X. \tag{3.2}$$

These attacks are interesting, but the masking of the input of the scalar multiplication is a classical countermeasure and moreover very effective against these attacks.

3.5.2 ERA 2: attacks on RSA with blinding exponent

In 2015, Schindler [Sch15] proposed a new attack exploiting again the Montgomery extra-reduction. This attack is applied on the modular exponentiation with the exponent blinding using the chosen input message.

The main drawback is the input blinding is a classical countermeasure to protect against this attack.

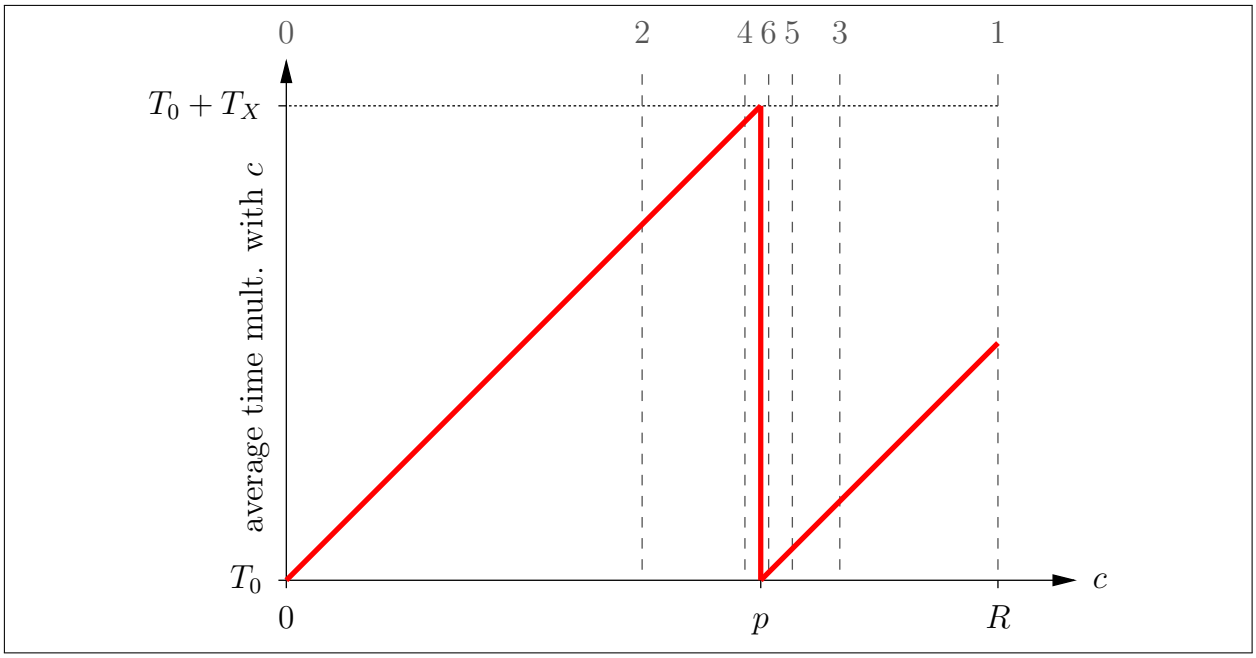


Figure 3.6: Average timing of a multiplication by a constant c (in red), and sketch of dichotomy attack by Schindler [Sch00] (in grey line with grey iteration numbers)

3.5.3 ERA 3: attacks on RSA with message blinding

Walter & Thomson [WT01] and Schindler [Sch02] show that there is a difference between the number of Montgomery extra-reductions, when the operation is a modular square or a modular multiplication.

Proposition 3.5 (Probability of Extra-Reduction (part 2/2) [Sch00, Lemma 1]) *Assuming uniform distribution of operands, the probabilities of an extra-reduction in a multiply X_{M_i} and in a square X_{S_i} at iteration i are*

$$\mathbb{P}(X_{M_i} = 1) = \mathbb{E}(X_{M_i}) = \frac{p}{4R} \qquad \textit{and} \qquad \mathbb{P}(X_{S_i} = 1) = \mathbb{E}(X_{S_i}) = \frac{p}{3R}. \tag{3.3}$$

We note that extra-reductions are 33% more likely when the operation is a square than when it is a multiply, irrespective of the ratio $\frac{p}{R} \in (\frac{1}{2}, 1)$. This allows one to break unprotected exponentiation algorithms.

We compare the theory (proposition 3.5) and experimental occurrences of extra-reductions on one million multiplications and squares, using these moduli:

1. `RSA-1024-p`: $p =$ `0xcd083568d2d46c44c40c1fa0101af2155e59c70b08423112af0c1202514bba5 \ 210765e29ff13036f56c7495894d80cf8c3baee2839bacbb0b86f6a2965f60db1`.

2. `RSA-1024-q`: $p =$ `0xca0eeea5e710e8e9811a6b846399420e3ae4a4c16647e426ddf8bbbcb11cd3f \ 35ce2e4b6bcad07ae2c0ec2ecbfcc601b207cdd77b5673e16382b1130bf465261`.

3. `RSA-1024-n`: $p =$ `RSA-1024-p` $\times$ `RSA-1024-q` (not a prime, but results apply).

4. `P-256`: $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1 =$ `0xffffffff00000001000000000000000000000000ffffffffffffffffffffffff`.

5. `brainpoolP256r1`: $p =$ `0xa9fb57dba1eea9bc3e660a909d838d726e3bf623d52620282013481d1f6e5377`.

Cryptographic parameters			$\mathbb{P}(X_{M_i} = 1)$		$\mathbb{P}(X_{S_i} = 1)$	
p	R	Ratio p/R	theory	experiment	theory	experiment
`RSA-1024-p`	2^{512}	0.800907	0.200227	0.200241	0.266969	0.266893
`RSA-1024-q`	2^{512}	0.789290	0.197323	0.198207	0.263097	0.263774
`RSA-1024-n`	2^{1024}	0.632147	0.158036	0.157875	0.210715	0.209865
`P-256`	2^{256}	1.000000	0.250000	0.250049	0.333333	0.333523
`brainpoolP256r1`	2^{256}	0.663991	0.165998	0.165846	0.221330	0.221134

Table 3.1: Extra-reduction probability for multiplications (M_i) and squares (S_i)

As shown in table 3.1, experimental probabilities are very close to the theory. Theoretical results rely on some assumptions (cleanly stated in Appendix), which justifies that this validation is useful.

Notice that the moduli do not need to be prime numbers. Consider the case of RSA implemented without CRT (see e.g., `RSA-1024-n`). It is particularly relevant to our scenario. Indeed, when the factorization $n = pq$ is unknown, the exponent randomization $d \leftarrow d + r\varphi(n)$, where r is a small (e.g., 64 bit) uniformly distributed random number [Koc96, section 10, page 9], is impossible since $\varphi(n) = (p-1)(q-1) = n + 1 - p - q$ cannot be computed without the knowledge of the two prime factors p and q. Thus, regular exponentiation is a very suitable protection, since more efficient than alternative approaches, such as signature verification or exponent splitting.

The proposition 3.5 allows to retrieve the bits of the private exponent when executing a classical algorithm such as Square-and-Multiply algorithms 2.1 or 2.2. For [WT01, Sch02], an attacker starts by taking an acquisition set with random and different messages. Note that if a masking countermeasure of the input is used to protect; this will not prevent this attack. For each i operation, the attacker identifies for each q acquisitions if there is an extra-reduction denoted $x_q^i = 1$ or not noted $x_q^i = 0$. The attacker calculates the extra-reduction average on all the acquisitions for each operation, this corresponds to the estimate of the probability $\hat{P}(X_i = 1)$. The extra-reduction probability of a square is $\frac{p}{3R}$ and the extra-reduction probability of a multiplication is $\frac{p}{4R}$ The figure 3.7 illustrates the attack of Walter & Thomson and Schindler.

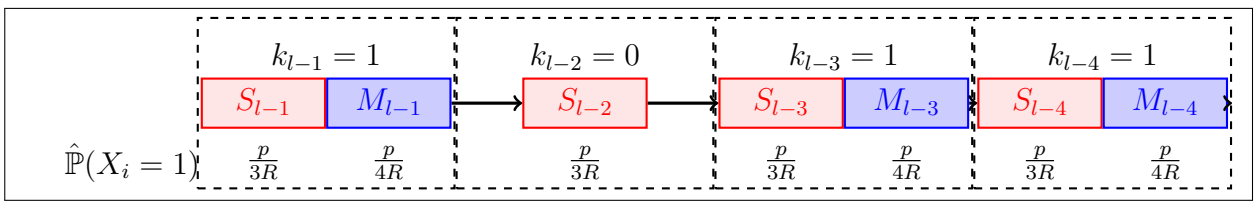


Figure 3.7: ERA3: Extra-reduction analysis by Walter & Thomson et Schindler

Using a regular algorithm protects against this attack, in fact as shown by figure 3.8, we can differentiate a bit equal to "0" of a bit equal to "1".

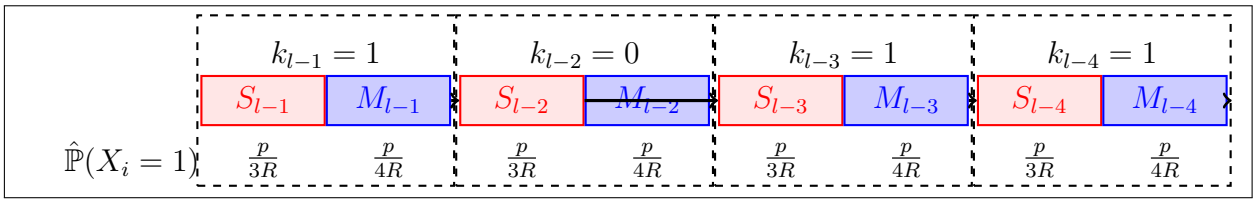


Figure 3.8: Protection against extra-reduction analysis ERA3 — used regular algorithm

3.6 Fault attacks

Fault injection attacks are a real-world threat to cryptosystems, in particular asymmetric cryptography.

3.6.1 Attacks

As put forward in the reference book on fault analysis in cryptography [JT12, Chp. 9], there are three main categories of fault attacks.
1) *Safe-error attacks* consist in testing whether an intermediate variable is dummy (usually introduced against simple power analysis [KJJ99]) or not, by faulting it and looking whether there is an effect on the final result.
2) *Cryptosystem parameter alterations* aim at weakening the algorithm in order to ease key extraction. For example [BMM00], invalid-curve fault attacks consist in moving an ECC computation to a weaker curve, enabling the attacker to use cryptanalysis attacks exploiting the faulty outputs.
3) Finally, the most serious attacks belong to the *differential fault analysis* (DFA) category. Often the attack path consists in comparing correct and faulted outputs, like in the well-known BellCoRe attack on CRT-RSA (RSA sped up using the Chinese Remainder Theorem), or the sign-change fault attack on ECC.

The *BellCoRe attack* [BDL97] on CRT-RSA introduced the concept of fault injection attacks. It is very powerful: faulting the computation even in a very random way yields almost certainly an exploitable result allowing to recover the secret primes of the RSA modulus $N = pq$.

The *sign-change attack* [BOS06] on ECC consists in changing the sign of an intermediate elliptic curve point during an elliptic curve scalar multiplication (ECSM). The resulting faulted point is still on the curve so the fault is not detected by traditional point validation countermeasures. Such a fault can be achieved by for instance changing the sign in the double operation of the ECSM algorithm (line 3 of algorithm 2.4). If the fault injection occurs during the last iteration of the loop, then the final result $\widehat{Q} = [-2\sum_{i=1}^{n-1} k_i 2^{i-1}]P + k_0 P = -Q + 2k_0 P$, i.e., either $\widehat{Q} = -Q$ or $\widehat{Q} = -Q + 2P$ depending on k_0, which reveals the value of k_0 to the attacker. This process can be iterated to find the other bits of the scalar, and optimization exist that trade-off between the number of necessary faulted results and the required exhaustive search.

Both RSA and ECC algorithms continue to be the target of **many** new fault injection attacks: see [BDF+14, LPM+14, BdSG+14, BGL14, MFGL15] just for some 2014 papers. Besides, this topic is emerging and other new fault attacks will appear sooner or later. Hence, the need for efficient and practical generic countermeasures against fault attacks is obvious. David Wagner from UC Berkeley concurs in [Wag04]: "*It is a fascinating research problem to establish a principled foundation for security against fault attacks and to find schemes that can be proven secure within that framework.*"

3.6.2 Countermeasures

Verifications compatible with mathematical structures can be applied either at computational or at algorithmic level.

Algorithmic protections have been proposed by Giraud [Gir06] (and many others [BNP07, LRT14, KKHH11]) for CRT-RSA, which naturally transpose to ECC, as shown in [KFSV11]. These protections are implementation specific (e.g., depend on the chosen exponentiation algorithm) and are thus difficult to automate, requiring specialized engineering skills.

Computational protections have been pioneered by Shamir in [Sha99] using *modular extension*, initially to protect CRT-RSA. The idea is to carry out the same computation in two different algebraic structures allowing to check the computation before disclosing its result. For example protecting a computation in $\mathbb{F}_p$ consists in carrying out the same computation in $\mathbb{Z}_{pr}$ and $\mathbb{F}_r$ ($\mathbb{Z}_{pr}$ is the direct product of $\mathbb{F}_p$ and $\mathbb{F}_r$), where r is a small number ($r \ll p$); the computation in $\mathbb{Z}_{pr}$ must match that of $\mathbb{F}_r$ when reduced modulo r, if not an error is returned, otherwise the result in $\mathbb{Z}_{pr}$ is reduced modulo p and returned. The principle of modular extension is sketched in figure 3.9.

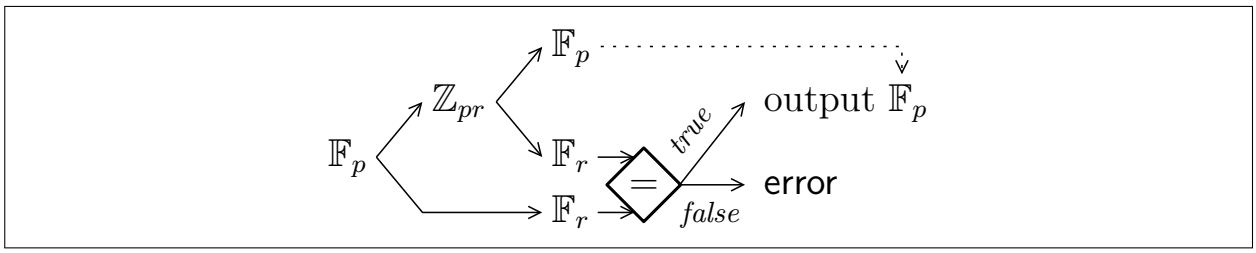


Figure 3.9: Sketch of the principle of *modular extension*.

In general, the modulus requires more words for its representations. For example, RSA-1024 with r on 32 bits, we have 32 words of 32 bits without protections and 33 words of 32 bits with this protection. We have one word overhead. However, there exists some elliptic curves like `P-521`, the r must be chosen on 23 bits without computational overhead. BSI recommends prime number size which allow for some margin, e.g. until 2022 it recommends to use 2000 bits for the public modulus in RSA [BSI17, section 3.5 p.39]. The recommendation allows to have 48 bits in 64-bits computer for some protection like the modular extension.

This method operates at low level (integer arithmetic), thereby enabling countermeasures (and optimizations) to be added on top of it. They are thus easily maintained, which explains why this method is quite popular. Indeed, there is a wealth of variants for CRT-RSA stemming from this idea [ABF+02, Vig08, JPY01, BOS03, CJ05, DGRS09], as well as a few proofs-of-concept transposing it to ECC [BOS06, BV07, Joy10]. Despite

the nonexistence of literature, the same idea could apply to post-quantum code-based cryptography, pairing, and homomorphic computation for instance. Therefore, our work focuses on computational countermeasures.

On the one hand, the variety of CRT-RSA countermeasures shows that fault attacks are a threat that is taken seriously by both the academic and the industrial communities. On the other hand, it bears witness to the handcrafted way these countermeasures were put together. Indeed, the absence of formal security claims and of proofs added to the necessity of writing implementations by hand results in many weaknesses in existing countermeasures and thus in many attempts to create better ones.

3.7 Conclusion

It is important to note that we have just detailed the attacks which justify the installation of some protections corresponding to the pyramid scheme figure 3.1. My starting point is to study implementation protected against some attacks. In the next 3 chapters, we are interested on implementations with some protections and I will show that they are also vulnerable despite their composition. In the last chapter, I would come back to the countermeasure against fault attacks named modular extension. We will go on to say that it is not well explained formally because it is not correct in all cases. We show a way to correct it and we will formally establish its level of security.

Chapter 4

Improvement of Online Template Attacks

This collaborative work with Louiza Papachristodoulou from Radboud University Nijmegen, Zakaria Najm, Lejla Batina, Jean-Luc Danger and Sylvain Guilley. This work was presented to the COSADE workshop in 2016 [DPN+16].

This improvement of Online Template Attack (IOTA) works exclusively on elliptic curve cryptography. This chapter presents a collision attack exploiting the doubling operation during an elliptic curve scalar multiplication (ECSM) implemented with some protections, as represented in the pyramid scheme (figure 4.1). Firstly, the description and requirements of the attack are explained (section 4.2). Secondly, we describe the two observed leakages (section 4.3) and their exploitation by experiments (section 4.4). Thirdly, a new method to correct and detection an error exploiting the horizontal leakage is detailed (section 4.5). Finally, the section 4.6 proposes a discussion about the efficiency of certain countermeasures against our attack (section 4.6).

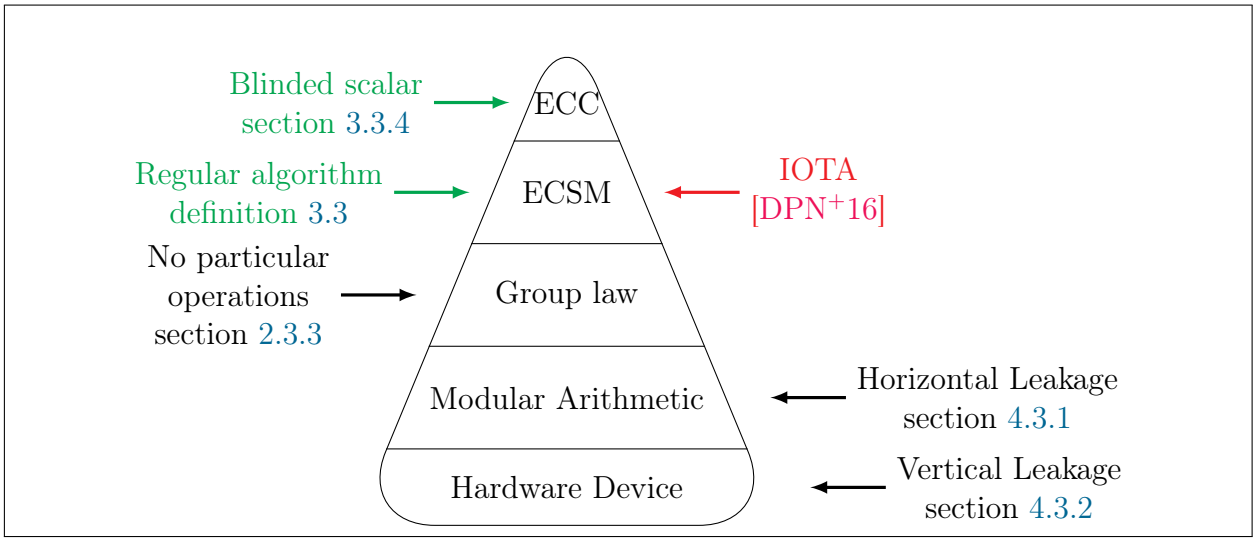


Figure 4.1: Side Channel Attack against simple countermeasures presented in this chapter

Contents

4.1 Introduction

Our improvement of Online Template Attack (IOTA) is an attack against elliptic curve scalar multiplication (ECSM) with some side channel protections. The elliptic curve scalar multiplication algorithm is regular (definition 3.3), there are no difference for the scalar bit value equals "0" or "1". The scalar can be randomized using several techniques (section 3.3.4). This attack names "Template" but the attacker cannot have access to another device to make his attack. This method can attack on ECDSA signature to find one nonce k to retrieve the secret key. The group law is not specific, but for our experiments the attack is made on a "double-and-add-always Left-to-Right" (DAA-L2R) algorithm (algorithm 3.3) using the group law in `mbedTLS` library using the `brainpoolP256r1` curve and `P-256` curve on an ARM cortex M4.

In the previous chapter, several side channel attacks were described. Our IOTA work is based on three kinds of side channel attacks: template, horizontal, and doubling attack. Template Attacks belong to yet another kind of attacks and are considered to be the most powerful method from the information-theoretic point of view, since they take advantage of most information available in a side channel observation [CRR02, MO08]. The attacker

is assumed to have one or limited numbers of side channel measurements from the target device, i.e. power, EM traces or timing, but he has access to a similar device, on which he can simulate the computations of the target (template building phase). The building phase of template is the same as described in MESD attack [MDS99]. Using a device with scalar control, the attacker takes two groups of acquisition one with the scalar bit equals "0" and one group with the scalar bit value equals "1". Horizontal attack [CFG+12, BJPW13] permits to attack ECSM with only one acquisition, but there is possible exclusively with the unified formula on the ECSM named "adding-always" (algorithm 3.6). The idea of attacking the doubling operation in the elliptic curve setting was originally proposed by Fouque and Valette in [FV03]. Their "Doubling Attack" is based on the fact that similar intermediate values may be manipulated when working with points P and $2P$. However, in most cases, the intermediate values during each iteration of ECSM are different than the input point. More details on these attacks are described in chapter 3, but the most important motivations of our attack are summarized in the table 4.1[1].

Attack works with	ECSM implementation	Point representation	Scalar blinding	Scalar control
"Doubling attack" [FV03]	✗ Only Left-to-Right	✗ Only affine	✗ Not possible	✓ Not necessary
"Template attack" [MO08, MDS99]	✓ All	✓ All	✓ Possible	✗ Mandatory
"Horizontal attack" [CFG+12, BJPW13]	✗ Only Add-Always	✓ All	✓ Possible	✓ Not necessary
OTA [BCP+14] IOTA [DPN+16]	✓ All	✓ All	✓ Possible	✓ Not necessary

Table 4.1: State-of-the-art of collision, template attacks

The most efficient result in practical template attacks on ECC is the Online Template Attack (OTA), presented in [BCP+14]. This attack requires one full target trace and one template trace per key-bit. With 256 templates, Batina et al. retrieve a 256-bit key on the twisted Edwards curve used for the `Ed25519` signature scheme, the library used is NaCl on 8-Bit AVR micro-controllers [HS13]. In our work, compared to original Online Template Attack (OTA) [BCP+14], one vertical leakage and one horizontal leakage are observed and exploited. The vertical leakage can be exploited using classical techniques (section 3.2.2) like maximum of correlation, minimum of absolute difference sum. The cause of the horizontal leakage is found and exploited in order to detect and correct a scalar bit error.

In the original OTA, the authors did not consider the success rate of the attack for an entire scalar and the fact that OTA can fail in recovering the scalar, if one key-bit guess is wrong. If a wrong key-bit assumption cannot be detected, then the error will propagate and the scalar cannot be recovered. Therefore, the advantage of our adaptive template attack over the original OTA is the fact that it detects and corrects errors. Making one assumption for each key-bit and deciding according to the established threshold if this bit

[1] ✗: Attack does not work. Countermeasures is efficient. ✓: Attack works with and without the countermeasure.

is the correct one, does not always give the correct result. In some instances of our attack, the templates obtained for a "0" key-bit assumption was very similar to the template made for the assumption that the key-bit is "1". To increase the success rate of our attack and to determine wrong guesses, we decide to obtain two template traces for each key-bit. The choice to use both assumptions to create template traces allows detecting and correcting any possible error to get back the whole scalar.

4.2 Attack description

4.2.1 Attack model for OTA

Online Template Attack (OTA), introduced in [BCP+14], is an adaptive template attack technique, which can be used to recover the secret scalar in a scalar multiplication algorithm. The main difference with the classical template attacks as described in section 3.3.2 is the absence of the building phase. The main assumption in the OTA attacker model is in his ability to choose an input point to the scalar multiplication algorithm, in order to generate template traces. As it is demonstrated in the original paper, OTA works with one *target trace* from the device under attack and one *template trace* per key-bit obtained from the attacker's device that runs the same implementation. Performing OTA in practice requires the following assumptions to be made regarding the attacker:

- The attacker knows the input P of the target device.
- He knows the implementation of the scalar multiplication algorithm and he is able to compute the intermediate values.
- He can choose the input points on a device similar to the target device.

Furthermore, the attack works with the following assumptions related to the device:

- The scalar can be randomized.
- The intermediate values are deterministic.

The OTA is then performed with three phases as follows:

1. The attacker first obtains a target trace with input point P from the target device.
2. He obtains template traces with input points $[m_i]P$, $m_i \in \mathbb{Z}$ for multiples of the point P, e.g. $2P$ or $3P$.
3. He computes the correlation between the target trace and the template trace. He assumes that the guess is correct when the correlation is higher a chosen threshold.

The main difference between the original OTA and our improvement (IOTA) is the number of Template acquisitions. Indeed, each hypothesis of one bit value is tested. The attacker generates two template traces corresponding to $2P$ and $3P$. For each assumption, the attacker computes the correlation between the target traces and template traces corresponding to this assumption. The correct guess is the one which maximizes the correlation.

The OTA and IOTA methods allow to retrieve all the scalar bits except the last bit value (the least significant bit in our case). The OTA and IOTA techniques are originally described for binary algorithms, but they can be easily adapted to the windows method by creating one template for a hypothesis made for each window. The attacker model for IOTA is more suitable for the Diffie-Hellman key-exchange protocol, because the input point can be selected. Nevertheless, this attack can be applied against the ECDSA algorithm, if the input point of the target device is known.

The algorithm 4.1 describes the IOTA technique when the DAA-L2R algorithm (algorithm 3.3) is used for the ECSM.

Algorithm 4.1: Improvement Online Template attack description

Require: P the input point, the domain, and the group law
Ensure: the scalar $k = (k_{l-1}, k_{l-2} \ldots, k_0)_2$
1: Acquisition of the target acquisition with the point P
2: Initialization of the first m_i with $i \leftarrow l$, we have $m_l \leftarrow 1$ and $k_{l-1} \leftarrow 1$
3: **for** $i = l - 1$ **down to** 1 **do**
4: Cut the doubling operation $ECDBL_{i-1}$ on the target acquisition
5: Compute the interesting points $[2m_i]P$ and $[2m_i + 1]P$ using the deterministic group law
6: Acquire the template acquisition with the input point $[2m_i]P$ and cut the first doubling named T0
7: Acquire the template acquisition with the input point $[2m_i + 1]P$ and cut the first doubling named T1
8: **if** MATCH($ECDBL_{i-1}$, T0) >MATCH($ECDBL_{i-1}$, T1) **then**
9: $k_i \leftarrow 0$ and $m_{i-1} \leftarrow 2m_i$
10: **else**
11: $k_i \leftarrow 1$ and $m_{i-1} \leftarrow 2m_i + 1$
12: **end if**
13: **end for**
14: **return** k

The line 1 in the algorithm 4.1 is the first step of the attack, it corresponds to the acquisition of the target trace with the input point P. The line 2 is the initialization of the value m_{l-1}, the most significant scalar bit value k_{l-1} by definition is equal to 1. The for-loop between lines 3-13 corresponds to the attack phase. This loop allows to retrieve each scalar bit value one by one. The explication of the line 4 is described the section 4.4.2. The description to compute the interesting point defined line 5 is made in section 4.2.2. The lines 6-7 correspond to the template acquisition phase required for the both hypotheses 0 and 1. The cut of the first doubling operation is explained in section 4.4.2. The matching phase defined line 8 by the MATCH function is explained at section 4.3 and depends on the leakage. This step decides the correct scalar bit value assumption. The lines 11-13 set the scalar bit value retrieved and allow to initialize the following multiple value m_{i-1} to continue the attack for the next bit value.

4.2.2 Constructing template traces for IOTA

Interesting points. At this point, it is important to explain precisely how the interesting points to generate the template traces are chosen (line 5 algorithm 4.1). With the term *interesting points* we mean the multiples of the point P that are expected to be the outputs of every iteration of the ECSM algorithm, i.e. $2P$ and $3P$ for the first bit of the scalar k_{l-2}. This is demonstrated with a graphical example depicted in figure 4.2. Let us

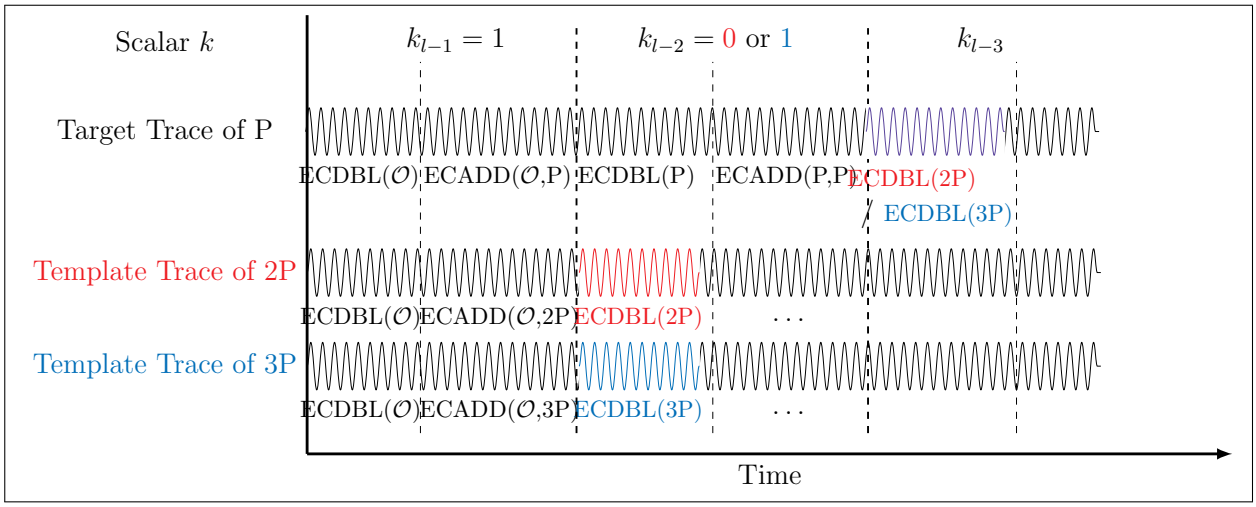


Figure 4.2: How to find the second msb k_{l-2} using the target trace with the template traces of $2P$ and $3P$

assume that the initial input point to the DAA-L2R algorithm (algorithm 3.3) is P and the most significant bit (k_{l-1}) of our secret scalar is 1. Then, the output of the second iteration (operations for k_{l-2}) is either $2P$ or $3P$. We compute the matching between the template traces $2P, 3P$, and the target trace, in order to find the most likely key-bit. The best matching value is considered to be the right key guess. For example, if $k_{l-2} = 0$, then the output of the second iteration is $2P$ and consequently the template trace for $2P$ gives higher matching value to the target trace than the template for $3P$.

We continue the same procedure of calculating the two possible outcomes for bit k_{l-3}, which are the template traces for $4P$ or $5P$, and then finding the highest correlation between the templates and the target trace. The figure 4.3 shows how the templates for the third bit k_{l-3} can be generated.

In general, for each iteration of the ECSM algorithm, we compare the second iteration of the ECSM execution (corresponding to the first doubling operation whose consumption is detected with EM) in the template trace with the $(i-1)^{\text{th}}$ doubling execution of the target trace (step 4 in algorithm 4.1). The for-loop in the DAA-L2R algorithm (algorithm 3.3) is indexed by i with $l-1 \geq i > 0$ where i is decremented after each iteration. So, the next doubling is noted $ECDBL_{i-1}$.

Choose one modular operation. For mounting our attack, we focus on the doubling operation inside the iteration loop of ECSM. This is the interesting operation that we trigger and create our templates acquisitions. In case the whole doubling operation is used to construct templates, it is not possible to achieve high similarity between our

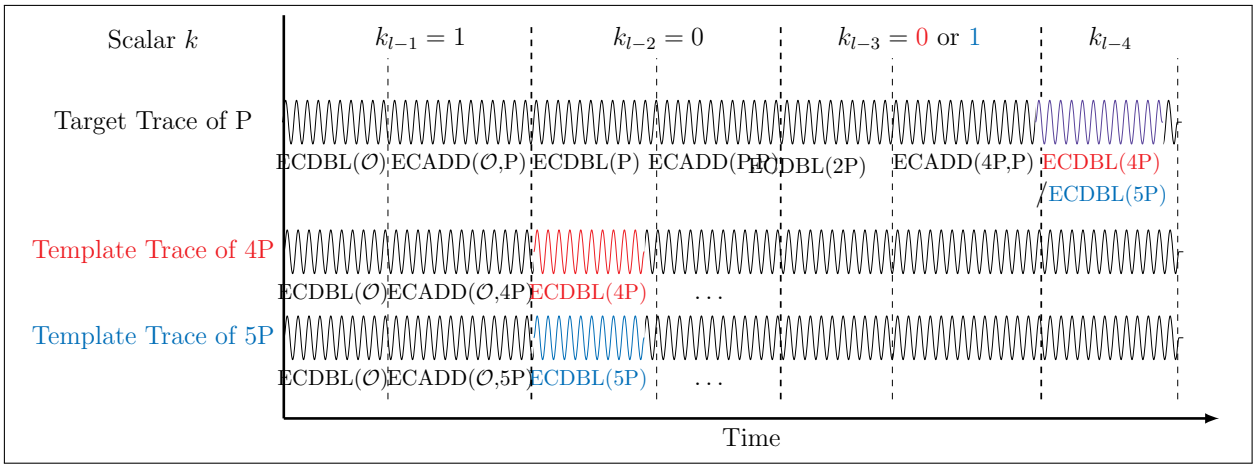


Figure 4.3: How to find the third msb k_{l-3} with $k_{l-2} = 0$ using the target trace with the template traces $4P$ and $5P$.

templates and the target, mainly due to the noise, the length of acquisition traces and the non-constant time implementation.

Furthermore, we cannot use the deterministic intermediate point as input point for the template acquisitions. In fact, in PolarSSL v1.3.7 (and generally mbedTLS) every input point is represented in affine coordinates and then converted to jacobian coordinates. This is a common approach for the input of constraint devices, since inversion during the doubling and adding operation using affine coordinates is not efficient. The *target trace* is obtained with input point $P = (x_P, y_P)$ given in affine coordinates. In order to compute the intermediate values of the points $2P = (X_{2P}, Y_{2P}, Z_{2P})$ and $3P = (X_{3P}, Y_{3P}, Z_{3P})$ with PolarSSL v1.3.7, we use the formulas defined in algorithm C.1 and algorithm C.2 in appendix C. Note that this does not correspond to the point $2P$ and $3P$ in affine coordinates, because $Z_{2P}, Z_{3P} \neq 1$ like described proposition 2.14. Therefore, we cannot compare directly the templates with input point $2P$ (resp. $3P$), since they are not in affine form.

However, by focusing on the operations in the first doubling of the DAA-L2R algorithm (algorithm 3.3) to construct the template traces, we achieve more accurate results. For the template pattern, we need only the pattern of one modular multiplication e.g. the first finite-field multiplication in the doubling[2]. In PolarSSL implementation (algorithm C.1), the first operations during the doubling of point $P = (X, Y, Z)$ are the following[3]:

$$\begin{aligned} T_1 &\leftarrow X \times X \mod p \\ T_2 &\leftarrow Y \times Y \mod p \\ T_3 &\leftarrow T_2 \times T_2 \mod p \\ &\vdots \end{aligned} \quad . \tag{4.1}$$

We create our templates with a specific input point Q_i such that the first field multipli-

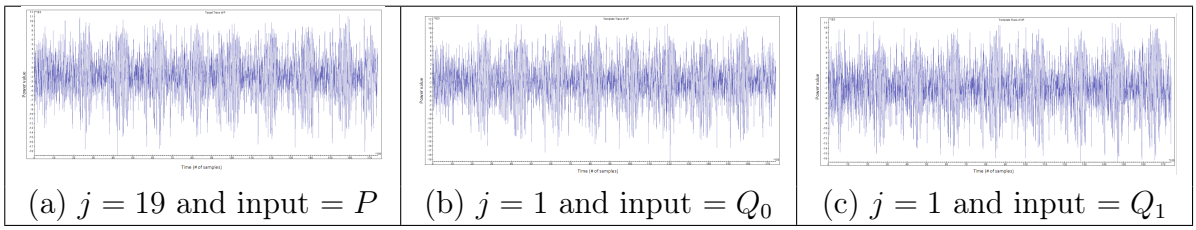
[2]In algorithm C.1, there are 7 squaring in order to perform the attack. Note that using the two input values of a multiplication can be more difficult to create the affine point with this two constraint values.

[3]Here, there is the beginning of the doubling operation implemented in PolarSSL v1.3.7. The sequence of the finite field operations in the doubling operation in the mbedTLS v2.2.0 changes to: $T_1 \leftarrow X \times X, T_2 \leftarrow 3 \times X$, but this does not affect the efficiency of our attack.

cation T_1 in $2P$ or $3P$ is the same with the one attacked on the target trace. The squaring of the X-coordinate of the intermediate value is not affected by the change of coordinates system. The way to construct the input point for templates is more sophisticated. Let us assume that we have the input point $Q_0 = (x_{Q_0}, y_{Q_0})$ in affine coordinates associated to the point value $2P$ and $Q_1 = (x_{Q_1}, y_{Q_1})$ corresponding to $3P$. We need to analyze the squaring of the X-coordinate in jacobian coordinates. The input point $Q_0 = (x_{Q_0}, y_{Q_0})$ should be a solution in $\mathbb{F}_p \times \mathbb{F}_p$ of the following system:

$$\begin{cases} x_{Q_0} = X_{2P} & \triangleright \text{ Collision} \\ y_{Q_0}^2 = x_{Q_0}^3 + ax_{Q_0} + b & \triangleright\ Q_0 \text{ is on curve} \end{cases} . \tag{4.2}$$

with a, b the parameters of the curve as defined in [BSI10, NIS13]. The number X used as input in the squaring is random, so $X_{2P}^3 + aX_{2P} + b$ is also random. If $x_{Q_0}^3 + ax_{Q_0} + b$ is not a square in the finite field, we can change one bit in X_{2P} as proposed in [BCP+14] we get another point on the curve that satisfies equation (4.2). We note here, that if we modify the 31 least significant bits (lsb) of X_{2P}, we have 99% probability $(1 - (1/2^{32}) \simeq 0.99)$ to find a point on the curve. In this case, the template matching explained in the section 4.3 does not change as described in [BCP+14]. We locate the first multiplication in the template trace corresponding to the squaring of the X-coordinate of the input point Q_0 or Q_1, depicted in figure 4.4(b) and in figure 4.4(c) respectively. With these two patterns and the target trace (figure 4.4(a)), we can perform the template matching.



(a) $j = 19$ and input $= P$ (b) $j = 1$ and input $= Q_0$ (c) $j = 1$ and input $= Q_1$

Figure 4.4: Pattern of the j-th multiplication in acquisition with different input.

4.3 Template Matching Phase

Template matching (line 8 in algorithm 4.1) is performed at suitable parts of the traces, where key-bit related assignments are used. Two independent leakages are observed in our experimental part. The first leakage described is the horizontal leakage due to the propagation of inner carry in the multiplication. The second leakage depends on the hardware device, with its power consumption or its electromagnetic emanation.

4.3.1 Horizontal leakage due to propagation of carry

The horizontal leakage in mbedTLS is a consequence of the software implementation during multiplication of large numbers e.g., a multiplication of two 256-bit field elements on a processor handling 32-bit data registers. In most cases, the multiplication of large numbers leaks due to the potential propagation of carry. This carry occurs during the register addition between two data registers (defined by the length of register). We observed that

in OpenSSL (a widely used open-source library) the different propagation of carry leaks in the same way as in mbedTLS, making this library vulnerable to some side channel attack, like timing attacks. The timing side channel leakage due to different propagation of carry can be eliminated by using a dummy operation, such as addition by zero. However, in side analysis an addition by zero can be detected using vertical leakage. Therefore, this method may create a constant time implementation, but it is still not really efficient to avoid the problem of the propagation of carry.

Horizontal leakage usually occurs when there are conditional statements in the algorithm. This is the case for PolarSSL v1.3.7, mbedTLS v2.2.0 and OpenSSL v1.0.2 and earlier versions. For mounting our attack, we focus on the doubling operation inside the ECSM. As explained in previous section 4.2.2, we cannot use the intermediate values (in jacobian coordinates) as input point (in affine) for the templates. However, we achieve more accurate results by focusing on the first finite-field multiplication in the first doubling of the DAA-L2R algorithm (algorithm 3.3) to construct the template traces.

In PolarSSL v1.3.7 and mbedTLS v2.2.0 a multiplication between two elements in the finite field is computed as described in algorithm 2.6. The result of the multiplication is stored in a 512-bit element, called *"multiplication-before-reduction"*; then the result is reduced modulo p (the characteristic of the finite field). For the curves defined in appendix B, one element in the finite field is 256-bit long. The micro-controller is Cortex-M4 with 32-bit registers for data operations (see section 4.4.1 for more details). Therefore, one field element corresponds to 8 words of 32 bits.

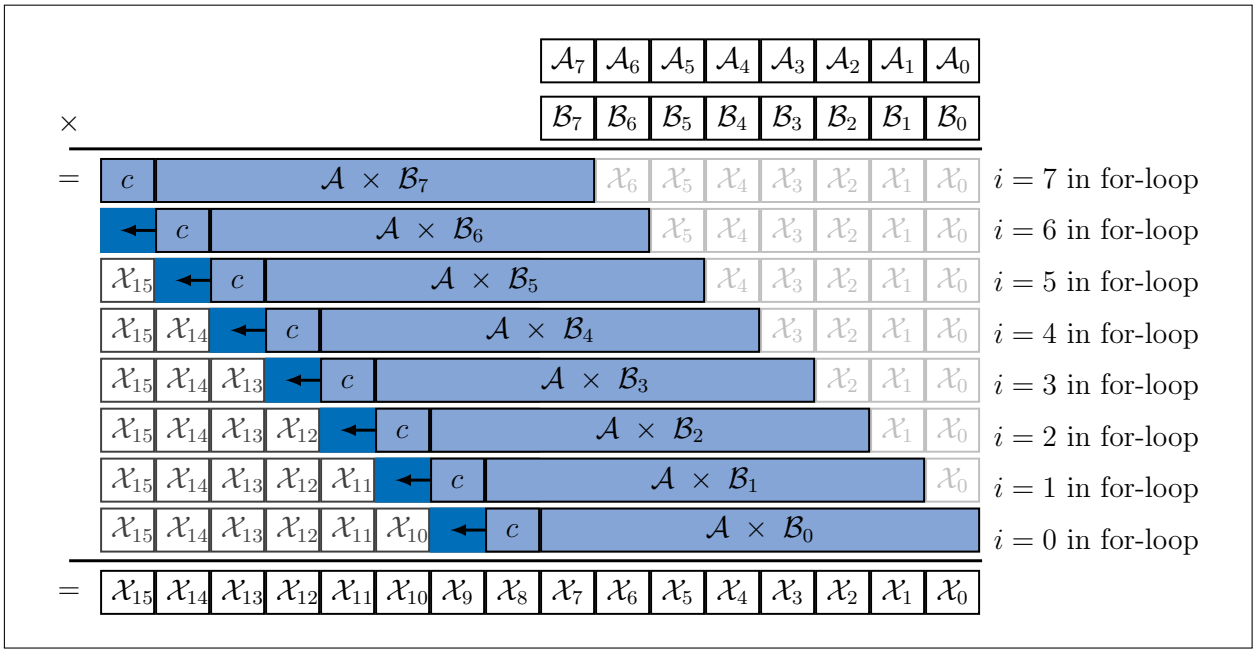


Figure 4.5: Propagation of carry during multiplication in the field

Let $\mathcal{A}$ and $\mathcal{B}$ be two 256-bit elements in the finite field. Then, $\mathcal{A}$ (resp. $\mathcal{B}$) can be written as 8 words $\mathcal{A}_i$ with $i \in \{0, 1, \ldots, 7\}$ (resp. $\mathcal{B}_i$) of 32 bits. $\mathcal{A}_0$ is the least significant word (LSW) of $\mathcal{A}$ and $\mathcal{A}_7$ is the most significant word (MSW) of $\mathcal{A}$. Let $\mathcal{X}$ be the result of the multiplication $\mathcal{A} \times \mathcal{B}$ before reduction; $\mathcal{X}$ can be represented by 16 words of 32 bits $(\mathcal{X}_{15}\mathcal{X}_{14} \ldots \mathcal{X}_0)_{32}$. The algorithm 2.6 shows how a multiplication is computed in mbedTLS.

The result $\mathcal{A} \times \mathcal{B}_i$ is stored in eight 32-bit words and there is a potential carry c, which needs to be stored separately (see step 3 in algorithm 2.6). This potential overflow creates a significant pattern that can be distinguished from its timing duration when $c \neq 0$; this pattern is the propagation of carry as depicted in figure 4.5. On the figure 4.5, the blue color represents each word of $\mathcal{X}_i$ updated at each step of the for-loop. When the final addition $\mathcal{X}_j + c$ (line 6) has got a carry, the next $\mathcal{X}_{j+1}$ is updated, this is represented by the arrow on the figure 4.5.

Theorem 4.1 (Probability to have inner carry) *The probability to have an inner carry $c = 1$ depends on the length in bits b of data register and on the MSW of finite field characteristic noted p_{MSW}:*

$$\mathbb{P}(c = 1) = \frac{1}{4}\frac{p_{MSW}}{2^b} \quad . \tag{4.3}$$

Proof Computing the probability of having an inner carry $c = 1$ is the same as computing the probability of $(X \times Y + R \times 2^b) \geq 2^{2b}$ with:

- X a random value between $[0, \max\{\mathcal{A}_{n-1} | \mathcal{A} \in \mathbb{F}_p\}]$,
- Y a random value between $[0, \max\{\mathcal{B}_i | \mathcal{B} \in \mathbb{F}_p, i \in \{0, \cdots, n-1\}\}]$, and
- R a random value between $[0, \max\{\mathcal{X}_i | \mathcal{X} \in \mathbb{Z}_{(p-1)^2}, i \in \{n-1, \cdots, 2n-1\}\}]$.

For all large numbers in the finite field, we have the values:

- $\max\{\mathcal{B}_i | \mathcal{B} \in \mathbb{F}_p, i \in \{0, \cdots, n-1\}\}$ equals $2^b - 1$,
- $\max\{\mathcal{X}_i | \mathcal{X} \in \mathbb{F}_{(p-1)^2}, i \in \{n, \cdots, 2n-2\}\}$ equals $2^b - 1$, and
- $\max\{\mathcal{A}_{n-1} | \mathcal{A} \in \mathbb{F}_p\}$ depends on the MSW of the characteristic of the finite field noted p_{MSW}.

The probability can be computed as follows:

$$\begin{aligned}
&\mathbb{P}(XY + 2^b R \geq 2^{2b}) \\
&= \sum_{x=0}^{p_{MSW}-1} \sum_{y=0}^{2^b-1} \sum_{r=0}^{2^b-1} \mathbb{P}(XY + 2^b R \geq 2^{2b} \mid X = x, Y = y, R = r)\mathbb{P}(X = x)\mathbb{P}(Y = y)\mathbb{P}(R = r) \\
&= \sum_{x=0}^{p_{MSW}-1} \sum_{y=0}^{2^b-1} \sum_{r=0}^{2^b-1} \mathbb{P}(xy + 2^b r \geq 2^{2b}) \frac{1}{p_{MSW}} \frac{1}{2^b} \frac{1}{2^b} \\
&= \frac{1}{p_{MSW}} \frac{1}{(2^b)^2} \sum_{x=0}^{p_{MSW}-1} \sum_{y=0}^{2^b-1} \sum_{r=0}^{2^b-1} \mathbb{1}_{xy+2^b r \geq 2^{2b}} \quad \text{, where } \mathbb{1} \text{ is the indicator,}
\end{aligned}$$

$$\text{i.e., } \mathbb{1}_z = \begin{cases} 0 & \text{if } z \text{ is false,} \\ 1 & \text{otherwise} \end{cases} \text{ which can be approximated by:}$$

$$\frac{1}{p_{MSW}} \frac{1}{(2^b)^2} \int_{x=0}^{p_{MSW}-1} \int_{y=0}^{2^b-1} \int_{r=0}^{2^b-1} \mathbb{1}_{xy+2^b r \geq 2^{2b}} \, \mathrm{d}r \, \mathrm{d}y \, \mathrm{d}x$$

$$\simeq \frac{1}{p_{MSW}} \frac{1}{(2^b)^2} \int_{x=0}^{p_{MSW}} \int_{y=0}^{2^b} \int_{r=0}^{2^b} \mathbb{1}_{xy+2^b r \geq 2^{2b}} \, \mathrm{d}r \, \mathrm{d}y \, \mathrm{d}x$$
$$= \frac{2^b}{p_{MSW}} \int_{x=0}^{p_{msw}} \int_{y=0}^{1} \int_{r=0}^{1} \mathbb{1}_{xy+r \geq 1} \, \mathrm{d}r \, \mathrm{d}y \, \mathrm{d}x \ ,$$

with $x \leftarrow x/2^b$, $y \leftarrow y/2^b$, $r \leftarrow r/2^b$ and $p_{msw} = p_{MSW}/2^b$. It holds, $\mathbb{1}_{xy+r\geq 1} = \mathbb{1}_{r \geq 1-xy}$. Besides, $1 - xy \in [1 - p_{msw}, 1] \subset [0, 1]$.

Indeed,

$$0 \leq x \leq p_{msw}, 0 \leq y \leq 1 \implies 0 \leq xy \leq p_{msw}, \text{ hence } 1 - p_{msw} \leq 1 - xy \leq 1 \ .$$

Therefore,

$$\begin{aligned}
\mathbb{P}(XY + 2^b R \geq 2^{2b}) &\simeq \frac{2^b}{p_{MSW}} \int_{x=0}^{p_{msw}} \int_{y=0}^{1} \int_{r=0}^{1} \mathbb{1}_{xy+r\geq 1} \, \mathrm{d}r \, \mathrm{d}y \, \mathrm{d}x \\
&= \frac{2^b}{p_{MSW}} \int_{x=0}^{p_{msw}} \int_{y=0}^{1} \int_{r=1-xy}^{1} \mathrm{d}r \, \mathrm{d}y \, \mathrm{d}x \\
&= \frac{2^b}{p_{MSW}} \int_{x=0}^{p_{msw}} \int_{y=0}^{1} xy \, \mathrm{d}y \, \mathrm{d}x = \frac{2^b}{p_{MSW}} \int_{x=0}^{p_{msw}} x \, \mathrm{d}x \times \int_{y=0}^{1} y \, \mathrm{d}y \\
&= \frac{2^b}{p_{MSW}} \left[\frac{x^2}{2}\right]_0^{p_{msw}} \times \left[\frac{y^2}{2}\right]_0^{1} = \frac{2^b}{p_{MSW}} \frac{p_{msw}^2}{2} \times \frac{1}{2} = \frac{2^b}{p_{MSW}} \frac{1}{4} p_{msw}^2 \\
&= \frac{1}{4} \frac{p_{MSW}}{2^b} \ .
\end{aligned}$$

□

Example For our experiments, we focus on the curves `brainpoolP256r1` and `P-256` defined in appendix B. For `brainpoolP256r1`, the most significant word of 32-bit length is p_{MSW} = `0xA9FB57DA` and the probability of having a propagation of carry is close to $\mathbb{P}(c = 1) = 0.166$. For `P-256`, the most significant word of 32-bit length is p_{MSW} equals $2^{32} - 1$, so this probability is close to $\mathbb{P}(c = 1) = 0.25$.

Proposition 4.2 (Probability to have an horizontal leakage) *As shown in figure 4.5, we can have 7 propagations during the multiplication, but we cannot detect the last propagation. So, the probability to have two templates with the same propagation of carry, denoted by $\mathbb{P}(T)$, is:*

$$\mathbb{P}(T) = \sum_{i=0}^{6} \binom{6}{i} p^{2i} (1-p)^{2(6-i)} \ , \tag{4.4}$$

where p is the probability to have an internal propagation of carry.

Proof Straight-forward.

Example For `P-256`, the probability to have horizontal leakage is 0.95 using $p = 0.25$. The probability to have horizontal leakage is 0.86 using $p = 0.17$ for `brainpoolP256r1`.

However, it is more interesting from the IOTA point of view to find out when a difference in the propagation of carry occurs between the target and template traces. This is the only part of `mbedTLS` that is non-constant time and we take advantage of this timing difference, every time it occurs. In this case, there is an obvious horizontal leakage between the target and the template traces, as depicted in figure 4.10.

4.3.2 Vertical leakage due to signal amplitude

In constant time executions of our implementation, there is no difference in the propagation of carry and the template traces are synchronized with the target trace (figure 4.11). In those cases, we observe only a vertical leakage due to the amplitude of the signal and the same method as described in [BCP+14] can be used. Our pattern matching technique, in order to distinguish the right hypothesis on the attacked bit of the scalar, is based on the Pearson correlation coefficient $\rho(X, Y)$ between the target trace and the template traces.

$$\rho(X,Y) = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2}\sqrt{\sum_i (Y_i - \bar{Y})^2}} = \frac{\langle X - \bar{X},\ Y - \bar{Y}\rangle}{||X - \bar{X}||\ ||Y - \bar{Y}||} \ . \tag{4.5}$$

We chose this metric, since it is both scale and offset-shift invariant. In this work, we use the maximum of the Pearson correlation to distinguish the right hypothesis, but we can use the others techniques described in section 3.2.2.

4.4 Experimental part

4.4.1 Acquisition Setup

The target device is an STM32F4 micro-controller, which contains an ARM Cortex-M4 processor running at its maximum frequency (168MHz). We imported the assembly code originally included in PolarSSL v1.3.7 to ARM Cortex-M4 and implemented the double-and-add-always procedure as described in [Cor99, Joy03]. For the acquisition, we used a 54855 Infiniium Agilent oscilloscope and a Langer EMV-TECHNIK RF-U5-2 near field probe. The sampling frequency is 1GSa/s with 50MHz hardware input low-pass filter enabled. Matlab 2014b is used for the analysis, and Inspector SCA tool [Ins] for depicting the traces in this paper. The position of the probe was determined to maximize the signal related to the activity of the 32×32 hardware multiplier[4].

For the curves defined in appendix B, one element in the finite field is 256-bit long. Thus, each operation over the field consists of manipulating eight processor words (8×32 bits). In our implementation, a multiplication-before-reduction consists of eight multiplications between a 256-bit element by each 32-bit words of the second element. It leads to eight easily identifiable patterns of eight blocks on EM traces. The length between two blocks can be different depending on the propagation of carry, as explained in section 4.3.1.

4.4.2 Pre-processing Phase

The pre-processing phase starts with choosing an input point P and obtaining the target trace from our target device; this is depicted in figure 4.6.

In this trace, we need to spot the multiplication patterns, which are eight blocks of 256×32 multiplications depicted in figure 4.7. We note here that this does not constitute a building phase in the usual template setting; it is just an identification phase. From the implementation and the device, we know that a 256-bit element is processed into 32-bit

[4]This is a simple identification phase, where we scan the device and find where the crypto processor is. Then we just move the probe around this position, in order to get a signal as clear as possible.

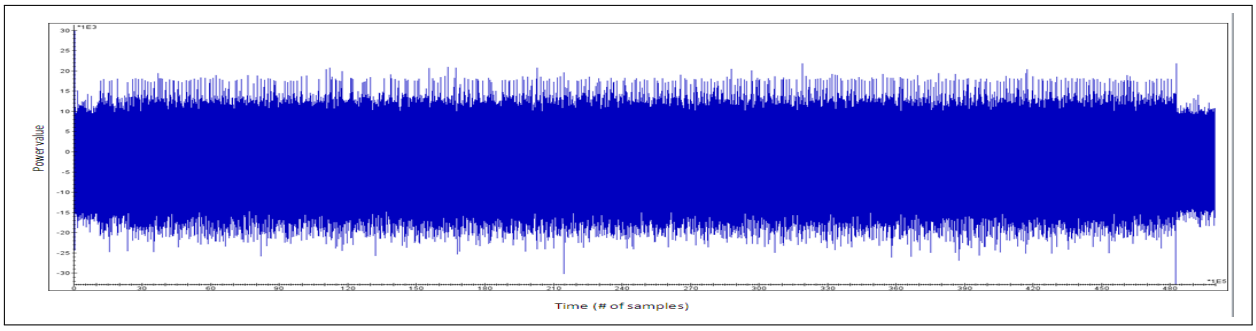


Figure 4.6: Electromagnetic emanation acquisition for ECSM on `P-256`with $k =$ `0xA5A5`

multipliers. Therefore, we expect to see eight patterns for each multiplication 256 by 32. The multiplication procedure is described in section 4.3.1.

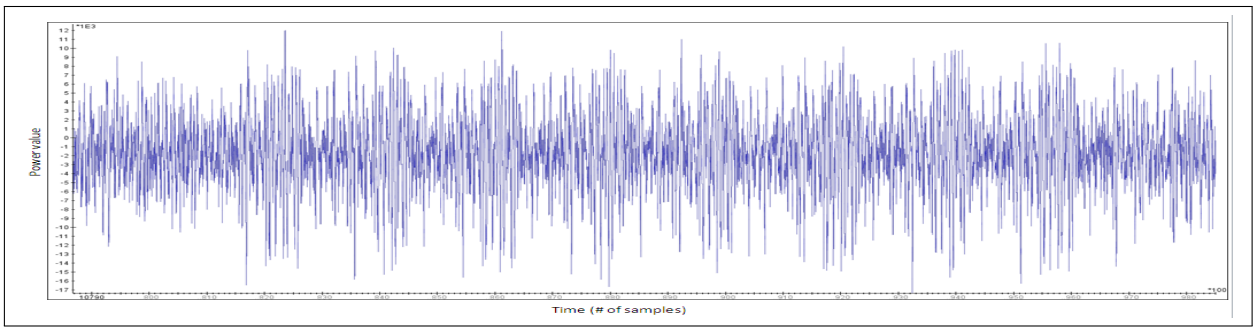


Figure 4.7: Pattern of multiplication-before-reduction

When we have a clear pattern for the multiplication, we cross correlate this pattern with our target trace and we obtain the cross-correlation pattern with one peak at the position of every multiplication. The figure 4.8 shows the cross correlation of the target trace with the multiplication pattern.

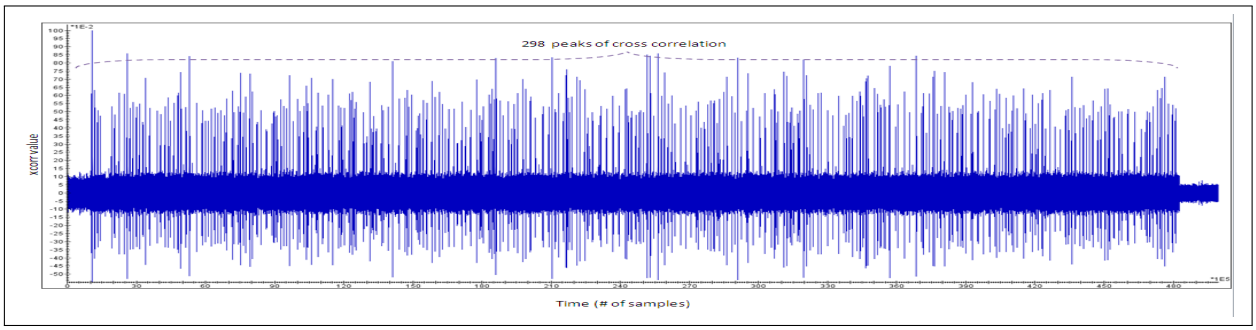


Figure 4.8: Cross correlation between the pattern of the multiplication and the target trace

By counting the peaks in the cross-correlation trace, we can find the part of the computation that we are interested in. For `brainpoolP256r1`, as explained in [BL], the doubling consists of 10 multiplications (except for the first doubling, where there are only 7 multiplications[5]), and the mixed addition consists of 11 multiplications. For `P-256`,

[5]Because in the beginning $Z = 1$ and we computed aZ^4 with 3 multiplications.

there is a particular parameter equal to $(-3 \mod p)$, so the multiplication by a in the doubling can be optimized. The doubling consists of 9 multiplications and the mixed addition of 11 multiplications.

In this way, we can "cut" the target trace in sections according to the loop of the ECSM operation (as in figure 4.9). The first iteration of the double-and-add always algorithm

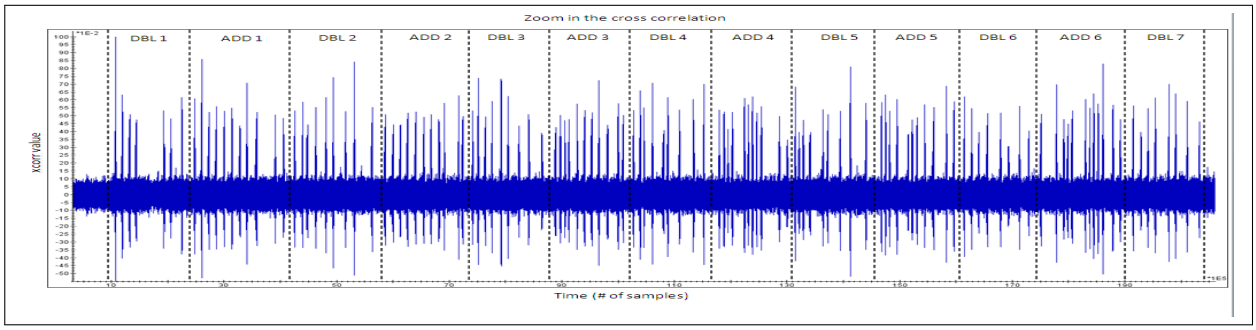


Figure 4.9: The first seven iterations of the ECSM algorithm on the curve

is completed after 18 peaks of cross-correlation. For the next iterations, we take into account that each doubling consists of 9 or 10 multiplications and each addition of 11. For the first scalar bit, the interesting section on the target is the 19th multiplication. For the second scalar bit, the interesting section is the 39th multiplication for `P-256` or the 40th multiplication for `brainpoolP256r1`. For the third scalar bit, the interesting section on the target is the 59th multiplication for `P-256` or the 61th multiplication for `brainpoolP256r1`, and so on for all the other bits of the scalar.

At the last step of this phase, we calculate multiples of the point P— each scalar bit hypotheses Q_0 and Q_1 — like described in section 4.2.2 using our PolarSSL v1.3.7 implementation.

4.4.3 Template Matching

In this section, we present how to perform template matching by making the right hypothesis on a scalar-bit. This procedure is described for the cases of horizontal and vertical leakage. The probability of having a horizontal leakage corresponds to the probability of having different propagation of carry between the two templates.

Horizontal leakage. Horizontal leakage is observed when there is different propagation of carry between two multiplications 256×32 in the field. When the propagation of carry is not the same between the two template traces in 95% of cases in `P-256` and 86% in `brainpoolP256r1` computed by the proposition 4.2, the acquisition are not synchronized. In this case, there exist a horizontal leakage. The cross correlation between the multiplication pattern and the target trace is performed before template matching, in order to choose the correct part of the target trace. Then we align the template and target traces and decide what the correct key-bit guess is.

In figure 4.10 we see the misalignment of the traces due to propagation of carry.

At the beginning of the multiplication (left part of figure 4.10) the two acquisitions are aligned. In fact, we aligned the first multiplication 256×32, as a consequence in this example the second multiplication 256×32 is also aligned, because there are no carry

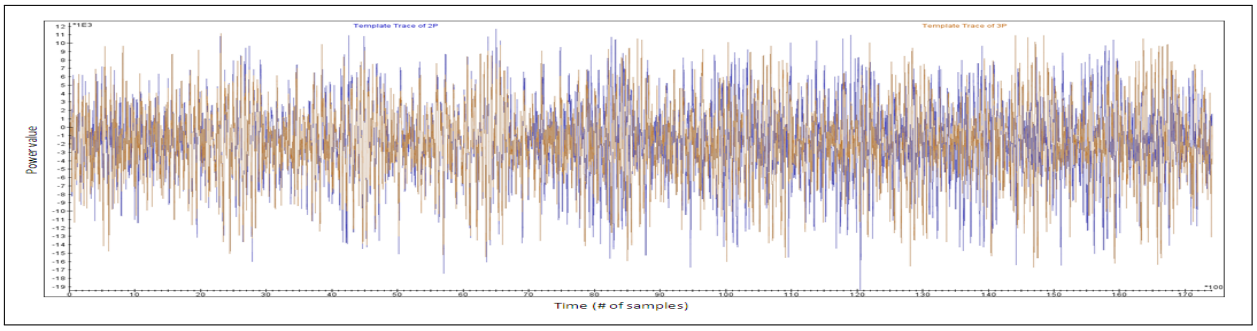


Figure 4.10: Misalignment of two template traces due to propagation of carry

between these two multiplications 256×32. The alignment of the beginning allows us to observe that the more one advances in time, that the algorithm of multiplication of large numbers executes, a desynchronization between the two traces appears. Indeed the more the algorithm unfolds, we have more chance to having different carry between the two acquisitions and therefore a misalignment is observed. On the right of the figure 4.10 the multiplication 256×32 are no aligned.

Vertical leakage. The absence of horizontal leakage is due to the same propagation of inner carry in the computation, it represents 14% of cases in `brainpoolP256r1` and 5% in `P-256` (pourcentage computed by proposition 4.2). When the implementation is executed in constant time and the template traces are synchronized with the target trace, the same method as described in [BCP+14] can be used. The propagation of carry is the same between the two templates and the target as depicted in figure 4.11; therefore, we can only observe a *vertical* leakage in our traces. In our experiments, we use the Pearson

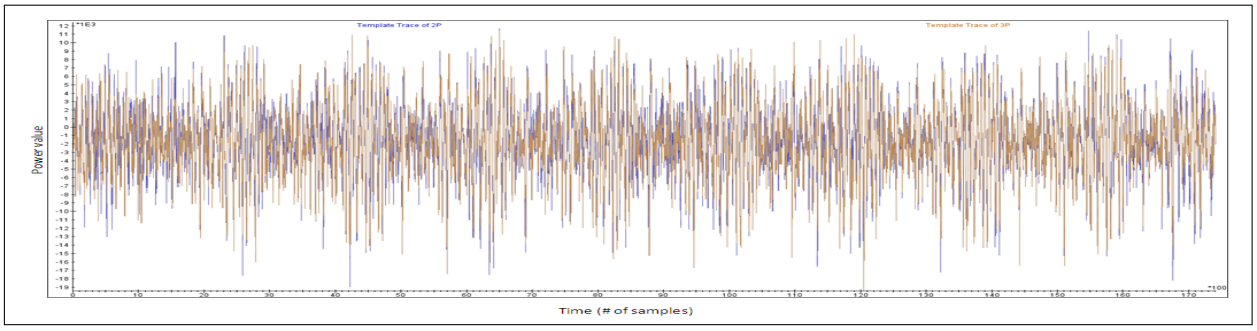


Figure 4.11: Two templates with the same propagation of carry

correlation coefficient $\rho(X, Y)$ as described in section 4.3 and we get a correlation of $0,81$ for the multiplication obtained from the target trace and the template trace of Q_{k_i} (the "correct" template value). The same value drops to $0,78$ for the correlation of the target trace with the template trace of Q_{1-k_i} (the "wrong" template value). The difference between the two correlation is not huge, so we will evaluate in the next section the success rate of the attack.

4.5 Success rate analysis and its improvement

In this part, we explain two methods to increase the global success rate of the attack. The first method is to increase the success for each key-bit value in the vertical leakage scenario. This method uses more template acquisitions for one input point. The second technique is to detect and correct an error, using template acquisition for more input points.

4.5.1 Success Rate for one key-bit

As explained in section 4.3.1, the probability to have a different propagation of carry between two templates is 95% for `P-256` and 86% for `brainpoolP256r1`. The horizontal attack scenario is easy, since if two templates have different propagation of carry, then the success rate of finding this bit is 100%. For the vertical attack scenario, the success rate depends on the input data. Therefore, we examine only the input data, for which the propagation of carry is the same in two template traces. The success rate per key-bit in vertical leakage is 76.23% for `P-256` and 69% for `brainpoolP256r1`. The total success rate to find one key-bit, independent of the leakage model, is $1 \times 0.95 + 0.76 \times 0.05 = 98.8\%$ for `P-256`, and $1 \times 0.86 + 0.69 \times 0.14 = 95.66\%$ for `brainpoolP256r1`. Averaging template traces can increase the vertical information leakage and reduce the noise of measurements. When the scalar is randomized, we cannot perform the attack with more than one target traces. However, we can still acquire more than one template traces. By using only one target trace with an average of a few template traces, the success rate increases as shown the table 4.2 on the `brainpoolP256r1` curve. For instance, by using 100 template traces

Number of average traces	1	10	50	100
Success Rate	69%	80.70%	91.60%	99.80%

Table 4.2: Different success rates on 3000 attacks according to the number of average template traces on `brainpoolP256r1` curve.

the success rate for one bit of scalar for `brainpoolP256r1` curve is $1 \times 0.86 + 0.99 \times 0.14 = 99.86\%$

4.5.2 Error-correcting bit from the template traces

The novelty of IOTA is the possibility of detection and correction of errors. In fact, in original OTA [BCP+14], the authors choose a threshold of correlation separating the right hypotheses and the bad hypotheses. Founding this threshold, when the acquisition is noisy, it is very difficult. But using an extra-cost of template acquisitions, the success rate for one bit value increases. As we described in the section 4.5.1, the success rate to retrieve one bit using OTA is close to 99%, which means that there is a 1% probability of having an unsuccessful attack due to a wrong key-guess. For a 256-bit scalar, if an error occurs in the beginning and it is not detected, the success rate for whole scalar bits in OTA scenario is 7.6% ($0.99^{255} \simeq 0.076$), since this error will propagate and affect all the bits after the wrong guess. Therefore, it is very important to detect and correct

errors before making new templates guess acquisitions. An error can be made when both template traces have the same propagation of carry. In order to be sure for a key-bit value, the value of this current bit and the following one can be computed.

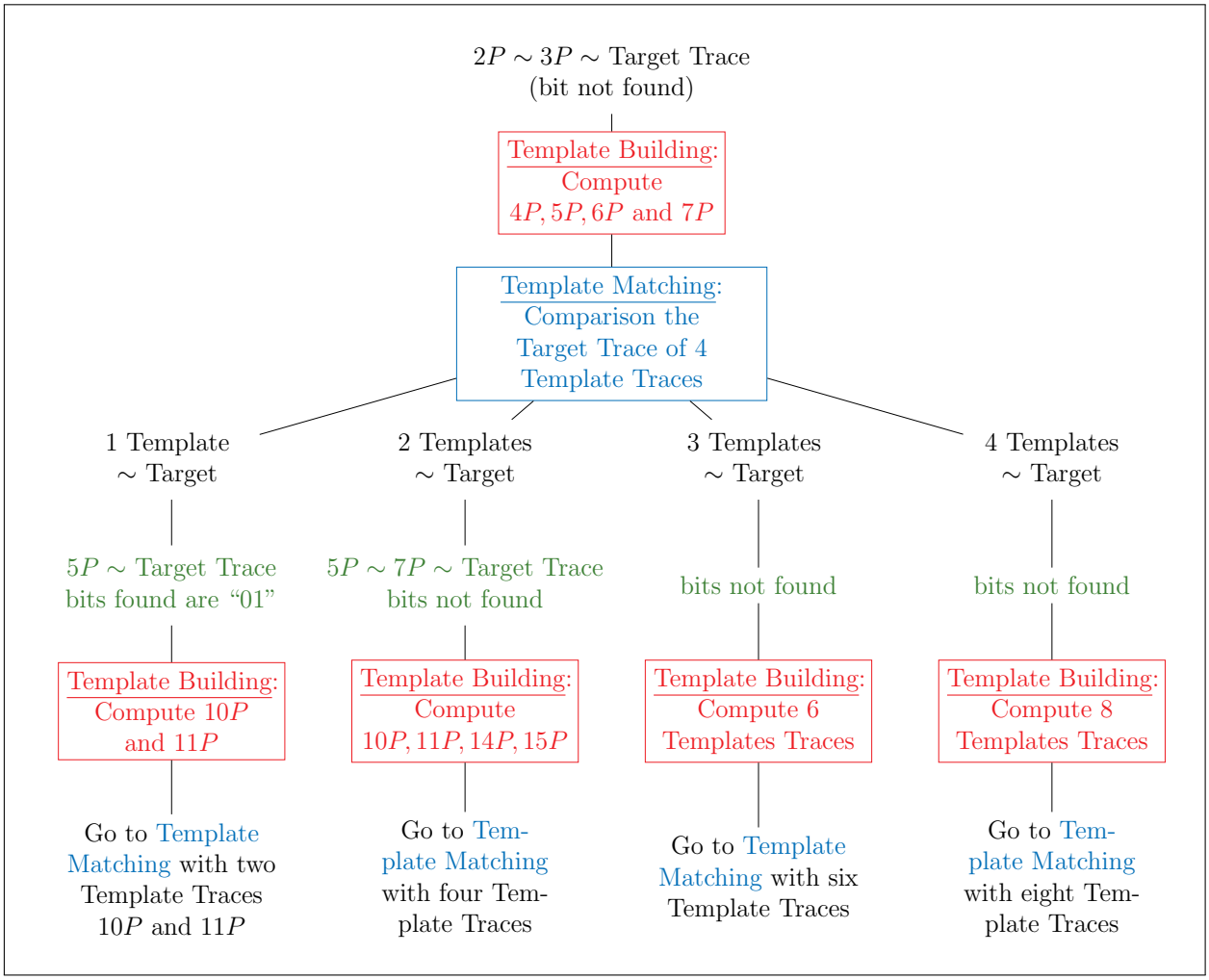


Figure 4.12: Four cases if the template trace $2P$ and $3P$ have the same propagation of carry

For instance, if the two templates for $2P$ and $3P$ have the same propagation of carry, then we create templates for $4P$, $5P$, $6P$ and $7P$. The following four cases can occur as shown on figure 4.12:

1 Template $\sim$ Target. One template has the same propagation with the target: Then, we can determine correctly both key-bit. For instance, if the template trace of $5P$ gives the same propagation of carry as the target trace, then the only possible bit values are "0" for the second key-bit and "1" for the third one.

1 Templates $\sim$ Target. Two templates have the same propagation with the target: We need to compute the next 4 templates $(4m_i, 4m_i+1, 4m_i+2, 4m_i+3$, where m_i is the i-th first bits of the exponent), and recover the next bit using these 4 template traces. By example, if $5P$ and

$7P$ have the same propagation of carry with the target trace, also we compute the four template trace with the input point $10P$,$11P$,$14P$ and $15P$.

2 Templates $\sim$ Target. Three templates have the same propagation with the target: We compute 6 templates, and recover the next bit using those 6 template traces.

3 Templates $\sim$ Target. Four templates have the same propagation with the target: We compute 8 templates, and recover the next bit using those 8 template traces.

The figure 4.13 is the generalization of IOTA method for each cases. The probability

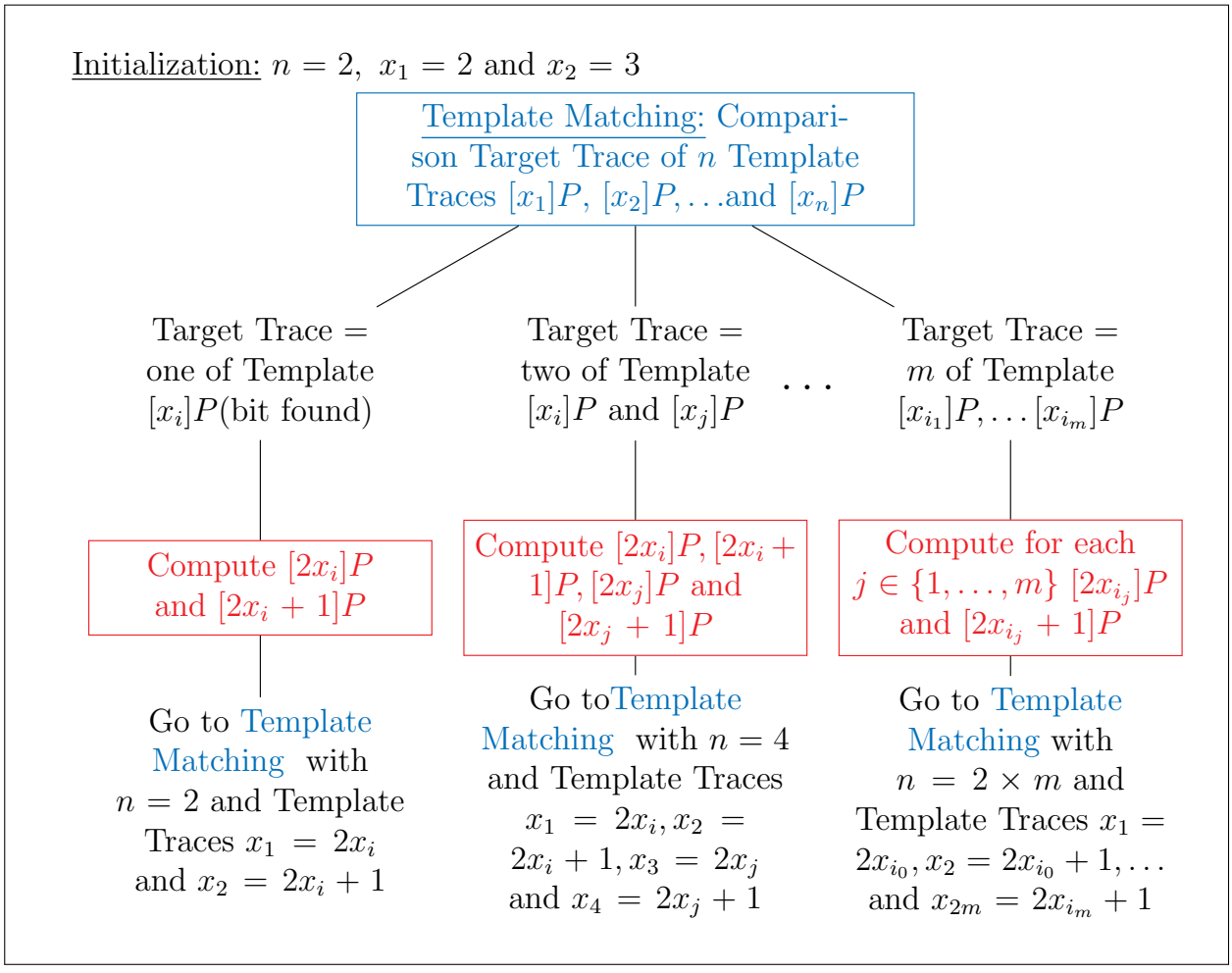


Figure 4.13: Generalization of the detection and correction method

to have one template trace with the same propagation of carry with the target trace is described in table 4.3. The probability computed on the table 4.3 depends on the number of template traces to compared (depth of the tree). For both curves, these probabilities reduce significantly for more template traces. These probabilities were computed with the same method described in proposition 4.2.

Depth d	Curve	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	...
2	`brainpoolP256r1`	86%	14%	-	-	-	-
2	`P-256`	94%	6%	-	-	-	-
4	`brainpoolP256r1`	86%	9.9%	2.5%	1.15%	-	-
4	`P-256`	94%	5.25%	0.59%	0.1%	-	-
8	`brainpoolP256r1`	86%	9.99%	2.5%	0.78%	0.25%	...
8	`P-256`	94%	5.2%	0.59%	0.09%	0.01%	...

Table 4.3: Probabilities to have m template traces with the same propagation of inner carry among d template traces.

As a conclusion, the number of template traces cannot increase exponentially. At the end of the attack, in order to retrieve the 256-bit scalar, there can be an uncertainty for the last 2 or 3 bits. By exhaustive key search or by comparing the corresponding templates with the template of $Q = [k]P$, the last 2^2 or 2^3 scalar key-bit can be found.

Remark When there is no timing difference due to the inner carry propagation inside the multiplication, the same method can be applied. In order to have a limited number of hypotheses, only the branches with the higher probabilities of success are keeping.

4.6 Countermeasures

At this point, it is clear that both OTA and our adaptive template attack are very efficient methods to attack ECSM during the execution of ECC protocols. These methods can be easily adapted to other ECSM algorithms as described in [Joy03, Riv11]. For the binary algorithm Montgomery Ladder [JY02], we can create templates for the doubling operation and find the correct key-bit. For the non-binary algorithm using windows, we can obtain templates for all hypotheses and make the same attack with more template traces. Since the most commonly used ECSM algorithms are vulnerable to our attack, it is interesting to see which countermeasures can be applied against it. We hereby study the classical countermeasures against side channel attacks and their efficiency against our attack.

4.6.1 Randomization of the scalar

The result of randomizing the scalar will be getting traces with $[k']P$ instead of $[k]P$, with k' defined below. The important property that thwarts randomization of the scalar, is the fact that we need only one target trace, the same randomized k' is manipulated throughout the attack. For the template traces, we always need the first part of the trace, which corresponds to the beginning of the ECSM algorithm running with input point a multiple m_i of P. This part of the trace is not affected by the randomization of the scalar.

For all randomization techniques detailed section 3.3.4, our attack can be applied; the *target trace* requires one acquisition on the second or third ECSM. This acquisition is possible using an oscilloscope with large memory depth. For the template traces, we make assumptions for each part of the ECSM. We retrieve a random scalar part for each ECSM part. In order to retrieve the scalar, we compute the addition (randomization 1.), the multiplication (randomization 2.) or both addition and multiplication of the

scalar (randomization 3.). For the three cases, we can recover the randomized scalar k'. Therefore, scalar randomization is not efficient against our type of attack.

4.6.2 Randomization of the point

The jacobian representation of the point can be easily randomized similar to projective coordinates. For jacobian coordinates, the randomization consists in selecting a random r in the finite field $\mathbb{F}_p$, and computing: $(X, Y, Z) \mapsto (r^2 \times X, r^3 \times Y, r \times Z)$. In most cases, the input point is in affine coordinates, so the randomization of the point is reduced to compute: $(x, y) \mapsto (r^2 \times x, r^3 \times y, r)$. The supplementary cost of this countermeasure is 5 finite field multiplications. For comparison, the cost of one ECSM using 256-bit scalar with a regular algorithm such as double-and-add-always is 5100 multiplications. Applying randomization of the input point does not allow predicting intermediate values of the calculation and prevents the construction in a deterministic way for the template traces. This countermeasure is implemented in mbedTLS, and it should be used when the device under attack has a random generator.

4.6.3 Random isomorphic elliptic curve

The idea to protect ECSM by transforming a curve through various random morphisms was initially proposed by Joye and Tymen in [JT01]. Assume that ϕ is a random isomorphism from $\mathcal{E}_K \to \mathcal{E'}_K$, which maps $P \in \mathcal{E}_K \to P' \in \mathcal{E'}_K$. Multiplying P' with k will give $Q' = [k]P' \in \mathcal{E'}_K$. With the inverse map ϕ^{-1} we can get back to $Q = [k]P$. For applying our attack, we need to know the internal representation of the point, so if P' is on a different curve that the adversary does not know, he cannot create input points in this representation.

4.6.4 Random field extension

This countermeasure prevents our attack, in the same way as described in [BCP+14]. Random field extension can be performed by changing the prime of the finite field that our computations take place. If we use random primes to generate our curves, our attack will not work.

4.7 Conclusions

In this chapter we presented a practical extension of OTA on `brainpoolP256r1` and `P-256` curves implemented on an ARM Cortex M4 micro-controller. A modified version of OTA is applied with the Pearson correlation coefficient as distinguisher for the correct hypothesis on the key-bit. Error detection and correction of a wrong key-bit guess is possible for our adaptive template attack, and increases the success rate of the attack from 7.6% to 99.8% for a 256-bit scalar. We achieve these results by averaging 100 template traces and using two template traces to recover each key-bit. Horizontal leakage due to conditional statements was not expected to be seen in recent cryptographic implementations, but unfortunately they are still used. An implementation without conditional statements would not prevent our attack, but it would reduce its success rate to that of

IOTA. Most of the countermeasures applicable to the original OTA attack should also work against our attack. Randomizing the input point, by randomizing its coordinates, for every execution of the attacked algorithm is the most efficient countermeasure against OTA/IOTA, though incurring with some cost for the performance of the implementation. Point randomization is also efficient against our attack, since we need to know the input point and its intermediate values. Actually, the adversary needs to be able to choose input points for templates. The countermeasure of point blinding must be activated, in order to use mbedTLS in ARM embedded devices. Adoption of this countermeasure is not straight-forward, because it requires the use of a random generator in the device under attack.

This chapter presents an "horizontal attack" using only one acquisition with the sensitive data. But the major requirement is the knowledge of the input data. In the two following chapters (chapter 5, chapter 6), the input data are unknown and blinded. Using an horizontal analysis followed by a vertical analysis, the sensitive data (scalar or exponent in RSA) can be retrieved in regular algorithms. The horizontal analysis has similarity with this chapter, but the vertical analysis is the novelty.

Chapter 5

Correlated Extra-reduction defeat blinding regular algorithm

This work was presented in the international workshop CHES 2016 [DGD+16] with Sylvain Guilley, Jean-Luc Danger, Zakaria Najm, and Olivier Rioul.

This chapter presents a new vertical side channel attack exploiting the information coming from the extra-reduction in Montgomery multiplication in order to break blinding regular algorithm. This side channel attack is mainly based on an attack against RSA cryptosystems, but this attack can be applied in particular cases to elliptic curve cryptography. It allows to mount successful attacks even against blinded asymmetrical computations with a regular exponentiation algorithm, such as "Square-and-Multiply Always" (algorithm 3.1) or "Montgomery Ladder" (algorithm 3.2). We investigate various attack strategies depending on the context—known or unknown modulus, known or unknown extra-reduction detection probability, etc.— and implement them on two devices: a single core ARM Cortex-M4 and a dual core ARM Cortex M0-M4. The figure 5.1 represents the countermeasures used to protect the private modular exponentiation. There are two steps: one horizontal step to find with only one acquisition the presence or absence of extra-reduction, and one vertical step to use the information of theses extra-reductions in order to recover the private exponent.

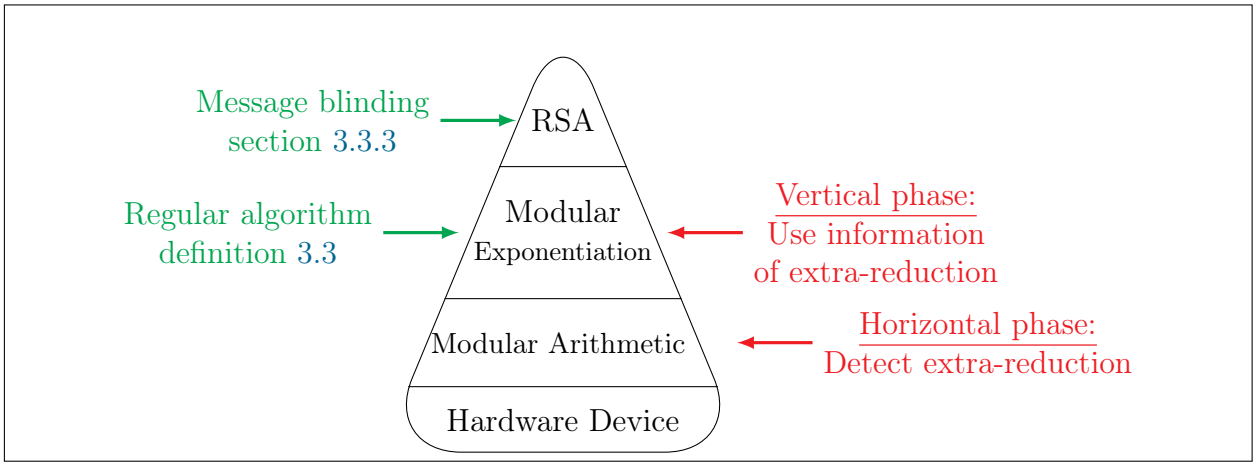


Figure 5.1: Biais exploited in these kinds of implementations with some countermeasures.

Contents

5.1 Introduction

The extra-reduction can be exploited using the execution time difference. The first timing attack against RSA was proposed by Kocher in [Koc98]. This attack is applied with the straight-forward method RSA. The CRT method (theorem A.3) protects against this attack. As explained in the chapter 3, the extra-reduction information is exploited mainly by Werner Schindler and his co-authors [Sch00, SKQ01, Sch02, SW03, ASK05, AS08, Sch15], but also by Walter& Thomson [WT01]. These attacks were classified in three families:

– `ERA 1`: The global timing attack is proposed to exploit extra-reduction to retrieve the private exponent without blinding protection [SKQ01, SW03, ASK05, AS08]. (see section 3.5.1)

– `ERA 2`: In [Sch15], Schindler presents an attack against modular exponentiation with blinding exponent using the global timing of the exponentiation execution. This attack uses the extra-reduction information to retrieve the modulo used in each private modular exponentiation. This attack does not work when the extra-reductions are implemented in constant time and when the message is blinded or unknown. (see section 3.5.2)

– `ERA 3`: Walter & Thomson [WT01] and Schindler [Sch02] have shown that extra-reductions allow to break RSA-CRT even with message blinding. Indeed, the extra-reduction probability depends on the type of operation (square, multiply, or multiply with a constant), for more details, see section 3.5.3. Regular exponentiation schemes can be regarded as protections since the operation sequence does not depend on the secret.

The "horizontal/vertical" side channel attacks against blinded exponentiation described in [CFG+10, WvWM11, HKT15] use the dependency between the input/output of operands in square and multiply algorithms. Such attacks exploit the *vertical* amplitude of the signal. Our work is thus complementary to these ideas since it considers a novel *horizontal* exploitable bias.

Our ERA is applied on modular arithmetic using the Montgomery form for integer (definition 2.33). The extra-reduction (definition 2.35) is a final subtraction at the end of the multiplication operation. Our ERA works even if the blinded regular modular exponentiation protection is used. The modular exponentiation is regular (definition 3.3), there is no difference for an exponent bit values equals "0" or "1". The input message in our cases is blinded and unknown (section 3.3.3), unlike to classical vertical analysis (section 3.3). Our ERA is composed of two phases:

– The horizontal phase uses the classical technique to find the information in absence or presence of extra-reduction.

– The vertical phase is the novelty of this attack; in fact we exploit the knowledge of the presence and absence of extra-reduction in consecutive operation. We will show that there exists a strong negative correlation between extra-reductions of two consecutive operations, provided that the first feeds the second.

To resume the motivations and differences between our ERA and the others attack, we construct table 5.1[1]. "CRT" means that the attack can be applied when the Chinese Remainder Theorem method is used (✓) or not used (✗). "Key protection" are the blinding exponent method described in section 3.3.3. "DPA protected Blinded Message" is the blinding message method described in section 3.3.3. "SPA protected Constant Time" means that the extra-reduction is compensated by a dummy operation, in order to have constant time execution. Our extra-reduction attack (our ERA) works with all these protections except the exponent blinding.

We show that despite message blinding and regular exponentiation, it is still possible for an attacker to take advantage of extra-reductions: A new bias is found, namely a strong negative correlation between the extra-reduction of two consecutive operations.

[1] ✗: Attack does not work. Countermeasures is efficient. ✓: Attack works with and without the countermeasure.

	CRT	Key Protection	DPA protected Blinded Message	SPA protected Constant Time
"Timing attack" [Koc98]	✗ No	✗ No	✗ No	✗ No
ERA 1 [SKQ01, SW03, ASK05, AS08]	✓ Yes	✗ No	✗ No	✗ No
ERA 2 [Sch15]	✓ Yes	✓ Yes	✗ No	✗ No
ERA 3 [Sch00, WT01, Sch02]	✓ Yes	✗ No	✓ Yes	✗ No
Our ERA [DGD+16]	✓ Yes	✗ No	✓ Yes	✓ Yes

Table 5.1: State-of-the-art of timing attacks and the attacks based on extra-reduction

As shown in this chapter, the bias can be easily leveraged to recover which registers are written to (at line 5 of algorithm 3.1 or at lines 4 and 5 of algorithm 3.2) which eventually leads to retrieve the secret key. The advantages of this method are the following:

- messages are unknown; this captures general situations such as RSA with OAEP or PSS padding and RSA input blinding [Koc96, Sec. 10];
- RSA parameters can be unknown; hence RSA-CRT is also vulnerable;
- all binary exponentiation algorithms are vulnerable, even the regular ones like "Square-and-Multiply-Always" (SMA), "Montgomery Ladder" (ML), etc.;
- our attack can also be applied to Elliptic Curve Cryptography (ECC).

From a mathematical viewpoint, we also provide a comprehensive framework for studying the joint probabilities of extra-reductions in a sequence of multiplies and squares.

The rest of this chapter is organized as follows: In section 5.2, the theoretical rationale for the strong negative correlation between extra-reductions of two chained operations is presented. Section 5.3 shows how this bias can be turned into a key recovery attack. Experimental validations for synthetic and practical traces are in section 5.4. Improvements of our attack and mitigation techniques are discussed in section 5.5 with a dedicated part for ECC cases (section 5.5.4). Section 5.6 concludes.

5.2 A bias to test the dependency of operations

5.2.1 Principle of correlated extra-reductions

In regular exponentiation algorithms, differentiating a multiply from a square does not allow simple side channel analysis to distinguish the value of the exponent bits. Indeed, at every iteration i ($l-1 \geq i > 0$ where i is decremented after each iteration), multiply and square operations are carried out unconditionally. However, the input value of each operation depends on the current exponent bit value k_i.

SMA algorithm. Figure 5.2 illustrates the dependence or independence between the input/output values of multiplication M_i and the input value of the following square S_{i-1} as a function of the bit value k_i during the "Square-and-Multiply-Always" (SMA - algorithm 3.1).

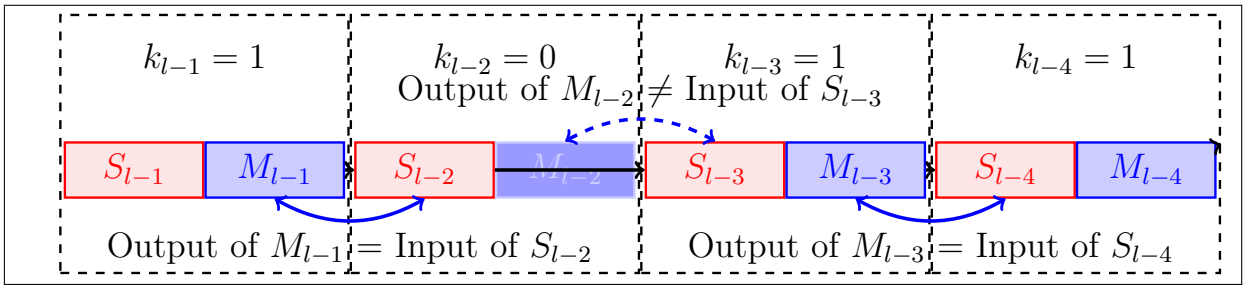


Figure 5.2: Comparison between the output value of multiplication with the input of the following square in the "Square-and-Multiply-Always" exponentiation algorithm (algorithm 3.1).

Intuitively, when the output of M_i is equal to the input of S_{i-1}, we can expect that the extra-reductions in both operations are strongly correlated.

ML algorithm. For the "Montgomery Ladder" (ML - algorithm 3.2) algorithm, the M_i and S_{i-1} operations do not depend directly on the key bit value k_i. As we can see on the figure 5.3, the dependence comes from the bit value k_i and the next bit value k_{i-1}. If the two bit values k_i and k_{i-1} are different then the output of multiplication M_i and the input of square S_{i-1} are equal and yield strongly correlated extra-reductions; in the opposite case they yield uncorrelated extra-reductions.

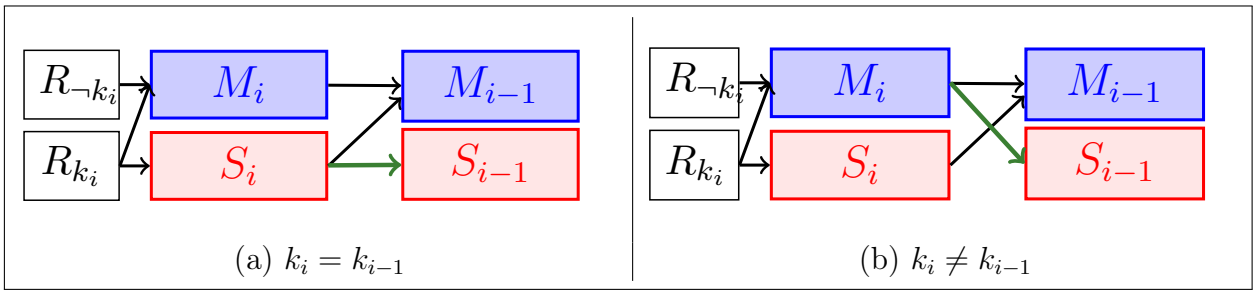


Figure 5.3: Dependency between the consecutive operation and the consecutive key bit value in the "Montgomery Ladder" exponentiation algorithm (algorithm 3.2)

Definition 5.1 (Guess Notation) *Let $\mathcal{G}_i$ be the "guess" Boolean random variable defined to be* `True`*(T) if the output of an operation at iteration i is equal to the input of the next operation at iteration $i-1$, and* `False`*(F) otherwise.*

Also let X_{M_i} be a random variable corresponding to the eXtra-reduction of the MMM multiplication at iteration i and $X_{S_{i-1}}$ be a random variable corresponding to the eXtra-reduction during the MMM square at iteration $(i-1)$.

Then $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T)$ is their joint probability when the output value of the multiplication is equal to the input value of the square, and $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$ is their

joint probability when the output value of the multiplication is not equal to the input value of the square.

The guess value $\mathcal{G}_i$ is linked to the key value depending on the regular exponentiation algorithm. For SMA and for a bit k_i, an attacker is able to estimate the probabilities $\hat{\mathbb{P}}(X_{M_i}, X_{S_{i-1}})$. This probability can be used to find the bit k_i as illustrated in figure 5.2 and explained in section 5.3 below. For ML, $\mathcal{G}_i$ depends on two consecutive key bits as explained also in section 5.3.

We have estimated the joint probabilities $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i)$ using 1.000.000 random values for both SMA and ML algorithms and the example `RSA-1024-p` defined in section 3.5.3.The values of the obtained probabilities are shown in table 5.2.

$(x_{M_i}, x_{S_{i-1}})$	(0,0)	(1,0)	(0,1)	(1,1)
$\mathbb{P}(x_{M_i}, x_{S_{i-1}}\|\mathcal{G}_i = T)$	0.541575	0.191615	0.258276	0.008532
$\mathbb{P}(x_{M_i}, x_{S_{i-1}}\|\mathcal{G}_i = F)$ for SMA	0.612756	0.120158	0.186803	0.080281
$\mathbb{P}(x_{M_i}, x_{S_{i-1}}\|\mathcal{G}_i = F)$ for ML	0.586105	0.147246	0.213521	0.053128

Table 5.2: Example of probabilities of eXtra-reduction X_{M_i} of multiply operation and $X_{S_{i-1}}$ of square operation knowing the Boolean value $\mathcal{G}_i$ for `RSA-1024-p`. The first line (correct guess) is applicable for both SMA and ML.

Furthermore using a rounded values to 10^{-3}, notice that $\mathbb{P}(X_M|\mathcal{G} = T) = \mathbb{P}(X_M|\mathcal{G} = F)$ equals to $\mathbb{P}(X_M)$ computed in table 3.1 using $\mathbb{P}(X_M|\mathcal{G}) = \mathbb{P}(X_{M_i}, X_{S_{i-1}} = 0|\mathcal{G} = T) + \mathbb{P}(X_{M_i}, X_{S_{i-1}} = 1|\mathcal{G} = T)$, and $\mathbb{P}(X_S|\mathcal{G} = T) = \mathbb{P}(X_S|\mathcal{G} = F)$ equals to $\mathbb{P}(X_S)$ computed in table 3.1, using $\mathbb{P}(X_S|\mathcal{G}) = \mathbb{P}(X_M = 0, X_S|\mathcal{G} = T) + \mathbb{P}(X_M = 1, X_S|\mathcal{G} = T)$ since irrespective of $\mathcal{G}$, the presence of an extra-reduction does not depend whether the square depends with the next multiplication or not.

Remark It is important to notice that for each $(x_{M_i}, x_{S_{i-1}}) \in \{0, 1\}^2$, the conditional joint probabilities are distinct: $\mathbb{P}(X_{M_i} = x_{M_i}, X_{S_{i-1}} = x_{S_{i-1}}|\mathcal{G}_i = F) \neq \mathbb{P}(X_{M_i} = x_{M_i}, X_{S_{i-1}} = x_{S_{i-1}}|\mathcal{G}_i = T)$. Also for $\mathcal{G}_i = F$ in ML, it can be observed that $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i) = \frac{p}{4R} \times \frac{p}{3R} = \mathbb{P}(X_{M_i}) \times \mathbb{P}(X_{S_{i-1}})$, which is consistent with the fact the two operations X_{M_i} and $X_{S_{i-1}}$ should be independent since they are completely unrelated.

It should be emphasized that the leakage identified in table 5.2 is fairly large, since the Pearson correlations ρ of the two random variables are[2]:

$$\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T) \approx -0.2535, \tag{5.1}$$
$$\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F) \approx +0.1510 \text{ in SMA}, \tag{5.2}$$
$$\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F) \approx -0.0017 \text{ in ML}. \tag{5.3}$$

To the best of our knowledge, such correlations have not been observed previously. A few observations are in order:

[2] $\rho(X_{M_i}, X_{S_{i-1}}) = \frac{\mathbb{C}ov(X_{M_i}, X_{S_{i-1}})}{\sigma_{X_{M_i}} \sigma_{X_{S_{i-1}}}} = \frac{\mathbb{P}(X_{M_i}=1, X_{S_{i-1}}=1) - (\mathbb{P}(X_{M_i}=1) \times \mathbb{P}(X_{S_{i-1}}=1))}{\sqrt{\mathbb{P}(X_{M_i}=1)(1-\mathbb{P}(X_{M_i}=1))}\sqrt{\mathbb{P}(X_{S_{i-1}}=1)(1-\mathbb{P}(X_{S_{i-1}}=1))}}$.

– when a square follows a multiply, and if there has been an extra-reduction in the multiplication, the result should be short, hence there is less chance for an extra-reduction to occur in the following square. This accounts for the negative correlation $\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T)$;

– from figure 5.2 iteration $i = l - 2$ where $k_i = 0$, we can see that one input of the multiplication M_i equals the input of the following squaring S_{i-1}. Since a square and a multiplication share a common operand, provided it is sufficiently large, both operations are likely to have an extra-reduction at the same time, which accounts for the positive correlation $\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$ for SMA;

– when a square and a multiply handle independent data, the extra-reductions are clearly also independent of each other, which explains the small value of $\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$ for ML.

As explained next, when extra-reductions can be detected reliably, the data-flow can be analyzed accurately thereby defeating regular exponentiation protections. In particular, knowing the global timing of the algorithm is insufficient for the attack to succeed, because the attacker must be able to relate the extra-reductions to a target operation i ($l - 1 \geq i \geq 0$).

5.2.2 Methodology to analyze the bias

In order to estimate the probability $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i)$, we first determine the distribution of the output value after one MMM (following the method described by Sato et al.[SST04]) and then compute the joint probability for each case.

Let A, B be two independent random variables uniformly distributed in $[0, p)$ (represented in Montgomery form); let C be equal to the MMM product of A and B and U corresponds to the MMM product of A and B before eXtra-reduction (if any). Variables C and U coincide with that of algorithm 2.8. As a matter of fact, an attacker cannot observe values, only extra-reductions which occur during Montgomery reduction (at line 4 of algorithm 2.8). We use notations $\mathbb{P}$ for probabilities and f for probability density functions (p.d.f.'s).

Figure 5.4 shows histograms for C and U obtained from one million simulations; the binning consists of 100 bins of the interval $[0, 2p)$. It can be observed that

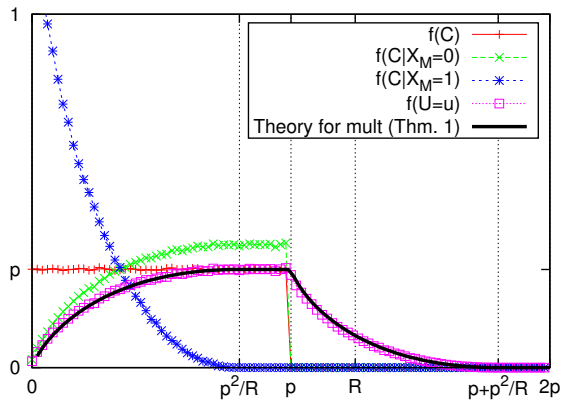
– the p.d.f. of C is uniform on $[0, p)$;

– the p.d.f. of U is a piece wise continuous function composed of a strictly increasing part, a constant part and a strictly decreasing part;

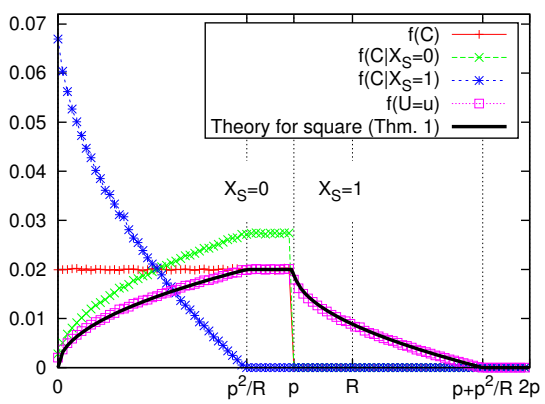
– the two conditional p.d.f.'s of C knowing $X_{M_i} \in \{0, 1\}$ (resp. $X_{S_i} \in \{0, 1\}$) are not uniform;

– for $c \in [0, p)$, one has $f(C = c) = f(U = c) + f(U = c + p)$ by definition of U;

– the maximum value of U is $p + p^2/R$, which is strictly smaller than $2p$.



(a) Multiplication



(b) Square

Figure 5.4: Distribution of the output value of Montgomery multiplication (*left*) and square (*right*) for `RSA-1024-p`.

Recall that we use the Montgomery reduction described in algorithm 2.8, where the reduction modulo p is carried out after every multiplication. This is also the case in [Sch00, Sch02], but *not* in [WT01, SW03] where the multiplicands lie in $[0, R)$. To complement those works, we now derive a closed-form expression of the output distribution of the Montgomery multiplication product and square (not found in [Sch00, Sch02]).

5.2.3 Mathematical derivations

This section provides a mathematical justification of the biases observed in table 5.2. In particular, it shows that such biases hold for all values of p and $R = 2^{\lceil \log_2(p) \rceil}$. Our closed-form expressions are derived as limits in distribution when $p \to +\infty$ that we shall write as approximations.

Theorem 5.2 (P.d.f. of MMM Before Extra-Reduction) *Asymptotically when modulus p is large, the result of a Montgomery multiplication before the final extra-reduction (at line 2 of algorithm 2.8) have piecewise p.d.f. given by*

$$f_U(u) = \begin{cases} \frac{Ru}{p^3}\left(1 - \ln(\frac{Ru}{p^2})\right) & \text{if } 0 \le u \le \frac{p^2}{R}; \\ \frac{1}{p} & \text{if } \frac{p^2}{R} \le u \le p; \\ \frac{1}{p} - \frac{R(u-p)}{p^3}\left(1 - \ln(\frac{R(u-p)}{p^2})\right) & \text{if } p \le u \le p + \frac{p^2}{R}; \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

The corresponding p.d.f. for the square is also in four pieces with the same intervals for u, and differs only from the multiplication in that it is equal to $\sqrt{Ru}/p^2$ when $0 \le u \le \frac{p^2}{R}$, and $1/p - \sqrt{R(u-p)}/p^2$ when $p \le u \le p + \frac{p^2}{R}$.

Proof See proof in following section 5.2.4. □

The theoretical values of theorem 5.2 nicely superimpose with experimentally estimated p.d.f.'s as shown in figure 5.4.

Corollary 5.3 *The expressions of theorem 5.2 allow to prove (independently) the result of proposition 3.5 in the case of the multiplication and of the square.*

Proof We recall that the presence of an extra-reduction of a multiplication operation is noted X_M and for a square X_S. Using the case 3 of equation (5.4) and the change of variable $u \leftarrow u - p$, we notice that, for multiplication:

$$\begin{aligned}\mathbb{P}(X_M = 1) &= \int_0^{p^2/R} \frac{1}{p} - \frac{Ru}{p^3}\left(1 - \ln\left(\frac{Ru}{p^2}\right)\right) \mathrm{d}u \\ &= \frac{p}{R} - \frac{R}{p^3}\left[\frac{Ru^2}{2p^3}\ln\frac{Ru}{p^2} - \frac{3Ru^2}{4p^3}\right]_0^{p^2/R} = \frac{p}{4R} \ .\end{aligned}$$

And regarding the square:

$$\mathbb{P}(X_S = 1) = \int_0^{p^2/R} \frac{1}{p} - \frac{\sqrt{Ru}}{p^2}\,\mathrm{d}u = \frac{p}{R} - \frac{\sqrt{R}}{p^2}\left[\frac{2}{3}u^{3/2}\right]_0^{p^2/R} = \frac{p}{3R} \ .$$

□

Theorem 5.4 (Joint Probability of Extra-Reduction) *The following joint probabilities in multiplication followed by a square do not depend on the iteration index i, where $l - 1 \geq i > 0$.*

When $\mathcal{G}_i = T$:

$\mathbb{P}(x_{M_i}, x_{S_{i-1}})$	$x_{S_{i-1}} = 0$	$x_{S_{i-1}} = 1$
$x_{M_i} = 0$	$1 - \frac{7}{12}\frac{p}{R} + \frac{1}{48}\left(\frac{p}{R}\right)^4$	$\frac{p}{3R} - \frac{1}{48}\left(\frac{p}{R}\right)^4$
$x_{M_i} = 1$	$\frac{p}{4R} - \frac{1}{48}\left(\frac{p}{R}\right)^4$	$\frac{1}{48}\left(\frac{p}{R}\right)^4$

When $\mathcal{G}_i = F$ in SMA:

$\mathbb{P}(x_{M_i}, x_{S_{i-1}})$	$x_{S_{i-1}} = 0$	$x_{S_{i-1}} = 1$
$x_{M_i} = 0$	$1 - \frac{7}{12}\frac{p}{R} + \frac{1}{8}\left(\frac{p}{R}\right)^2$	$\frac{p}{3R} - \frac{1}{8}\left(\frac{p}{R}\right)^2$
$x_{M_i} = 1$	$\frac{p}{4R} - \frac{1}{8}\left(\frac{p}{R}\right)^2$	$\frac{1}{8}\left(\frac{p}{R}\right)^2$

When $\mathcal{G}_i = F$ in ML:

$\mathbb{P}(x_{M_i}, x_{S_{i-1}})$	$x_{S_{i-1}} = 0$	$x_{S_{i-1}} = 1$
$x_{M_i} = 0$	$1 - \frac{7}{12}\frac{p}{R} + \frac{1}{12}\left(\frac{p}{R}\right)^2$	$\frac{p}{3R} - \frac{1}{12}\left(\frac{p}{R}\right)^2$
$x_{M_i} = 1$	$\frac{p}{4R} - \frac{1}{12}\left(\frac{p}{R}\right)^2$	$\frac{1}{12}\left(\frac{p}{R}\right)^2$

Proof See proof in following section 5.2.4. □

It can be easily checked that theorem 5.4 accurately matches experimental probability estimations given in table 5.2.

Corollary 5.5 *The corresponding correlation coefficients are*

$$\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T) = \frac{\frac{p^4}{48R^4} - \frac{p^2}{12R^2}}{\sqrt{\frac{p}{4R}\left(1-\frac{p}{4R}\right)\frac{p}{3R}\left(1-\frac{p}{3R}\right)}},$$

$$\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F) = \frac{\frac{p^2}{24R^2}}{\sqrt{\frac{p}{4R}\left(1-\frac{p}{4R}\right)\frac{p}{3R}\left(1-\frac{p}{3R}\right)}} \text{ in SMA,}$$

$$\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F) = 0 \text{ in ML.}$$

Proof Apply Pearson's correlation definition on the results of theorem 5.4. □

When the guess is correct, $\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T)$ is negative and increasingly negative as p/R increases, where

$$-\frac{3}{16}\sqrt{\frac{5}{7}} \approx -0.158 \quad \leq \quad \rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T) \quad \leq \quad -\frac{3}{4\sqrt{6}} \approx -0.306.$$

When the guess is incorrect, either the correlation is null (in the case of ML), or it is positive and increasing with p/R, where for $1/2 \leq p/R \leq 1$,

$$\frac{1}{2\sqrt{5\times 7}} \approx 0.085 \quad \leq \quad \rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F) \quad \leq \quad \frac{1}{2\sqrt{6}} \approx 0.204.$$

The variations of the correlation coefficients between X_{M_i} and $X_{S_{i-1}}$ in the three scenarios of corollary (5.5) are plotted in figure 5.5.

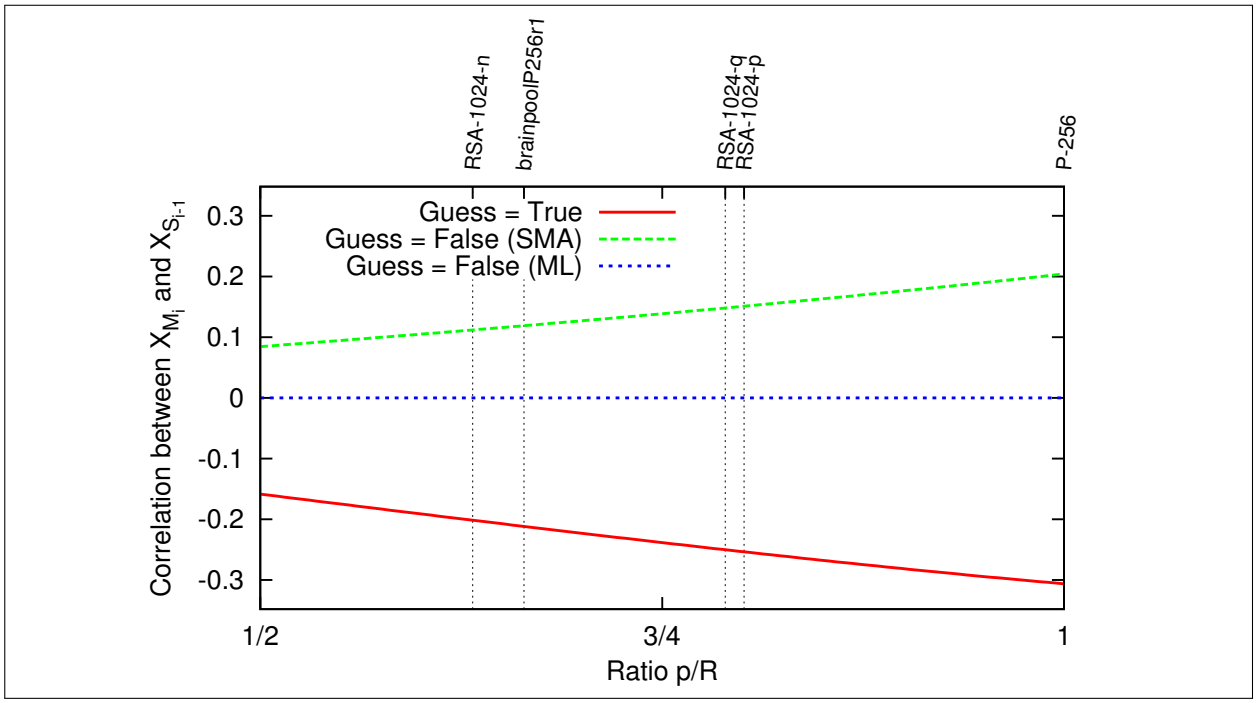


Figure 5.5: Pearson's correlation between X_{M_i} and $X_{S_{i-1}}$.

Figure 5.5 shows that the correlation difference between guesses `True`/`False`is greater for the SMA algorithm than for the ML algorithm. Thus our attack on SMA should outperform that on ML. Also notice that the larger the ratio p/R is , the larger the correlation difference is; hence, we expect `P-256` to be easier to break than `brainpoolP256r1` with our attack.

5.2.4 Proof of theorems 5.2 and 5.4

Before proving theorems, we need the following technical lemmas 5.6, 5.7, 5.8, 5.9 and 5.10.

Technical lemmas

Let A and B two independent discrete random variables uniformly distributed on $\{0,\dots,p-1\}$. In the sequel, we are interested in asymptotic convergence in distribution when $p\to+\infty$. In general, it is known that a series of discrete random variables X_p converges in distribution to X (continuous random variable, having a density function), if and only if there is a convergence of cumulative density functions (c.d.f.):

$$\mathbb{P}(X_p\le x)\to\mathbb{P}(X\le x)\qquad\text{when } p\to+\infty.$$

This is why we will work on c.d.f. Besides, it is well known that if A is uniformly distributed on $[0,p-1]$, then A/p converges in distribution to $\mathcal{U}([0,1])$ when $p\to+\infty$. Indeed, for all $x\in(0,1)$,

$$\mathbb{P}\left(\frac{A}{p}\le x\right)=\sum_{0\le a<px}\frac{1}{p}=\frac{\lfloor px\rfloor}{p}\to x=\int_0^x \mathrm{d}u.$$

Now, it is incorrect to write that for $0\le a<p$, the limit of $\mathbb{P}(A\le a)$ is $\frac{a}{p}$ (since $p\to\infty$). However, we will use this abuse of notation in the sequel.

Lemma 5.6 *Let A,B two random variables uniformly distributed over $[0,p)$. Let x,y two values in $[0,p^2)$. Then, in the limit $p\to+\infty$, we have:*

$$f(x)=\frac{1}{p^2}\ln\frac{p^2}{x}\qquad\text{where } f \text{ is the density function of } AB, \tag{5.5}$$

$$f(y)=\frac{1}{2p\sqrt{y}}\qquad\text{where } f \text{ is the density function of } A^2, \tag{5.6}$$

$$f(x,y)=\frac{1}{2p^2y}\mathbb{1}_{[0,\frac{\sqrt{y}}{p}]}\left(\frac{x}{p^2}\right)\qquad\text{where } f \text{ is the density function of } (AB,A^2). \tag{5.7}$$

Proof Case of the product. As A and B are independent, (A,B) has a uniform law on $\{0,\dots,p-1\}^2$. Hence the convergence in distribution of the pair $(A/p,B/p)$ to a uniform law on $[0,1]^2$. Thus, for all $x\in(0,1)$,

$$\mathbb{P}\left(\frac{AB}{p^2}\le x\right)\to\iint_{[0,1]^2}\mathbb{1}_{uv<x}\,\mathrm{d}u\,\mathrm{d}v\qquad\text{when } p\to+\infty.$$

The convergence is illustrated in figure 5.6 as the proportion of points below the hyperbola curve represented in red.

As $uv<x\iff v<x/u$ and $x/u<1\iff u>x$, we have

$$\iint_{[0,1]^2}\mathbb{1}_{uv<x}\,\mathrm{d}u\,\mathrm{d}v=\int_0^1\min\left(\frac{x}{u},1\right)\mathrm{d}u=\int_0^x\mathrm{d}u+\int_x^1 x\frac{\mathrm{d}u}{u}=x(1-\ln x).$$

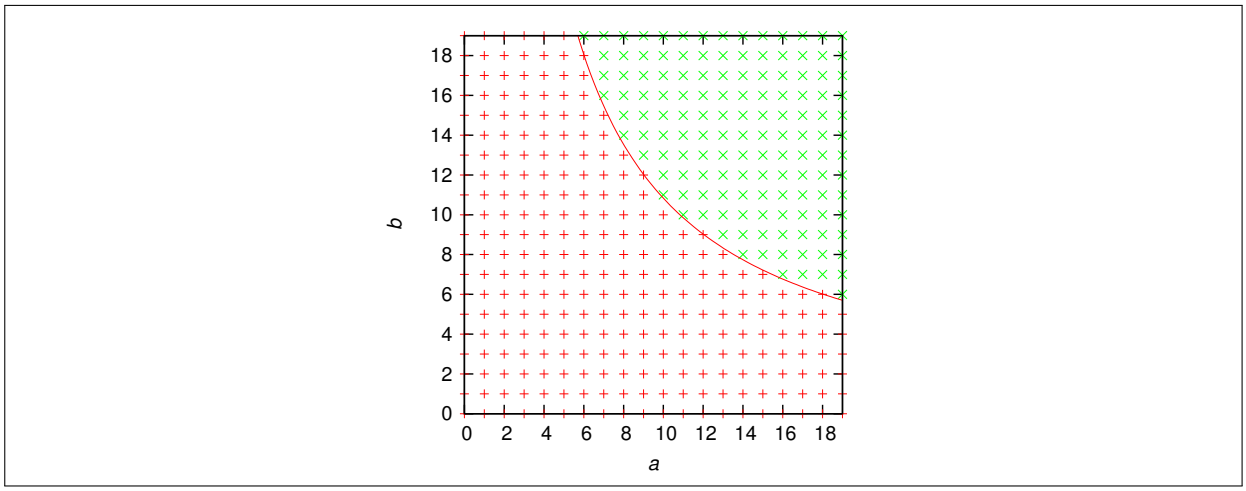


Figure 5.6: Illustration of $\{(a,b) \in \{0, p-1\}^2 \mid ab < p^2 x\}$ for prime $p = 19$ and $x = 0.3$

Thus, the limit distribution of AB/p^2 has for density the derivative:

$$f(x) = \ln \frac{1}{x}.$$

As $AB = p^2(AB/p^2)$, a variable change yields equation (5.5).

Case of the square. For all $y \in (0,1)$,

$$\mathbb{P}\left(\frac{A^2}{p^2} < y\right) = \mathbb{P}\left(\frac{A}{p} < \sqrt{y}\right) \to \sqrt{y} \qquad \text{when } p \to +\infty.$$

The limit distribution has for density the derivative:

$$f(y) = \frac{1}{2\sqrt{y}},$$

hence equation (5.6) by change of variable $A^2 = p^2(A^2/p^2)$.

Case of the pair multiplication and square. We have, for all $x, y \in (0,1)$:

$$\mathbb{P}\left(\frac{AB}{p^2} < x \;\middle|\; \frac{A^2}{p^2} = y\right) = \mathbb{P}\left(\frac{B}{p} < \frac{x}{\sqrt{y}} \;\middle|\; \frac{A^2}{p^2} = y\right) = \mathbb{P}\left(\frac{B}{p} < \frac{x}{\sqrt{y}}\right) \to \min\left(\frac{x}{\sqrt{y}}, 1\right)$$

because A and B are independent and because B is uniform. The conditional distribution of $(AB/p^2 \mid A^2/p^2 = y)$ has in the limit the (uniform) density:

$$f(x) = \frac{1}{\sqrt{y}} \mathbb{1}_{[0,\sqrt{y}]}(x)$$

and so, the joint probability $(AB/p^2, A^2/p^2)$ in $(x,y) \in [0,1]^2$ has the following limit

$$\mathbb{P}\left(AB/p^2 = x, A^2/p^2 = y\right) \to \frac{1}{\sqrt{y}} \mathbb{1}_{[0,\sqrt{y}]}(x) \frac{1}{2\sqrt{y}} = \frac{1}{2y} \mathbb{1}_{[0,\sqrt{y}]}(x) \quad \text{(by (5.6))}.$$

Again, a variable change yields equation (5.7). □

Lemma 5.7 *Let A a random variable defined on $[0,p]$, with density f. Then the probability density function of A^2 in $z \in [0,p^2]$ is equal to $f(\sqrt{z}) = \frac{1}{2\sqrt{z}}$.*

Proof Use (5.6) in lemma 5.6 in a variable change. □

Lemma 5.8 *Let u an integer such that $0 \leq u < p$. The set $\mathcal{C}_u = \{z, 0 \leq z < p^2, \text{ s.t. } z + (zv \bmod R)p = Ru\}$ is equal to $\mathcal{C}_u = \{(Ru \bmod p) + ip, \text{ where } 0 \leq i \leq \min\left(p, \lfloor \frac{Ru}{p} \rfloor\right)\}$.*

Proof Let z such that $z+(zv \bmod R)p = Ru$. Clearly, we have $(z \bmod p) = (Ru \bmod p)$, hence $\mathcal{C}_u \subseteq \{((Ru \bmod p) + ip, \text{ where } i \in \mathbb{N}\}$. But given the bounds on z, we have $0 \leq i < p$. Let us precise which values of i make $(Ru \bmod p) + ip$ belong to $\mathcal{C}_u$.

We have $(Ru \bmod p) + ip + (((Ru \bmod p)v + ipv) \bmod R)p = Ru$, hence, as $Ru - (Ru \bmod p) = \lfloor \frac{Ru}{p} \rfloor$, $i + (((Ru \bmod p)v - i) \bmod R) = \lfloor \frac{Ru}{p} \rfloor$. In this expression, $(Ru \bmod p) = Ru - \lfloor \frac{Ru}{p} \rfloor p$. Let us denote $1 + vp = \ell R$, where $0 < \ell < p$. We have

$$\begin{aligned}(((Ru \bmod p)v - i) \bmod R) &= (Ruv - \left\lfloor \frac{Ru}{p} \right\rfloor (\ell R - 1) - i \bmod R) \\ &= (\left\lfloor \frac{Ru}{p} \right\rfloor - i \bmod R) \\ &= \begin{cases} \lfloor \frac{Ru}{p} \rfloor - i & \text{if } 0 \leq i \leq \lfloor \frac{Ru}{p} \rfloor, \\ R + \lfloor \frac{Ru}{p} \rfloor - i & \text{if } i > \lfloor \frac{Ru}{p} \rfloor. \end{cases}\end{aligned}$$

Consequently, the condition is met if and only if $0 \leq i \leq \lfloor \frac{Ru}{p} \rfloor$. Hence the proof of the lemma, as i is upper bounded both by p and $\lfloor \frac{Ru}{p} \rfloor$. □

Lemma 5.9 (Approximation of finite summations) *Let I a large number (comparable to p). When $I \to +\infty$, we have,*

$$\sum_{i=0}^{I} i^\alpha \to \frac{1}{1+\alpha} I^{1+\alpha} \text{ for } \alpha \in \mathbb{R} \setminus \{-1\}, \tag{5.8}$$

$$\sum_{i=1}^{I} 1/i \to \ln(I), \tag{5.9}$$

$$\sum_{i=0}^{I} \ln(i) \to I \ln(I) - I. \tag{5.10}$$

Proof When $I \to +\infty$, $\sum_{i=0}^{I} i^\alpha \to \int_0^I x^\alpha \, \mathrm{d}x = \frac{1}{1+\alpha} I^{1+\alpha}$. Similarly, $\sum_{i=1}^{I} 1/i \to \int_1^I 1/x \, \mathrm{d}x = \ln(I)$. Eventually, by Stirling formula, we have that $\ln I! = I \ln I - I + \mathcal{O}(\ln I) \to I \ln I - I$. □

Lemma 5.10 (Miscellaneous approximations) *When $p \to +\infty$, we have*

$$\left\lfloor \frac{Ru}{p} \right\rfloor \to \frac{Ru}{p} \quad \text{for } 0 \leq u \leq p, \tag{5.11}$$

$$\frac{(Ru \bmod p) + ip}{p} \to i \quad \text{for } i \gg 1. \tag{5.12}$$

Proof The first equation arises from $\lim_{p \to +\infty} \lfloor p \rfloor / p \to 1^-$, whereas the second one holds all the more for large values of i, since $(Ru \bmod p) < p \ll ip$ when $i \gg 1$. □

Proof of theorem 5.2

Here is the proof of theorem 5.2:

Proof As explained in section 5.2.4, $\mathcal{P}(U = u)$ tends to a density $f(U = u)$ when $p \to +\infty$.

Case $0 \leq u \leq p$: let U be the result of MMM before the final reduction (definition at line 2 of algorithm 2.8). We have, in the limit $p \to +\infty$:

$$\begin{aligned}
\mathbb{P}(U = u) &= \sum_{a=0}^{p-1} \sum_{b=0}^{p-1} \mathbb{P}(A = a, B = b)\ \mathbb{1}_{ab+(abv \bmod R)p=Ru} \\
&= \frac{1}{p^2} \sum_{z=1}^{p^2} \ln\left(\frac{p^2}{z}\right) \mathbb{1}_{z+(zv \bmod R)p=Ru} && \text{(denote } z = ab \text{ and apply equation (5.5) of lemma 5.6)} \\
&= \frac{1}{p^2} \sum_{i=1}^{\min\left(p, \lfloor \frac{Ru}{p} \rfloor\right)} \ln\left(\frac{p^2}{(Ru \bmod p) + ip}\right) && \text{(by lemma 5.8)} \\
&\approx \frac{1}{p^2} \sum_{i=1}^{\min\left(p, \frac{Ru}{p}\right)} \ln\left(\frac{p}{i}\right) && \text{(by lemma 5.10)} \\
&= \frac{\min\left(p, \frac{Ru}{p}\right)}{p^2} \left(\ln(p) - \ln\left(\min\left(p, \frac{Ru}{p}\right)\right) + 1\right) && \text{(by equation (5.10) in lemma 5.9)} \\
&= \begin{cases} \frac{uR}{p^3}\left(1 - \ln\frac{uR}{p^2}\right) & \text{if } p \leq \frac{Ru}{p}, \text{ i.e., } u \geq \frac{p^2}{R}, \\ \frac{1}{p} & \text{if } p > \frac{Ru}{p}, \text{ i.e., } u < \frac{p^2}{R}. \end{cases}
\end{aligned}$$

Case $p < u < 2p$: By the definition of U and C, for each $c \in [0, p)$ we have

$$\mathbb{P}(C = c) = \mathbb{P}(U = c) + \mathbb{P}(U = c + p).$$

$C = ABR^{-1} \bmod p$, so C is a random variable uniformly distributed over $[0, p)$.

Then, $p < u < 2p$, we have, by the definition of C (at line 4 or 6 of algorithm 2.8):

$$\mathbb{P}(U = u) = \mathbb{P}(C = u - p) - \mathbb{P}(U = u - p) = 1/p - \mathbb{P}(U = u - p).$$

In the case where $A = B$, the demonstration is similar. For $0 \leq u \leq p$, we have:

$$\begin{aligned}
\mathbb{P}(U = u) &= \sum_{z=0}^{p^2} \frac{1}{2p\sqrt{z}} \mathbb{1}_{z+(zv \bmod R)p=Ru} && \text{(denote } z = a^2 \text{ and apply (5.6) of lemma 5.6)} \\
&= \frac{1}{2p} \sum_{i=1}^{\min\left(p, \lfloor \frac{Ru}{p} \rfloor\right)} ((Ru \bmod p) + ip)^{-1/2} && \text{(by lemma 5.8)} \\
&\approx \frac{1}{2p} \sum_{i=1}^{\min\left(p, \frac{Ru}{p}\right)} (ip)^{-1/2} && \text{(by lemma 5.10)}
\end{aligned}$$

$$= \frac{1}{p^{3/2}}\sqrt{\min\left(p, \frac{Ru}{p}\right)} \quad \text{(by equation (5.8) for } \alpha = -1/2 \text{ in lemma 5.9)}$$
$$= \begin{cases} \frac{\sqrt{uR}}{p^2} & \text{if } p \leq \frac{Ru}{p}, \text{ i.e., } u \geq \frac{p^2}{R}, \\ \frac{1}{p} & \text{if } p > \frac{Ru}{p}, \text{ i.e., } u < \frac{p^2}{R}. \end{cases}$$

□

Proof of theorem 5.4

Here is the proof of theorem 5.4:

Proof Let A a random variable resulting from an MMM with an extra-reduction, and denote by f its density function. Then we have $f(a) = 1/p - aR/p^3(1 - \ln(aR/p^2))$ if $0 \leq a \leq p^2/R$, and 0 if $p^2/R \leq a \leq p$. We denote by U the value of A^2 before the final reduction.

First case, where $0 < u < p$ (lemma 5.8 applies): We have:

$$\begin{aligned}
\mathbb{P}(U = u) &= \sum_{z=0}^{p^2} f(z)\frac{1}{2p\sqrt{z}}\, \mathbb{1}_{z+(zv \bmod R)p=Ru} \quad \text{(use lemma 5.7)} \\
&= \sum_{i=1}^{I} f(\sqrt{(Ru \bmod p) + ip})\frac{1}{2\sqrt{(Ru \bmod p) + ip}} \quad \text{(by lemma 5.8)} \\
&\approx \sum_{i=1}^{I} f(\sqrt{ip})\frac{1}{2\sqrt{ip}}, \qquad \text{(by approximation equation (5.12) of lemma 5.10)}
\end{aligned} \tag{5.13}$$

where the upper bound I is $\min(p, \frac{Ru}{p}, \frac{1}{p}\left(\frac{p^2}{R}\right)^2)$. As $\frac{1}{p}\left(\frac{p^2}{R}\right)^2 = p(p/R)^2 < p$, we have:

$$I = \begin{cases} \frac{p^3}{R^2} & \text{if } \frac{Ru}{p} > \frac{p^3}{R^2}, \text{ i.e., } u > \frac{p^4}{R^3} = p(p/R)^3 \text{ (but with constraint } p > u), \\ \frac{Ru}{p} & \text{otherwise.} \end{cases}$$

We can rewrite equation (5.13) (where the dependency in u is via I) as

$$\begin{aligned}
\mathbb{P}(U = u) &= \frac{1}{2\sqrt{p}}\sum_{i=1}^{I}\frac{1}{\sqrt{i}}\left(\frac{1}{p} - \sqrt{i}\frac{R}{p^{5/2}}(1 - \ln(\frac{R}{p^{3/2}}) - \frac{1}{2}\ln(i))\right) \\
&= \frac{1}{2p^{3/2}}\sum_{i=1}^{I}\frac{1}{\sqrt{i}} - \frac{R}{2p^3}\sum_{i=1}^{I}\left(1 - \ln(\frac{R}{p^{3/2}}) - \frac{1}{2}\ln(i)\right) \\
&= \frac{\sqrt{I}}{p^{3/2}} - \frac{IR}{2p^3}\left(1 - \ln(\frac{R}{p^{3/2}})\right) + \frac{R}{4p^3}\sum_{i=1}^{I}\ln(i) \qquad \text{(Using equation (5.8) of lemma 5.9 for } \alpha = 1/2) \\
&= \frac{\sqrt{I}}{p^{3/2}} - \frac{IR}{2p^3}\left(1 - \ln(\frac{R}{p^{3/2}})\right) + \frac{RI}{4p^3}(\ln(I) - 1) \quad \text{(Using equation (5.10) of lemma 5.9)} \\
&= \frac{\sqrt{I}}{p^{3/2}} + \frac{IR}{4p^3}\left(-2 + 2\ln(\frac{R}{p^{3/2}}) + \ln(I) - 1\right)
\end{aligned}$$

$$= \frac{\sqrt{I}}{p^{3/2}} + \frac{IR}{4p^3}\left(\ln\left(\frac{R^2 I}{p^3}\right) - 3\right)$$

So, when $p > u > p(p/R)^3$, we have:

$$\mathbb{P}(U = u) = \frac{1}{R} + \frac{1}{4R}(-3) = \frac{1}{4R}.$$

And when $u < p(p/R)^3$, we have:

$$\mathbb{P}(U = u) = \frac{\sqrt{Ru}}{p^2} + \frac{R^2 u}{4p^4}\left(\ln\left(\frac{R^3 u}{p^4}\right) - 3\right).$$

Second case, where $p < u < 2p$):

Like in theorem 5.2, we have $\mathbb{P}(U = u) = \frac{1}{p} - \mathbb{P}(U = u - p)$.

Now, we have that (first case situation only):

$$\begin{aligned}
&\mathbb{P}(X_{M_i} = 1, X_{S_{i-1}} = 0 \mid \mathcal{G}_i = T) = \int_0^p \mathbb{P}(U = u)\,\mathrm{d}u \\
&= \int_0^{p^4/R^3} \frac{\sqrt{Ru}}{p^2} + \frac{R^2 u}{4p^4}\left(\ln\left(\frac{R^3 u}{p^4}\right) - 3\right)\mathrm{d}u + \frac{p}{4R}\left(1 - \frac{p^3}{R^3}\right) \\
&= \frac{p^4}{R^4}\int_0^1 \sqrt{u'}\,\mathrm{d}u' + \frac{p^4}{4R^4}\int_0^1 u'(\ln u' - 3)\,\mathrm{d}u' + \frac{p}{4R}\left(1 - \frac{p^3}{R^3}\right) \qquad (\triangleright\ u' = uR^3/p^4) \\
&= \frac{p^4}{R^4}\left[\frac{2}{3}u'^{3/2}\right]_0^1 + \frac{p^4}{4R^4}\left[\frac{1}{2}u'^2\ln(u') - \frac{1}{4}u'^2\right]_0^1 - \frac{3p^4}{4R^4}\left[\frac{u'^2}{2}\right]_0^1 + \frac{p}{4R}(1 - \frac{p^3}{R^3}) \\
&= \frac{p}{4R} + \frac{p^4}{R^4}\left(\frac{2}{3} - \frac{1}{16} - \frac{3}{8} - \frac{1}{4}\right) = \frac{p}{4R} - \frac{p^4}{48R^4}.
\end{aligned}$$

Other entries of the table of theorem 5.4 corresponding to $\mathcal{G}_i = T$ can be deduced from the partial probabilities given in proposition 3.5 namely

$$\begin{aligned}
&\mathbb{P}(X_{M_i}{=}0, X_{S_{i-1}}{=}0|\mathcal{G}_i = T) + \mathbb{P}(X_{M_i}{=}1, X_{S_{i-1}}{=}0|\mathcal{G}_i = T) = \mathbb{P}(X_{S_{i-1}}{=}0) = 1 - \frac{p}{3R}, \\
&\mathbb{P}(X_{M_i}{=}0, X_{S_{i-1}}{=}1|\mathcal{G}_i = T) + \mathbb{P}(X_{M_i}{=}1, X_{S_{i-1}}{=}1|\mathcal{G}_i = T) = \mathbb{P}(X_{S_{i-1}}{=}1) = \frac{p}{3R}, \\
&\mathbb{P}(X_{M_i}{=}0, X_{S_{i-1}}{=}0|\mathcal{G}_i = T) + \mathbb{P}(X_{M_i}{=}0, X_{S_{i-1}}{=}1|\mathcal{G}_i = T) = \mathbb{P}(X_{M_i}{=}0) = 1 - \frac{p}{4R}, \\
&\mathbb{P}(X_{M_i}{=}1, X_{S_{i-1}}{=}0|\mathcal{G}_i = T) + \mathbb{P}(X_{M_i}{=}1, X_{S_{i-1}}{=}1|\mathcal{G}_i = T) = \mathbb{P}(X_{M_i}{=}1) = \frac{p}{4R}.
\end{aligned}$$

Regarding the case $\mathcal{G}_i = F$ for SMA, we shall compute the probability density that a multiplication before extra-reduction is equal to u_1 and that a square is equal to u_2, whereby one operand of the multiplication is the input of the square and the second operand is uniformly distributed over $[0, p]$. Let us assume $0 \leq u_1, u_2 < p$. This density is equal to:

$$
\begin{aligned}
&f(U_1 = u_1, U_2 = u_2) \\
&= \sum_{x=0}^{p}\sum_{y=0}^{p} f(AB = x, A^2 = y)\ \mathbb{1}_{x+(xv \bmod R)p = u_1 R}\ \mathbb{1}_{y+(yv \bmod R)p = u_2 R} \\
&= \sum_{x=0}^{p}\sum_{y=0}^{p} \frac{1}{2p^2 y}\mathbb{1}_{[0,\frac{\sqrt{y}}{p}]}\left(\frac{x}{p^2}\right)\ \mathbb{1}_{x+(xv \bmod R)p = u_1 R}\ \mathbb{1}_{y+(yv \bmod R)p = u_2 R} \qquad \text{(by equation (5.7) of lemma 5.6)} \\
&= \sum_{i_1=0}^{\min(\lfloor Ru_1/p \rfloor, p)} \sum_{i_2=0}^{\min(\lfloor Ru_2/p \rfloor, p)} \frac{\mathbb{1}_{\left[0,\frac{\sqrt{(Ru_2 \bmod p)+i_2 p}}{p}\right]}\left(\frac{(Ru_1 \bmod p)+i_1 p}{p^2}\right)}{2p^2((Ru_2 \bmod p)+i_2 p)} \qquad \text{(by lemma 5.8)} \\
&\approx \frac{1}{2p^3} \sum_{i_2=0}^{\min(\lfloor Ru_2/p \rfloor, p)} \frac{1}{i_2} \sum_{i_1=0}^{\min(\lfloor Ru_1/p \rfloor, p)} \mathbb{1}_{[0,\sqrt{i_2/p}]}\left(\frac{i_1}{p}\right) \qquad \text{(by equation (5.12) in lemma 5.10, twice)} \\
&= \frac{1}{2p^3} \sum_{i_2=0}^{\min(\lfloor Ru_2/p \rfloor, p)} \min\left(\sqrt{\frac{p}{i_2}}, \frac{\min(Ru_1/p, p)}{i_2}\right) \qquad \text{(by equation (5.11) in lemma 5.10, twice).}
\end{aligned}
\tag{5.14}
$$

The figure 5.7 illustrates that the general formula equation (5.14) takes five different expressions depending on the regions where (u_1, u_2) live.

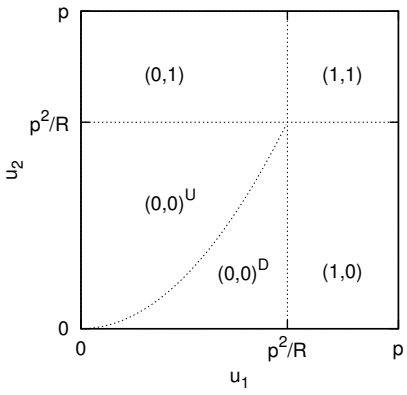


Figure 5.7: Study of values $p(U_1 = u_1, U_2 = u_2 \mid \mathcal{G}_i = F)$ for the case SMA

They are detailed below:

– $(0,0)^D$: when $u_1 \leq p^2/R$, $u_2 \leq p^2/R$, and $u_2 \leq \frac{R}{p^2}u_1^2$:

$$f(U_1 = u_1, U_2 = u_2) = \frac{\sqrt{Ru_2}}{p^3}.$$

– $(0,0)^U$: when $u_1 \leq p^2/R$, $u_2 \leq p^2/R$, and $u_2 > \frac{R}{p^2}u_1^2$:

$$f(U_1 = u_1, U_2 = u_2) = \frac{u_1 R}{p^4} + \frac{u_1 R}{2p^4}\ln\frac{p^2 u_2}{u_1^2 R}.$$

– $(1,0)$: when $p^2/R \leq u_1 < p$ and $u_2 \leq p^2/R$:

$$f(U_1 = u_1, U_2 = u_2) = \frac{\sqrt{Ru_2}}{p^3} \qquad \begin{array}{l}\text{(same expression as in}\\ \text{neighboring region } (0,0)^D).\end{array}$$

– $(0,1)$: when $u_1 \leq p^2/R$ and $p^2/R \leq u_2 < p$:

$$f(U_1 = u_1, U_2 = u_2) = \frac{u_1 R}{p}\left(1 - \frac{u_1 R}{p^2}\right).$$

– $(1,1)$: when $p^2/R \leq u_1 < p$ and $p^2/R \leq u_2 < p$:

$$f(U_1 = u_1, U_2 = u_2) = \frac{1}{p^2}.$$

We have that, when $\mathcal{G}_i = F$ in SMA,

$$\begin{aligned}\mathbb{P}(X_{M_i} = 0, X_{S_{i-1}} = 0) &= \iint_{[0,p]^2} f(U_1 = u_1, U_2 = u_2)\,\mathrm{d}u_1\,\mathrm{d}u_2 \\ &= 1 - \frac{7}{12}\frac{p}{R} + \frac{1}{8}\left(\frac{p}{R}\right)^2.\end{aligned}$$

Again, partial probabilities given in proposition 3.5 allow to derive the three other probabilities of the table of table 5.4 corresponding to $\mathcal{G}_i = F$ in SMA.

Eventually, when $\mathcal{G}_i = F$ in ML, we have independent multiplication and square, hence the factorization $\mathbb{P}(X_{M_i} = 0, X_{S_{i-1}} = 1) = \mathbb{P}(X_{M_i} = 0)\mathbb{P}(X_{S_{i-1}} = 1)$. □

5.3 Exploiting the bias using our attack

In the previous section 5.2, we notice that for random numbers,

– the presence / absence of an extra-reduction of a multiplication, and

– the presence / absence of an extra-reduction of a square that follows

are highly correlated. This allows to test whether some data produced by an operation is fed into the next operation. We can differentiate whether an operation uses as input the data from the output of another operation by computing $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G} = T)$ or by computing $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G} = F)$ (section 5.3.1). The bit value k_i can be estimated using the Pearson correlation ρ as a distinguisher, a threshold $\mathcal{T}$ depending of the knowledge of the attacker (section 5.3.2) and a decision function denoted by $\mathcal{F}_{\mathcal{ALG}}$ (section 5.3.3) which depends of the regular exponentiation algorithm and the used distinguisher. We detail first the attack using the Pearson correlation as distinguisher, but in the second chapter 6, we explain the attack with Maximum likelihood distinguisher to compare the attack in various adversarial contexts.

5.3.1 Attacker's method

An attacker calls Q times the cryptographic operation with a static key k and measures the corresponding side channel trace. For each trace $q \in \{1, \ldots, Q\}$, $(l-1)$ pairs of extra-reductions $(x^q_{M_i}, x^q_{S_{i-1}})_{l-1 \geq i > 0}$ are captured. The complete acquisition campaign is denoted $(\underline{x_{M_i}}, \underline{x_{S_{i-1}}})$, and is a matrix of size $Q \times (l-1)$ pairs of bits. Notice that neither the input nor the output of the cryptographic algorithm is required. For all $i \in \{l-1, \ldots, 1\}$ and $q \in \{1, \ldots, Q\}, x^q_{M_i}$ is equal to 1 (resp. 0) if the eXtra-reduction is present (resp. missing) during the multiplication M_i for query q. Similarly, $x^q_{S_{i-1}}$ is equal to 1 (resp. 0) if the eXtra-reduction is present (resp. missing) during the square S_{i-1} for query q. For each pair of random variable $(x_{M_i}, x_{S_{i-1}}) \in \{0,1\}^2$, the attacker first computes the estimated probability $\hat{\mathbb{P}}(X_{M_i} = x_{M_i}, X_{S_{i-1}} = x_{S_{i-1}})$, using:

$$\hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}}) = \frac{1}{Q} \sum_{q=1}^{Q} \mathbb{1}_{(x^q_{M_i} = x_{M_i}) \wedge (x^q_{S_{i-1}} = x_{S_{i-1}})}. \tag{5.15}$$

The attacker then computes the Pearson correlation[3] $\hat{\rho}(X_{M_i}, X_{S_{i-1}})$ using the estimated probability $\hat{\mathbb{P}}(X_{M_i}, X_{S_{i-1}})$. Figure 5.8 permits to illustrate the attack's methods with three acquisitions. Finally, the attacker estimates the exponent bit k_i with his knowledge corresponding to threshold $\mathcal{T}$ and decision function $\mathcal{F}_{\mathcal{ALG}}$.

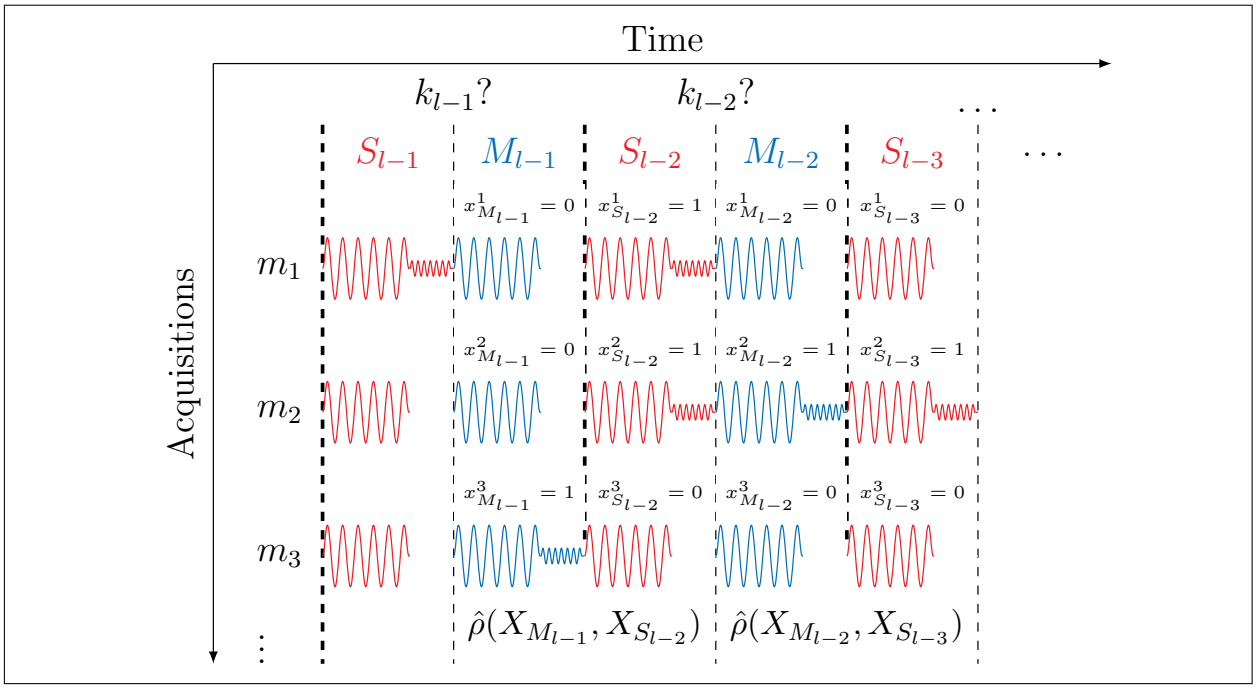


Figure 5.8: Method to compute the estimated Pearson correlation for two first k_i values

[3] $\hat{\rho}(X_{M_i}, X_{S_{i-1}}) = \frac{\hat{\mathbb{C}ov}(X_{M_i}, X_{S_{i-1}})}{\hat{\sigma}_{X_{M_i}} \hat{\sigma}_{X_{S_{i-1}}}} = \frac{\hat{\mathbb{P}}(X_{M_i}=1, X_{S_{i-1}}=1) - (\hat{\mathbb{P}}(X_{M_i}=1) \times \hat{\mathbb{P}}(X_{S_{i-1}}=1))}{\sqrt{\hat{\mathbb{P}}(X_{M_i}=1)(1-\hat{\mathbb{P}}(X_{M_i}=1))}\sqrt{\hat{\mathbb{P}}(X_{S_{i-1}}=1)(1-\hat{\mathbb{P}}(X_{S_{i-1}}=1))}}$.

5.3.2 Attacker's knowledge

In public key cryptography, the attacker wants to recover the private exponent in RSA or the private scalar in ECC. In our attacks, we assume these secret values are static, as for instance in RSA-CRT decryption or static Diffie-Hellman key agreement protocol.

– In RSA-SFM[4] and ECC, the attacker knows the parameters p and R defined in section 2.4.3. In RSA-SFM, p is equal to the public modulus n_{RSA}. In ECC, p equals the characteristic of the finite field over which the elliptic curve is defined. The attacker can compute the Pearson correlations $\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T)$ and $\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$ using corollary (5.5). The threshold for the successful attack is defined by:

$$\mathcal{T} = \frac{\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T) + \rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F)}{2}. \tag{5.16}$$

– In RSA-CRT, the attacker does not know the parameters p and R defined in section 2.4.3, because the prime factors p_{RSA} and q_{RSA} are secret parameters. Hence the determination of the probabilities by theory or simulation are impossible. To be exact, as underlined by Werner Schindler in [Sch02, Sec. 10, page 277], the ratio p/R can be estimated using the empirical probability for an extra reduction in a squaring, which is equal to $p/3R$ (recall proposition 3.5). However, using the Q measurements $(\underline{x_{M_i}}, \underline{x_{S_{i-1}}})$, the attacker is able to determine the mean estimated probability $\hat{\mathbb{E}}_i\hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}})$ for all $(x_{M_i}, x_{S_{i-1}}) \in \{0,1\}^2$ by[5]:

$$\hat{\mathbb{E}}_i\hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}}) = \frac{\sum_{i=1}^{l-1} \hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}})}{l-1}. \tag{5.17}$$

The attacker then computes the mean estimated Pearson correlations using the mean estimated probability (equation (5.17)), and the threshold for the successful attack is defined by:

$$\mathcal{T} = \frac{\hat{\mathbb{E}}_i\hat{\mathbb{P}}(X_{M_i} = 1, X_{S_{i-1}} = 1) - \hat{\mathbb{E}}_i\hat{\mathbb{P}}(X_{M_i} = 1) \times \hat{\mathbb{E}}_i\hat{\mathbb{P}}(X_{S_{i-1}} = 1)}{\sqrt{\hat{\mathbb{E}}_i\hat{\mathbb{P}}(X_{M_i} = 1)\hat{\mathbb{E}}_i\hat{\mathbb{P}}(X_{M_i} = 0)}\sqrt{\hat{\mathbb{E}}_i\hat{\mathbb{P}}(X_{S_{i-1}} = 1)\hat{\mathbb{E}}_i\hat{\mathbb{P}}(X_{S_{i-1}} = 0)}}. \tag{5.18}$$

In fact, the threshold value $\mathcal{T}$ computed in equation (5.16) or equation (5.18) does not depend on i. The indication of index i was kept as a reminder that the multiplication M_i is done in the iteration which precedes that of the square S_{i-1}.

5.3.3 Decision function

The decision function depending of the regular algorithm and the used distinguisher ρ is denoted as $\mathcal{F}_{\mathcal{ALG}}$. We detail this function for the SMA (algorithm 3.1) and ML (algorithm 3.2) algorithms.

[4]RSA-StraightForward Method; it is RSA without CRT.

[5]Notice that in some cases, e.g. if the key bits happen not to be balanced, $\hat{\mathbb{E}}_i\hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}})$ can be estimated in a less biased way using $\max_{i=1}^{l-1}\{\hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}})\} - \min_{i=1}^{l-1}\{\hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}})\}$.

SMA algorithm. In the SMA algorithm, the scalar bit k_i decides whether the output of M_i is the input of S_{i-1} or not (see figure 5.2). If the bit value k_i equals 1, then the square S_{i-1} depends on M_i ($\mathcal{G}_i = T$), otherwise the output value of M_i is different from the input value of S_{i-1} ($\mathcal{G}_i = F$). Using the section 5.2, we see that $\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T) < \rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$, so the decision function $\mathcal{F}_{SMA}$ is defined by:

$$\hat{k}_i = \mathcal{F}_{SMA}(\rho, \mathcal{T}) = \begin{cases} 0 & \text{if } \hat{\rho}(X_{M_i}, X_{S_{i-1}}) \geq \mathcal{T}, \\ 1 & \text{otherwise.} \end{cases} \tag{5.19}$$

ML algorithm. For the Montgomery Ladder (ML) algorithm, the M_i and S_{i-1} operations do not depend directly on the key bit value k_i. The dependence comes from the bit value k_{i-1} and the previous bit value k_i. If the two bits value k_{i-1} and k_i are different then the output of multiplication M_i and the input of square S_{i-1} are equal ($\mathcal{G}_i = T$), otherwise these output/input are different ($\mathcal{G}_i = F$). An illustration is provided in figure 5.3. Using section 5.2, we see that $\rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T) < \rho(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$, so the decision function $\mathcal{F}_{ML}$ using the previously estimated bit $\hat{k}_{i-1}$ is defined for each i ($l - 1 > i \geq 1$) by:

$$\hat{k}_i = \mathcal{F}_{ML}(\hat{k}_{i-1}, \rho, \mathcal{T}) = \begin{cases} \hat{k}_{i-1} & \text{if } \hat{\rho}(X_{M_i}, X_{S_{i-1}}) \geq \mathcal{T}, \\ \neg\hat{k}_{i-1} & \text{otherwise.} \end{cases} \tag{5.20}$$

Each attack step considers the difference between two consecutive key bits $\theta_i = k_i \oplus k_{i-1}$. Knowing the first key bit value, each bit k_{i-1} can be retrieve using the θ_i and the previous bit k_i.

Regarding the second most significant bit k_{l-1} of the exponent, either both values $k_{l-1} = 0$ and $k_{l-1} = 1$ are tested to recover the full secret key, or our attack can be applied between the first square FS (defined at line 2 of algorithm 3.2) and the square S_{l-1} (line 5 of algorithm 3.2) (see figure 5.9).

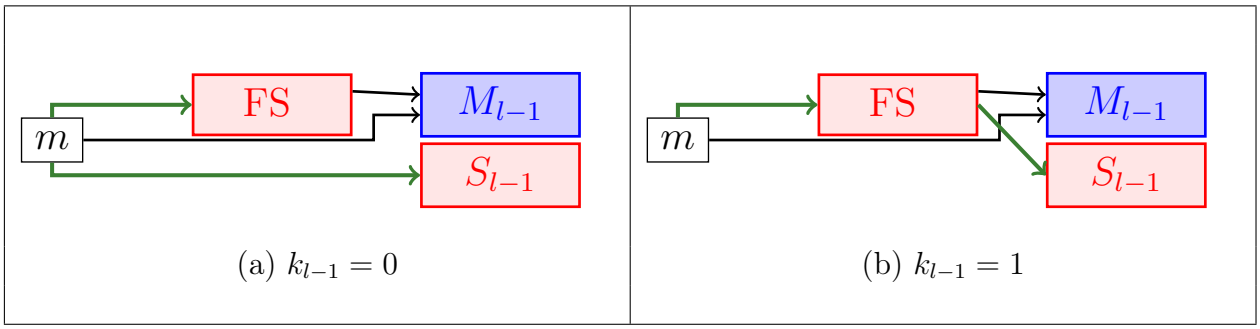


Figure 5.9: First iteration of the Montgomery Ladder algorithm.

Lemma 5.11 *The following joint probabilities of extra-reduction in first square equals or not the second square in Montgomery Ladder algorithm depends on the value of* k_{l-1}.

	$\mathbb{P}(X_{FS} = X_{S_{l-1}}\|k_{l-1})$	$\mathbb{P}(X_{FS} \neq X_{S_{l-1}}\|k_{l-1})$
$k_{l-1} = 0$	1	0
$k_{l-1} = 1$	$1 - \frac{2}{3}\frac{p}{R} + \frac{2}{21}\left(\frac{p}{R}\right)^4$	$\frac{2}{3}\frac{p}{R} - \frac{2}{21}\left(\frac{p}{R}\right)^4$

Proof

– For $k_{l-1} = 0$, the squaring FS and S_{l-1} have the same input (like showed by the green paths in figure 5.9(a)), so the probabilities of the extra-reduction is the same, that implies the first line of the table.

– For $k_{l-1} = 1$, the same input of the squaring S_{l-1} equals the output of the squaring FS (like showed by the green paths in figure 5.9(b)). The probabilities $\mathbb{P}(X_{FS} = X_{S_{l-1}}|k_{l-1} = 1)$ using the joint probabilities $\mathbb{P}(X_{FS}, X_{S_{l-1}})$.

$$\begin{aligned}\mathbb{P}(X_{FS} = 1, X_{S_{l-1}} = 1) &= \int_0^1 \int_0^{m^2 \frac{p}{R}} \int_0^{s_{l,1}^2 \frac{p}{R}} 1 ds_{l-1,1} ds_{l,1} dm \\ &= \frac{1}{3 \times 7} \left(\frac{p}{R}\right)^4 = \frac{1}{21} \left(\frac{p}{R}\right)^4 .\end{aligned} \tag{5.21}$$

$$\begin{aligned}\mathbb{P}(X_{FS} = 0, X_{S_{l-1}} = 0) &= \int_0^1 \int_{m^2 \frac{p}{R}}^1 \int_{s_{l,1}^2 \frac{p}{R}}^1 1 ds_{l-1,1} ds_{l,1} dm \\ &= 1 - \frac{2}{3}\frac{p}{R} + \frac{1}{21} \left(\frac{p}{R}\right)^4 .\end{aligned} \tag{5.22}$$

We have $\mathbb{P}(X_{FS} = X_{S_{l-1}}|k_{l-1} = 1) = \mathbb{P}(X_{FS} = 1, X_{S_{l-1}} = 1) + \mathbb{P}(X_{FS} = 0, X_{S_{l-1}} = 0)$, that implies the second line of the table.

□

In fact, the First Square (FS) corresponds to the square of R_0, so if $k_{l-1} = 0$ then the square S_{l-1} is the square of $R_{k_{l-1}} = R_0 = m$ that is equal to FS, so the both eXtra-reductions are the same. Using the estimated joint probabilities of X_{FS} and $X_{S_{l-1}}$, we can find the first estimated bit $\hat{k}_{l-1}$. A possible distinguisher for $(X_{FS}, X_{S_{l-1}})$ can be $\mathcal{D}(X_{FS}, X_{S_{l-1}}) = \hat{\mathbb{P}}(X_{FS} = X_{S_{l-1}})$ and a threshold $\mathcal{T} = (\mathbb{P}(X_{FS} = X_{S_{l-1}}|k_{l-1} = 0) + \mathbb{P}(X_{FS} = X_{S_{l-1}}|k_{l-1} = 1))/2 = 1 - \frac{2}{6}\frac{p}{R} + \frac{2}{42}\left(\frac{p}{R}\right)^4$.

$$\hat{k}_{l-1} = \mathcal{F}_{ML}(\mathcal{D}, \mathcal{T}) = \begin{cases} 0 & \text{if } \mathcal{D}(X_{FS}, X_{S_{l-1}}) < \mathcal{T} \\ 1 & \text{otherwise.} \end{cases} \tag{5.23}$$

With this method, we find the first bit k_{l-1} and continue the attack using the decision function describe by equation (5.20) to find the other bits.

5.3.4 Summary of the attack.

To estimate the exponent k by $\hat{k}$, we define two attacks:

– The attack named "ρ-attack-Hard", knowing the values of $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = T)$ and $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$, using the threshold $\mathcal{T}$ computed by equation (5.16).

– The attack named "ρ-attack-Soft", when the theoretical value $\mathbb{P}(X_{M_i}, X_{S_{i-1}}|\mathcal{G}_i)$ is unknown. It uses the estimated probability $\hat{\mathbb{P}}(X_{M_i}, X_{S_{i-1}})$ to compute the threshold $\mathcal{T}$ by equation (5.18).

Algorithm 5.1: ρ-attack using histogram method for probability estimation

Require: $(\underline{x_{M_i}}, \underline{x_{S_{i-1}}})$, a set of Q pairs of $(l-1)$ bits

Ensure: An estimation $\hat{k} \in \{0,1\}^{l-1}$ of the secret exponent

1: **for** $i = l-1$ **down to** 1 **do**
2: **for** $(x_{M_i}, x_{S_{i-1}}) \in \{0,1\}^2$ **do**
3: $\hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}}) \leftarrow 0$
4: **end for**
5: **for** $q = 1$ **to** Q **do**
6: $\hat{\mathbb{P}}(X_{M_i} = x^q_{M_i}, X_{S_{i-1}} = x^q_{S_{i-1}}) \leftarrow \hat{\mathbb{P}}(X_{M_i} = x^q_{M_i}, X_{S_{i-1}} = x^q_{S_{i-1}}) + 1$
7: **end for**
8: **for** $(x_{M_i}, x_{S_{i-1}}) \in \{0,1\}^2$ **do**
9: $\hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}}) \leftarrow \hat{\mathbb{P}}(x_{M_i}, x_{S_{i-1}}) \ / \ Q$ ▷ Normalization
10: **end for**
11: Compute $\hat{\rho}(X_{M_i}, X_{S_{i-1}})$ using $\hat{\mathbb{P}}(X_{M_i}, X_{S_{i-1}})$
12: **end for**
13: Compute $\mathcal{T}$ depending on the attacker's knowledge
14: **for** $i = l-1$ **down to** 1 **do**
15: $\hat{k}_i \leftarrow \mathcal{F}_{\mathcal{ALG}}\left(\hat{\rho}(X_{M_i}, X_{S_{i-1}}), \mathcal{T}\right)$ ▷ Threshold
16: **end for**
17: **return** $\hat{k}$

Algorithm 5.1 describes the attack to recover a full key.

Lines 1-12 correspond to the computation of the estimated probabilities for each bit k_i defined by equation (5.15). Line 13 is the computation of the threshold: if the attack is ρ-attack-Hard the attacker uses (5.16), otherwise the attack is ρ-attack-Soft and she uses equation (5.18). The lines 14-16 compute the full estimated key using the decision function $\mathcal{F}_{\mathcal{ALG}}$, defined by the equations (5.19) or (5.20).

5.4 Experimental results

In the first part of this section, we detail a simulated attack which exploits the bias (explained in corollary (5.5)) to determine the number of queries necessary for the success of the attack. Then, we detail the side channel part (*local timing analysis* using power consumption and electromagnetic analysis to distinguish *functional vs dummy subtractions*) in order to detect whether an eXtra-reduction is performed ($X = 1$) or not ($X = 0$) during the Montgomery reduction (algorithm 2.8).

5.4.1 Simulations

Let `RSA-1024-p` defined at section 3.5.3 the modulus p used in the SMA algorithm (algorithm 3.1). We generated one thousand random queries and saved for all MMM the information whether an extra-reduction is done or not. The length of static key k is 512 bits. As detailed in the ρ-attack (algorithm 5.1) we computed the estimated probabilities $\hat{\mathbb{P}}(X_{M_i}, X_{S_{i-1}})$ and the estimated Pearson correlation $\hat{\rho}(X_{M_i}, X_{S_{i-1}})$ to retrieve

each k_i. The estimated threshold $\mathcal{T}$ computed by equation (5.18) in our simulation is equal to -0.06076, which is an excellent approximation of the theoretical threshold equation (5.16). To retrieve each bit if the exponent, we used the decision function $\mathcal{F}_{SMA}$ described for ρ-attack in SMA by equation (5.19).

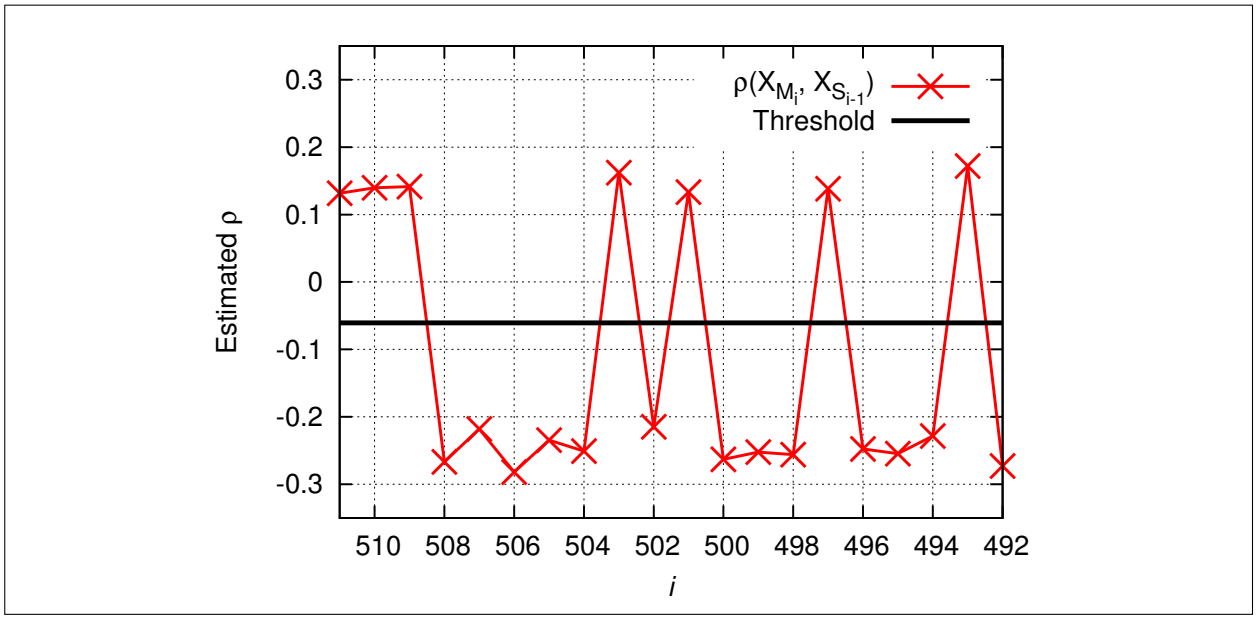


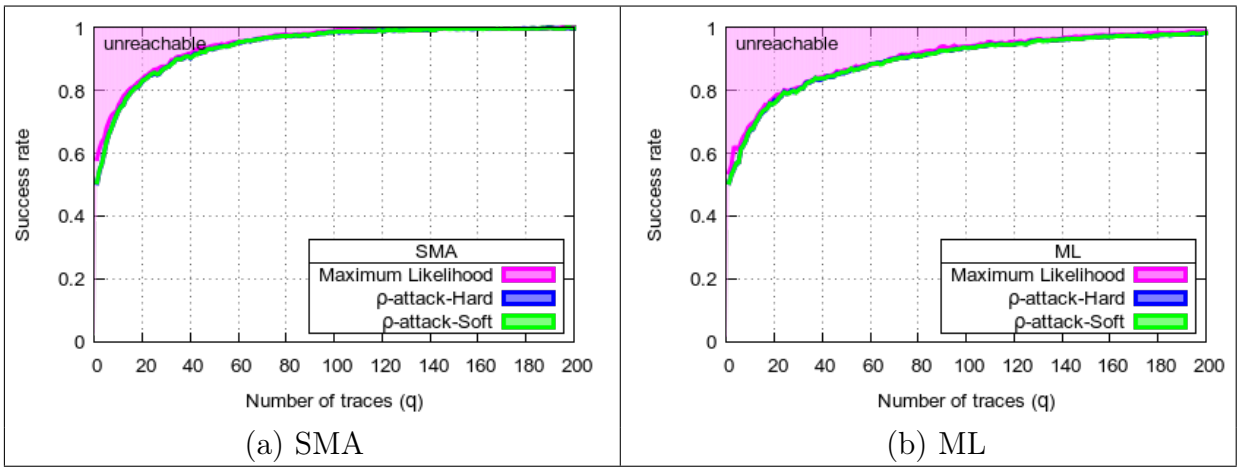
Figure 5.10: Estimated Pearson correlations using 1000 randoms queries for `RSA-1024-p` for the first 20 iterations.

Figure 5.10 shows the estimated Pearson correlation values $\hat{\rho}(X_{M_i}, X_{S_{i-1}})$ for the first iterations. It can be easily seen that the estimated key value by this sequence corresponds to `0x1000111110101110111010`011...= `0x11f5dd3`... Our ρ-attack retrieves the 511 bits of the exponent using 1000 randoms queries with success rate 100%.

Success Rate Curves. We implemented ρ-attack-Hard and ρ-attack-Soft in the ideal case, i.e., without noise. The success rate to recover one bit of the exponent is represented in figure 5.11, for both SMA and ML cases. Interestingly, ρ-attack-Hard and ρ-attack-Soft yield the same success rate, which happens to be (very close to) the optimal value. This optimal value is that obtained with the maximum likelihood distinguisher discussed in chapter 6.

The reason for the hard and soft attacks to have similar success probability is that the online estimation of the threshold is very good. Indeed, in the example of figure 5.11, the threshold $\mathcal{T}$ (equation (5.18)) is estimated based on $512Q$ traces, which is huge (one needs only to estimate 4 probabilities to get the estimation of $\mathcal{T}$). So, in the rest of this section, we make no difference between the hard and soft versions of the attacks from a success rate point of view.

The ρ-attacks are very close to the Maximum Likelihood attack for a similar reason. Estimating the difference between two random variables of very little dimensionality (recall that $(X_{M_i}, X_{S_{i-1}})$ lives in $\{0,1\}^2$) can be done almost equivalently in the proportional scale [WOS14] (Pearson correlation) as in the context of information theoretic attacks (maximum likelihood attack-chapter 6).

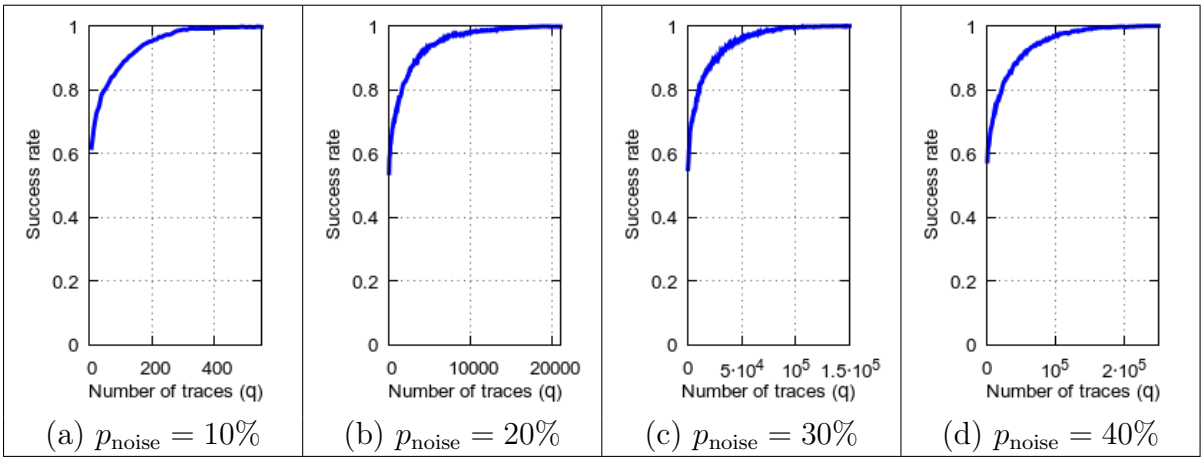


(a) SMA (b) ML

Figure 5.11: Evolution of the success rate for the ρ-attack-Soft and the ρ-attack-Hard as a function of the number Q of queries (upper bound is the maximum likelihood), for `RSA-1024-p`.

We may also notice that as the *distinguisher margin* [WO11] is larger for SMA than for ML (recall figure 5.5), the former attack requires less traces to reach a given success rate.

In practical cases, detecting an extra-reduction using only one acquisition can lead to errors. The probability to have an error is denoted by p_{noise}. We show in figure 5.12 that the attack continues to be successful (albeit with more traces) over a large range of p_{noise} values. Evidently when $p_{\text{noise}} = 50\%$ the attack becomes infeasible.



(a) $p_{\text{noise}} = 10\%$ (b) $p_{\text{noise}} = 20\%$ (c) $p_{\text{noise}} = 30\%$ (d) $p_{\text{noise}} = 40\%$

Figure 5.12: Evolution of the success rate for the ρ-attack in function of queries Q using $p =$ `RSA-1024-p` for four increasing noise values.

5.4.2 Experimental detection of extra-reductions

Two Montgomery reduction implementations will be analyzed in this section. We raise the following questions.

1. How to exploit the *local timing* to distinguish the eXtra-reduction using power consumption measurements, on OpenSSL v1.0.1k-3 ?
2. How to exploit the difference between a *real* and a *dummy* final subtraction using electromagnetic (EM) emanations, on mbedTLS v 2.2.0 ?

1a) Experiment setup in power. The target is a dual core LPC43S37 micro-controller fabricated in CMOS 90 nm Ultra Low Leakage process soldered on an LPCXpresso4337 board, and running at its maximum frequency (208 MHz). The side channel traces were obtained measuring the instantaneous power consumption with a PICOSCOPE 6402C featuring 256 MB of memory, 500 MHz bandwidth and 5 GS/s sampling rate. We executed the private function of RSA in OpenSSL with the private primes parameters defined by `RSA-1024-p` and `RSA-1024-q` defined in section 3.5.3.The private modular exponentiation is RSA-CRT with a regular algorithm.

1b) OpenSSL Experiment. The extra-reduction is explicit in the source code of OpenSSL, as shown in listing 5.1.

Listing 5.1: Extra-reduction in OpenSSL code. File `crypto/bn/bn_mont.c`, function `BN_from_montgomery`

```
309 if (BN_ucmp(ret, &(mont->N)) >= 0)
310 {
311         if (!BN_usub(ret,ret,&(mont->N))) goto err;
312 }
```

A simple power analysis using the delay (referred to as "`SPA-Timing`") between two MMM operations found whether the extra-reduction is present ($X = 1$) or not ($X = 0$). On the Cortex-M4 core, the delay between the M_i and S_{i-1} when $X_{M_i} = 1$ is 41.4952 μs, whereas the delay when $X_{M_i} = 0$ is 41.1875 μs. For the square operation S_{i-1}, the delay is 41.5637 μs when $X_{S_{i-1}} = 1$ and it is 41.2471 μs when $X_{S_{i-1}} = 0$. All in one, the observable timing differences are respectively 308 ns and 317 ns. When OpenSSL is offloaded on the Cortex-M0 core of the LPC43S37, the timing difference is respectively 399 ns and 411 ns. The success rate of this detection attack is 100%, hence $p_{\text{noise}} = 0$.

To identify this delay, the method could be the following. The presence of an extra-reduction can be identified, even when multiplications are not executed in constant time (due to instruction prefetch enabled on Cortex-M4 core). Figure 5.13 depicts the spectrogram of a power trace acquired on a Cortex-M4 running at 208MHz, with Nfft=65536, and Cauchy windowing of size 40 with 1 sample overlapping.

2a) Experiment setup in EM. The target device is an STM32F4 micro-controller, which contains an ARM Cortex-M4 processor running at its maximum frequency (168 MHz). For the acquisition, we used a Tektronix oscilloscope and a Langer near field probe. The sampling frequency is 1 GSa/s with 50 MHz hardware input low-pass filter enabled. The position of the probe was determined to maximize the signal related to the activity of the hardware 32×32 processor. We executed the private function of RSA in mbedTLS, with the private primes parameters defined by `RSA-1024-p` and `RSA-1024-q` in section 3.5.3.The private modular exponentiation is RSA-CRT with a regular algorithm.

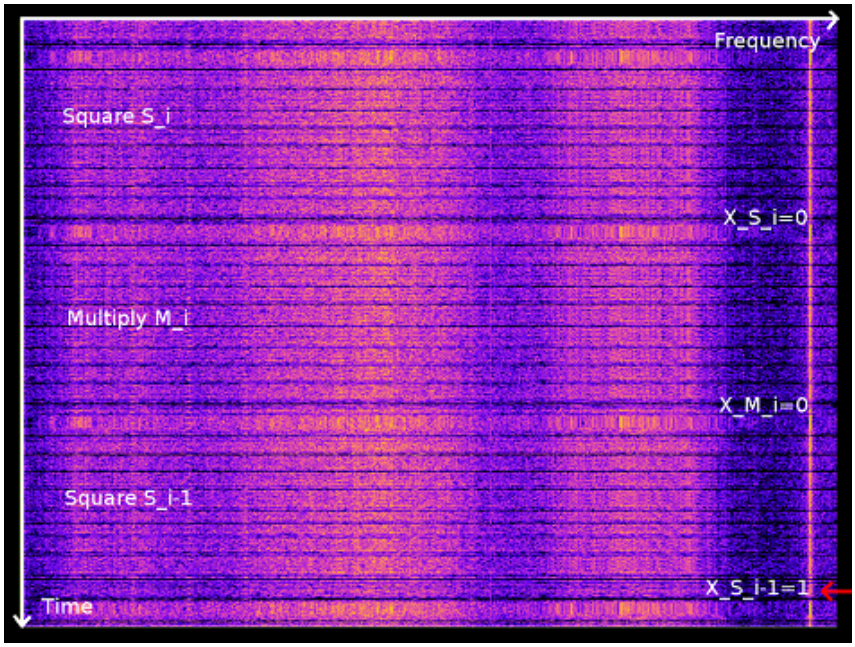


Figure 5.13: OpenSSL RSA-CRT-1024-P Spectrogram on Cortex-M4 core

2b) mbedTLS experiment. The big number library of mbedTLS implements a protection against timing attacks. A subtraction is also carried out: it is either functional or dummy, as shown in listing 5.2.

Listing 5.2: Extra-reduction in mbedTLS code. File `library/bignum.c`, function `mpi_montmul`

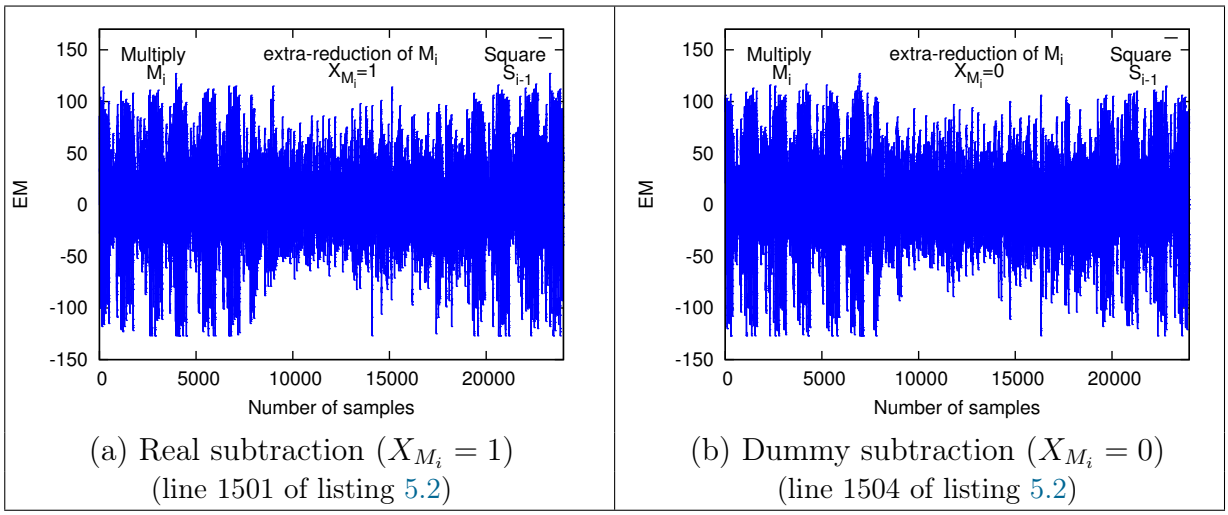
```
1500 if( mpi_cmp_abs( A, N ) >= 0 )
1501         mpi_sub_hlp( n, N->p, A->p );
1502 else
1503         /* prevent timing attacks */
1504         mpi_sub_hlp( n, A->p, T->p );
```

In order to achieve constant-time MMM, mbedTLS library implements a countermeasure using a dummy subtraction. In order to test the efficiency of the countermeasure, the duration of the real and dummy subtraction were compared as shown in figure 5.14. The time durations are the same. Therefore, the `SPA-Timing` attack is not practical anymore.

In a view to differentiate the two patterns, we use a horizontal side channel analysis [BJP+15], namely Pearson correlation (`max-corr`) [BBB+13] or the sum of the absolute differences (`min-abs-diff`). We build two reference patterns of the real subtraction $RP(X = 1)$ and dummy subtraction $RP(X = 0)$, and compare these patterns with one acquisition.

For this experiment, we use 500 acquisitions to build template $RP(X = 1)$ and again 500 acquisitions to make $RP(X = 0)$. The detection attack using one acquisition $\mathcal{A}_x$ where the extra-reduction $X = x$ is considered successful:

– when $\rho(\mathcal{A}_x, RP(X = x)) > \rho(\mathcal{A}_x, RP(X = \neg x))$ for `max-corr`, and



(a) Real subtraction ($X_{M_i} = 1$) (line 1501 of listing 5.2)

(b) Dummy subtraction ($X_{M_i} = 0$) (line 1504 of listing 5.2)

Figure 5.14: Electromagnetic acquisition focuses on one real subtraction (*left*) and pattern of one dummy subtraction (*right*) between two consecutive MMM operations.

– when $\mathbb{E}(|\mathcal{A}_x - RP(X = x)|) < \mathbb{E}(|\mathcal{A}_x - RP(X = x)|)$ for `min-abs-diff`.

The success rate of the extra-reduction detection using 30000 acquisitions is 82.50% for `max-corr` and 83.47% for `min-abs-diff`, hence $p_{\text{noise}} < 20\%$.

5.4.3 Conclusions on experiments

By combining the detection of extra-reductions using side channel analysis (section 5.4.2) and the theoretical attack to decide whether or not there is a dependency between various MMMs (section 5.3), we deduce the number of queries Q needed to recover the secret exponent k. Table 5.3 summaries the results.

Type of attack side channel for detection	`SPA-Timing`	`max-corr`	`min-abs-diff`
Detection probability for one query $= 1 - p_{\text{noise}}$	100%	82.50%	83.47%
Number of queries (SMA)	≈ 200	≈ 10000	≈ 10000
Number of queries (ML)	≈ 400	≈ 20000	≈ 20000

Table 5.3: Summary of the number of queries (see figure 5.12(b)) to retrieve all key bits of a secret exponent, as a function of side channel detection method and regular exponentiation algorithm.

5.5 Discussion

5.5.1 Attack using consecutive square operations

The attack works also if we consider the input/output dependencies between the eXtra-reduction of two consecutive square operations. To find the bit value k_i of the key, we compute the theoretical probabilities $\mathbb{P}(X_{S_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$ and $\mathbb{P}(X_{S_i}, X_{S_{i-1}}|\mathcal{G}_i = T)$, and make the same kind of attack like described in section 5.3. In figure 5.15, we can see when there is a dependency between the output of S_i and the input of S_{i-1} according to the bit value k_i during the modular exponentiation SMA.

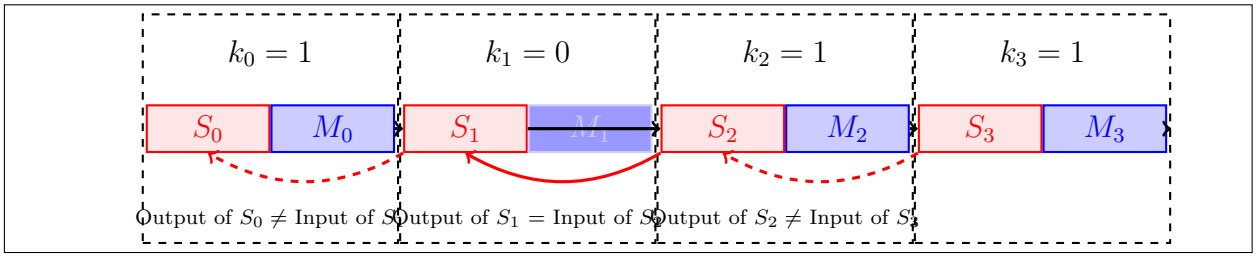


Figure 5.15: Dependency of the operation between Square i and Square $i+1$ in a Square-and-Multiply-Always algorithm

When one compares the figure 5.15 to the figure 5.2, one can clearly see that the bit estimation $\hat{k}_i$ in all the attack algorithms is the opposite of the estimation described by equation (5.19). Indeed, when the bit value is 0, the input of square S_{i+1} is equal to the output of the square S_i ($\mathcal{G}_i = T$), and when the bit value is 1, the input of square S_{i-1} is different from the output of the square S_i ($\mathcal{G}_i = F$). We have the same relation between $\rho(X_{S_i}, X_{S_{i-1}}|\mathcal{G}_i = T)$ and $\rho(X_{S_i}, X_{S_{i-1}}|\mathcal{G}_i = F)$. So, the new decision function $\mathcal{F}_{SMA}$ is defined by:

$$\hat{k}_i = \mathcal{F}_{SMA}(\rho, \mathcal{T}) = \begin{cases} 0 & \text{if } \rho(X_{S_i}, X_{S_{i-1}}) \leq \mathcal{T}, \\ 1 & \text{otherwise.} \end{cases} \tag{5.24}$$

The main advantage of using only the squares is that the time to retrieve the eXtra-reduction is divided by two. When the length of the exponent is 1024 bits, we need to detect only 1024 eXtra-reductions during the square operations. Note that the best strategy is to use both methods to increase the success rate, and reduce the number of required queries.

Remark For the ML algorithm, we can choose the two consecutive squares S_i and S_{i-1}. If the two bits value k_i and k_{i-1} are equal, then the Output/Input of two squares are equals ($\mathcal{G}_i = T$), else these operations are independent ($\mathcal{G}_i = F$). Moreover, the side channel information garnered from extra-reductions in both (M_i, S_{i-1}) and (S_i, S_{i-1}) can be advantageously combined to speed up the key recovery.

5.5.2 Other exponentiation implementations

This work limits the attack to binary regular algorithms, but some other algorithms can also be attacked by exploiting correlations between eXtra-reductions.

Multiply-always. The Multiply-Always algorithm applies when the square implementation is the same as the multiply operation. In ECC, it is equivalent to the Add-Always algorithm, when the doubling and adding operations are unified. These kinds of algorithms can be attacked by the classical Square-and-Multiply proposed by Schindler in [Sch00] or his improvements [SKQ01, ASK05, AS08] using the result of proposition 3.5.

Window algorithm. The window exponentiation algorithm is used to provide efficient computation. The timing attacks proposed by Schindler in [Sch02] allows to find the exponent during the window algorithm, also with the help of extra-reductions.

5.5.3 Efficient countermeasures

To avoid our attack, one of the following countermeasures described here is sufficient.

Use exponent blinding. For RSA (resp. ECC), a classical countermeasure is the exponent blinding (resp. scalar blinding). This countermeasure is efficient to avoid the attack, because to compute the estimated probabilities for the Q queries, the exponent bit must be fixed. Although Berzati and al. in [BCG10] show that the exponent blinding is partially ineffective on some bits depending on the chosen modulo, the bias seems not easily exploitable.

Use another Montgomery reduction. In [ÖBPV08, algorithm 2], Örs and al. describe an MMM algorithm without final subtraction. Thus, there is no eXtra-reduction to exploit in our attack.

5.5.4 ECC particular case

The new attacks have been detailed on RSA but are also applicable to ECC with appropriate customization for various ECC implementations. In ECC, each iteration on the regular algorithm is composed by one Elliptic Curve Add operation and one Elliptic Curve Double operation; however, the Elliptic Curve operation is itself composed of finite field operation (square, multiply, addition, etc.). In order to exploit this bias in ECC, the consecutive operations that an attacker can select are the last operation (multiply or square) used to compute a value of one point coordinate in one iteration and the first operation (multiply or square) using this coordinate value in the next iteration.

As an example [BL] for addition `madd-2004-hmv`, the Z-coordinate in output of addition is computed by a multiplication $Z3 = Z1 \times T1$ and for doubling `dbl-2007-bl`, the Z-coordinate in input of doubling is a square $ZZ = Z1 \times Z1$. This example is retrieved in the PolarSSL v1.3.7. In appendix C, the doubling algorithm and addition algorithm used in PolarSSL v1.3.7 are described. The multiplication in line 14 of adding algorithm (algorithm C.2) comes before the square in line 9 of doubling algorithm (algorithm C.1).

5.6 Conclusion

This chapter has presented a new theoretical and practical attack against asymmetrical computation with regular exponentiation using extra-reductions as a side channel. The working factor is the existence of a strong bias between the extra-reductions during the Montgomery Modular Multiplication of two consecutive operations. This new bias can be exploited in each regular binary algorithm, because each iteration consists in a square and a multiply whose inputs depend on the outputs of an operation from the previous iteration. In elliptic curve cryptography, the attack can be applied when one square or a multiplication in doubling operation feeds another operation in the addition.

In this chapter, a demonstration of the leakage information provided by extra-reduction in two consecutive operations permits to break blinded regular exponentiation. The main exploited information leakage provides of the multiplication feeds in the following square, but there are information of all consecutive operation, as see in section 5.5.1. In the following chapter 6, we described the improvement of this attack. The improvement permits to use more information of extra-reduction in consecutive operations, in order to reduce the number of queries. It is important to reduce the number of queries. In fact when the probability to have an error in detection of extra-reduction increases the number of queries required to succeed of the attack grows up. The next chapter permits to reduced by 2 the number of queries required to retrieve a whole exponent.

Chapter 6

Optimization of extra-reduction analysis

This work is an improvement of the extra-reduction analysis presented in the previous chapter. It is a collaborative work with Werner Schindler and Sylvain Guilley.

In the previous chapter 5, extra-reduction analysis on RSA were investigated. The side channel information was used to identify, which Montgomery multiplications require extra reductions. Two exponentiation algorithms were considered, namely the always square and multiply exponentiation and the Montgomery ladder. The overall attacks split into many individual decisions whether the "guess" value $\mathcal{G}_i$ is `True` or `False`, if two consecutive operations are dependent on their input or their output or are totally independent. The presented attacks in chapter 5 were successful but for these decisions only two –one squaring and one multiplication– out of four Montgomery operations were exploited. However the approach of the previous chapter is too complex: the derivation of the p.d.f of values for multiple operations becomes intractable. For these reasons, we resort to another way to estimated the distribution of the extra-reduction which need not to estimate the p.d.f of the values. We leverage on previous work of Schindler [Sch02]: this paper simplifies the characterization of the extra-reduction distribution using two elegant properties of Montgomery Modular Multiplication (MMM). First of all the random variables corresponding to the result of the MMM may be viewed as independent and identically distributed (iid) and uniformly. Second the presence or absence of an extra-reduction is obtained as an indicator function with a threshold on the values (which does not require to know the values p.d.f.)

In this chapter, we optimize the attack by exploiting all available piece of information. We begin with definitions and we formulate the target of our attack. In section 6.2 we analyze the stochastic properties of the Montgomery multiplication, and in lemma 6.3 we develop a formula for the joint probability of several extra reductions. The following sections treat the estimation of two parameters, which are usually unknown, and the Maximum-Likelihood estimator is derived.

Contents

6.1 Introduction

In this chapter we only consider the "Montgomery ladder" (left-to-right), which is described in algorithm 3.2. Unlike previous chapter 5, we do not consider the "square and always multiply" algorithm (algorithm 3.1). It is obvious how the applied mathematical methods can be transferred to the "square and always multiply" exponentiation algorithm.

We assume that the basis m has been blinded (basis blinding, a.k.a. message blinding). The attack applies to both to RSA with CRT and to RSA without CRT. We further assume that the arithmetic operations apply the Montgomery's multiplication (section 2.4.3). As in chapter 5 we assume that a side channel attack yields the (possibly noisy) information, which Montgomery multiplications need an extra reductions. The applied mathematical techniques are similar to that in [Sch02, AS08] where attacks on different variants of fixed window exponentiation algorithms were analyzed thoroughly.

To avoid clumsy formulations we always target RSA with CRT in the following where p denotes one prime factor of the RSA modulus n. We note that the attack on RSA without CRT works identically and is even simpler since there is no need not estimate the ratio n/R.

The following definition 6.1 is a reminder of the notations, necessary to understand in this chapter.

Definition 6.1 *For $i = l, l-1, \ldots, 0$ and $j = 0, 1$ the term $r_{i,j}$ denotes the value of register R_j after the key bit k_i has been processed. Further, $s_{i,j} = r_{i,j}/p \in [0, 1)$ stands for the normalized register values. For $i = l-1, \ldots, 0$ we set $x_{M_i} = 1$ if the first Montgomery operation for key bit k_i ('multiplication') needs an extra reduction (ER) and $x_{M_i} = 0$ else. Analogously, $x_{S_i} = 1$ if the second Montgomery operation for key bit k_i ('squaring') needs an ER and $x_{S_i} = 0$ else. In the context of random variables the abbreviation 'iid' stands for 'independent and identically distributed'. The indicator function $\mathbb{1}_A(x)$ assumes the value 1 if $x \in A$ and 0 else. For $b \in \mathbb{Z}$ the term $b \pmod p$ denotes the unique element in $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, which is congruent to b modulo p. The letter R denotes the Montgomery constant. Finally, for $a, b \in \mathbb{Z}_p$ we define $\mathrm{MM}(a, b; p) = abR^{-1} \pmod p$ (Montgomery multiplication). For an integer u, $(\underline{x_{M_i}}, \underline{x_{S_i}}, \ldots, \underline{x_{M_{i-u}}}, \underline{x_{S_{i-u}}})$ is a set of Q u-tuples of $(l-u)$ bits. For all $i \in \{l-u, \ldots, 1\}$ and $q \in \{1, \ldots, Q\}$, $x^q_{M_i}$ is equal to 1 (resp. 0) if the eXtra-reduction is present (resp. missing) during the multiplication M_i for query q. Similarly, $x^q_{S_i}$ is equal to 1 (resp. 0) if the eXtra-reduction is present (resp. missing) during the square S_i for query q.*

6.2 The core of our attack

We interpret the $s_{i,j}$ as realizations of random variables $S_{i,j}$, i.e. values taken on by $S_{i,j}$, which assume values in $[0,1)$. Analogously, we view x_{M_i} and x_{S_i} as realizations of $\{0,1\}$-valued random variables X_{M_i} and X_{S_i}. Lemma 6.2(i) - (ii) collect known stochastic properties of Montgomery's multiplication algorithm while the assertions (iii)-(iv) follow the strategy in [Sch02, AS08].

Lemma 6.2 *[Montgomery multiplication]*
(i) The Montgomery multiplication $\mathrm{MM}(a,b;p)$ *requires an extra reduction iff*

$$\left(\frac{a}{p}\frac{b}{p}\frac{p}{R} + \frac{abp(\mathrm{mod}\ R)}{R} \geq 1\right) \quad \textit{iff} \quad \left(\frac{\mathrm{MM}(a,b,p)}{p} < \frac{a}{p}\frac{b}{p}\frac{p}{R}\right). \tag{6.1}$$

(ii) Assume that $a \in \mathbb{Z}_p$ *and that the random variable* B *is uniformly distributed on* $\mathbb{Z}_p$*. Further,* U *and* V *denote independent random variables, which are uniformly distributed on* $[0,1)$*. Then approximately*

$$\mathbb{P}\left(\frac{a}{p}\frac{B}{p}\frac{p}{R} + \frac{aBp(\mathrm{mod}\ R)}{R} \geq 1\right) = \mathbb{P}\left(\frac{a}{p}\frac{p}{R}U + V \geq 1\right) = \frac{p}{2R}\frac{a}{p}, \tag{6.2}$$

$$\mathbb{P}\left(\frac{B}{p}\frac{B}{p}\frac{p}{R} + \frac{B^2p(\mathrm{mod}\ R)}{R} \geq 1\right) = \mathbb{P}\left(\frac{p}{R}U^2 + V \geq 1\right) = \frac{p}{3R}. \tag{6.3}$$

(iii) The random variables $S_{l,0}, S_{l,1}, S_{l-1,0}, \ldots, S_{0,0}, S_{0,1}$ *may be viewed as iid uniformly distributed on* $[0,1)$*.*
(iv) For $i = l-1, \ldots, 0$ *we have:*

$$X_{M_i} = \begin{cases} \mathbb{1}_{\{S_{i,1} < S_{i+1,0}S_{i+1,1}\frac{p}{R}\}} & \textit{if } k_i = 0 \\ \mathbb{1}_{\{S_{i,0} < S_{i+1,0}S_{i+1,1}\frac{p}{R}\}} & \textit{if } k_i = 1 \end{cases}, \tag{6.4}$$

$$X_{S_i} = \begin{cases} \mathbb{1}_{\{S_{i,0} < S_{i+1,0}^2\frac{p}{R}\}} & \textit{if } k_i = 0 \\ \mathbb{1}_{\{S_{i,1} < S_{i+1,1}^2\frac{p}{R}\}} & \textit{if } k_i = 1 \end{cases}. \tag{6.5}$$

(v) For the indicator functions we obtain

$$\mathbb{1}_{\{X_{M_i}=1\}} = X_{M_i}, \ \ \mathbb{1}\{X_{M_i} = 0\} = 1 - X_{M_i}, \tag{6.6}$$

$$\mathbb{1}_{\{X_{S_i}=1\}} = X_{S_i}, \ \ \mathbb{1}\{X_{S_i} = 0\} = 1 - X_{S_i}. \tag{6.7}$$

Proof The assertions (i)-(ii) are shown in [Wer02] (see Lemma A3 and its proof). The core idea of the approximate representations equation (6.2) and equation (6.3) is that a small deviation of the random variable B (resp. of B/p) causes only a small deviation of the first summand but implies an 'uncontrolled large' deviation of the second summand over the unit interval. We note that if U and V are independent then U and $(a/R)U + V(\mathrm{mod}\ 1)$ are independent, too. Since the basis m has been basis-blinded we may assume that $s_{l,0} = r_{l,0}/p = m/p$ is a realization of a random variable $S_{l,0}$, which is uniformly distributed on the unit interval $[0,1)$. Following equation (6.3) we further assume that $S_{1,0}$ is also uniformly distributed on $[0,1)$, and that $S_{l,0}$ and $S_{l,1}$ are independent (see also Remark 6.2). Now let us assume that the random variables $S_{l,0}, S_{l,1}, S_{l-1,0}, \ldots, S_{i+1,1}$

are iid uniformly distributed on $[0,1)$. If $(k_i, k_{i-1}) = (0,0)$ we may replace (a/p), U (approximation of B/p) and V in equation (6.2) by $S_{i+1,0}$, $S_{i+1,1}$ and $V_{i,0}$, and analogously U and V in equation (6.3) by $S_{i+1,0}$ and $V_{i,1}$ where $V_{i,0}$ and $V_{i,1}$ are uniformly distributed on $[0,1)$ and independent of $S_{l,0}, \ldots, S_{i+1,1}$. Further, the assumption that $V_{i,0}$ and $V_{i,1}$ are independent seems reasonable since $S_{i+1,1}$ and $S_{i,1}$ are independent. This assumption finally implies that the random variables $S_{l,0}, \ldots, S_{i,1}$ are independent. Formula equation (6.4) follows from equation (6.1) if we replace the terms (a/p) and (B/p) by $S_{i+1,0}$ and $S_{i+1,1}$ (c.f. equation (6.2), and further $\mathrm{MM}(a,b;p)/p$ by $S_{i,1}$. Analogously, to verify equation (6.5) one replaces in equation (6.1) the terms (B/p) and $\mathrm{MM}(a,b;p)/p$ by $S_{i+1,0}$ and $S_{i,1}$, respectively. The cases $(k_i, k_{i-1}) \in \{(1,0),(0,1),(1,1)\}$ are likewise. Assertion (vi) follows immediately from the definition of indicator functions. This completes the proof of lemma 6.2.

Remark [The independence assumption]
A central assertion of lemma 6.2, which is used in lemma 6.3, is that random variables $S_{i,j}$ may be viewed iid uniformly distributed on $[0,1)$. This property has been deduced from the (approximate) stochastic representations equation (6.2) and equation (6.3). In a strict sense this claim is certainly not correct, e.g. because the normalized register values $r_{i,j}/p$ only assume values in the finite set $\mathbb{Z}_p/p \subseteq [0,1)$, and to mention just one missing number theoretical property, the $r_{1,j}$ cannot assume non-quadratic residua in $\mathbb{Z}_p$. However, this not relevant for our purposes since we are only interested in the (joint) probabilities of extra reductions. These events can be characterized by 'metric' conditions in $\mathbb{R}$ (c.f. equation (6.1),equation (6.2), equation (6.3)). (In particular, even for small intervals $I \subseteq [0,1)$ we may assume that approximately half of $\{u \in \mathbb{Z}_p \mid u/p \in I\}$ are quadratic residua in $\mathbb{Z}_p$.) It should be noted that the iid assumption on the normalized intermediate random variables of the exponentiation algorithm (here: the $S_{i,j}$) has been proven successful e.g. in [Sch00, Wer02, Sch02, ASK05, AS08], and it will turn out to be successful in the following, too.

The overall attack consists of many independent decisions. Each of these attack steps (decisions) considers all Montgomery multiplications simultaneously, which are carried out when u consecutive key bits $(k_i, \ldots, k_{i-u+1})$ are processed. Lemma 6.3 is the core of our attack. It provides the probabilities, which are needed later in lemma 6.9 (maximum likelihood decision strategy).

Lemma 6.3 *Let $u \geq 1$ and $\vec{\theta} = (\theta_1, \ldots, \theta_u) \in \{0,1\}^u$.*
(i) The term equation (6.8) quantifies the probability that the extra reduction vector $(x_{M_i}, x_{S_i}, \ldots, x_{M_{i-u+1}}, x_{S_{i-u+1}})$ occurs if $(k_i, \ldots, k_{i-u+1}) = (\theta_1, \ldots, \theta_u)$. The probabilities are expressed by integrals over $[0,1)^{2u+2}$.

$$\begin{aligned} &\mathbb{P}_{\vec{\theta}}(X_{M_i} = x_{M_i}, X_{S_i} = x_{S_i}, \ldots, X_{M_{i-u+1}} = x_{M_{i-u+1}}, X_{S_{i-u+1}} = x_{S_{i-u+1}}) \qquad (6.8) \\ &= \int_0^1 \int_0^1 \int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_{2u-1}}^{b_{2u-1}} \int_{a_{2u}}^{b_{2u}} 1 \, ds_{i-u+1,1} ds_{i-u+1,0} \ldots ds_{i,1} ds_{i,0} ds_{i+1,1} ds_{i+1,0} \, . \end{aligned}$$

Note: When the key bit k_j (for $j \in \{i, i-1, \ldots, i-u+1\}$) is processed the register value R_v ($v \in \{0,1\}$) corresponds to the integration variable $s_{j,v}$. The integration boundaries (a_{2j}, b_{2j}) and (a_{2j-1}, b_{2j-1}) correspond to the integration with regard to the variables $s_{i-j+1,1}$ and $s_{i-j+1,0}$, respectively ($j = 1, \ldots, u$).

The integration boundaries depend on the hypothesis $\vec{\theta} = (\theta_1, \ldots, \theta_u)$ *and the observed extra reduction vector* $(x_{M_i}, x_{S_i}, \ldots, x_{M_{i-u+1}}, x_{S_{i-u+1}})$. *More precisely, for* $j \in \{1, \ldots, u\}$ *we have*

If $\theta_j = 0$ *then*

$$(a_{2j}, b_{2j}) = \begin{cases} (0, s_{i-j+2,0} s_{i-j+2,1} \frac{p}{R}) & \text{if } x_{M_{i-j+1}} = 1 \\ (s_{i-j+2,0} s_{i-j+2,1} \frac{p}{R}, 1) & \text{if } x_{M_{i-j+1}} = 0 \end{cases} \tag{6.9}$$

$$(a_{2j-1}, b_{2j-1}) = \begin{cases} (0, s_{i-j+2,0}^2 \frac{p}{R}) & \text{if } x_{S_{i-j+1}} = 1 \\ (s_{i-j+2,0}^2 \frac{p}{R}, 1) & \text{if } x_{S_{i-j+1}} = 0 \end{cases} \tag{6.10}$$

If $\theta_j = 1$ *then*

$$(a_{2j}, b_{2j}) = \begin{cases} (0, s_{i-j+2,1}^2 \frac{p}{R}) & \text{if } x_{S_{i-j+1}} = 1 \\ (s_{i-j+2,1}^2 \frac{p}{R}, 1) & \text{if } x_{S_{i-j+1}} = 0 \end{cases} \quad \textit{and} \tag{6.11}$$

$$(a_{2j-1}, b_{2j-1}) = \begin{cases} (0, s_{i-j+2,0} s_{i-j+2,1} \frac{p}{R}) & \text{if } x_{M_{i-j+1}} = 1 \\ (s_{i-j+2,0} s_{i-j+2,1} \frac{p}{R}, 1) & \text{if } x_{M_{i-j+1}} = 0 \end{cases} \tag{6.12}$$

(ii) Let $\vec{1} = (1, \ldots, 1)$ *(with* u *components). For each hypothesis* $\vec{\theta} \in \{0,1\}^u$ *and each extra reduction vector* $(x_{M_i}, x_{S_i}, \ldots, x_{M_{i-t+1}}, x_{S_{i-u+1}})$ *we have*

$$\mathbb{P}_{\vec{\theta}}(X_{M_i} = x_{M_i}, \ldots, X_{S_{i-u+1}} = x_{S_{i-u+1}}) = \mathbb{P}_{\vec{1}-\vec{\theta}}(X_{M_i} = x_{M_i}, \ldots, X_{S_{i-u+1}} = x_{S_{i-u+1}}) \,. \tag{6.13}$$

Proof By lemma 6.2(iv) the random variables $X_{M_i}, X_{S_i}, \ldots, X_{M_{i-u+1}}, X_{S_{i-u+1}}$ can be expressed by indicator functions, which depend on the random variables $S_{i+1,1}, S_{i+1,0}, \ldots, S_{i-u+1,1}, S_{i-u+1,0}$. This allows to express the probability equation (6.8) by an integral over $[0,1)^{2u+2}$ of a product of indicator functions. Further, for $j < u$ the indicator functions $\mathbb{1}_{\{X_{M_{i-j+1}} = x_{M_{i-j+1}}\}}$ and $\mathbb{1}_{\{X_{S_{i-j+1}} = x_{S_{i-j+1}}\}}$ actually only depend on $s_{i+1,1}, s_{i+1,0}, \ldots, s_{i-u+2,1}, s_{i-u+2,0}$ while $\mathbb{1}_{\{X_{M_{i-u+1}} = x_{M_{i-u+1}}\}}$ and $\mathbb{1}_{\{X_{S_{i-u+1}} = x_{S_{i-u+1}}\}}$ merely depend on $s_{i-u+2,1}, s_{i-u+2,0}, s_{i-u+1,1}, s_{i-u+1,0}$. This allows to express equation (6.8) in the form

$$\int_{[0,1)^{2u}} \prod_{j=1}^{u-1} \left(\mathbb{1}_{\{X_{M_{i-j+1}} = x_{M_{i-j+1}}\}} \cdot \mathbb{1}_{\{X_{S_{i-j+1}} = x_{S_{i-j+1}}\}} \right) \quad \times \\ \left(\int_{a_{2u-1}}^{b_{2u-1}} \int_{a_{2u}}^{b_{2u}} 1 \, ds_{i-u+1,1} ds_{i-u+1,0} \right) ds_{i-u+2,1} \ldots ds_{i+2,0} \tag{6.14}$$

with suitable integration boundaries $a_{2u-1}, b_{2u-1}, a_{2u}, b_{2u}$. These integration boundaries follow immediately from lemma 6.2(iv) and (ii) with $t = i-u+1$. This verifies the formula equation (6.9) to equation (6.12) for $j = u$. The integral over $[0,1)^{2u}$ can be transformed in the same way into a sequence of one-dimensional integrals. Since the integration boundaries $a_1, b_1, \ldots, a_{2u-2}, b_{2u-2}$ depend only on the left-hand indicator functions, i.e. on the observations $x_{M_i}, x_{S_i}, \ldots, x_{M_{i-u+2}}, x_{S_{i-u+2}}$ lemma 6.3(i) can be verified by induction on u.

We first note that

$$\phi \colon [0,1)^{2u+2} \to [0,1)^{2u+2},$$
$$\phi(s_{i+1,1}, s_{i+1,0}, \ldots, s_{i-u+1,1}, s_{i-u+1,0}) = (s_{i+1,0}, s_{i+1,1}, \ldots, s_{i-u+1,1}, s_{i-u+1,0})$$

(swapping the right-hand indices from 0 to 1 and vice versa) defines a volume-preserving diffeomorphism on $[0,1)^{2u+2}$. As already pointed out above the probabilities equation (6.13) can be expressed by integrals over $[0,1)^{2u+2}$ of indicator functions

$$\prod_{j=1}^{u} \mathbb{1}_{[\vec{\theta}]\{X_{M_{i-j+1}}=x_{M_{i-j+1}}\}} \cdot \mathbb{1}_{[\vec{\theta}]\{X_{S_{i-j+1}}=x_{S_{i-j+1}}\}} \quad \text{and}$$

$$\prod_{j=1}^{u} \mathbb{1}_{[\vec{1}-\vec{\theta}]\{X_{M_{i-j+1}}=x_{M_{i-j+1}}\}} \cdot \mathbb{1}_{[\vec{1}-\vec{\theta}]\{X_{S_{i-j+1}}=x_{S_{i-j+1}}\}} \quad \text{respectively.}$$

The terms $\vec{\theta}$ and $\vec{1}-\vec{\theta}$ indicate the hypotheses. From lemma 6.2(iv) we conclude that $\mathbb{1}_{[\vec{1}-\vec{\theta}]\{X_{M_{i-j+1}}=x_{M_{i-j+1}}\}} = \mathbb{1}_{[\vec{\theta}]\{X_{M_{i-j+1}}=x_{M_{i-j+1}}\}} \circ \phi$ and $\mathbb{1}_{[\vec{1}-\vec{\theta}]\{X_{S_{i-j+1}}=x_{S_{i-j+1}}\}} = \mathbb{1}_{[\vec{\theta}]\{X_{S_{i-j+1}}=x_{S_{i-j+1}}\}} \circ \phi$ for all $j \leq u$, which completes the proof of assertion (ii).

Lemma 6.3(ii) says that the information contained in the extra reduction vectors $(x_{M_i}, \ldots, x_{S_{i-u+1}})$ does not allow to distinguish between the hypotheses $\vec{\theta}$ and $\vec{1}-\vec{\theta}$. This means that we can only determine the set $\{\vec{\theta}, \vec{1}-\vec{\theta}\}$. In particular, it would be pointless to consider the case $u=1$. For $u=2$ one can distinguish between the cases $(k_i, k_{i-1}) \in \{(0,0),(1,1)\}$ and $(k_i, k_{i-1}) \in \{(0,1),(1,0)\}$, or equivalently, between $k_i = k_{i-1}$ and $k_i \neq k_{i-1}$. For $u \geq 2$ the parameter $\vec{\theta} \in \{(\theta_1, \ldots, \theta_u), (1-\theta_1, \ldots, 1-\theta_u)\}$ corresponds to

$$(k_i \oplus k_{i-1} = \theta_1 \oplus \theta_2, \ldots, k_{i-u+2} \oplus k_{i-u+1} = \theta_{u-1} \oplus \theta_u)\,. \tag{6.15}$$

Here '$\oplus$' denotes the addition modulo 2.

Remark (i) Lemma 6.3 can be applied to all u-tuples $(k_i, \ldots, k_{i+u-1})$ for $i = l-1, \ldots, u-1$. Combining the information from all u-tuples only provides the vector $(k_{l-1} \oplus k_{l-2}, \ldots, k_1 \oplus k_0)$. This information determines the whole key $k = (k_l = 1, k_{l-1} \ldots, k_0)$ since k is odd due to $\gcd(k, \phi(p)) = 1$.
(ii) The probabilities in lemma 6.3 do not depend on the index i. By lemma 6.3(ii) it suffices to compute 2^{3u-1} probabilities of type equation (6.8). Example 6.2 illustrates the calculation of one particular probability, and the appendix chapter D contains two tables with all probabilities for $u=2$ and the four tables for $u=3$.
(iii) For $u=2$ our attack aims at pairs of consecutive key bits (k_i, k_{i-1}). This is like the previous attack in chapter 5 but the previous attack only exploits the Montgomery multiplications $(x_{S_i}, x_{M_{i-1}})$ while our attack considers $(x_{M_i}, x_{S_i}, x_{M_{i-1}}, x_{S_{i-1}})$. The probabilities, which are applied in the previous attack, are the marginal probabilities of the probabilities equation (6.8) with regard to $(x_{M_i}, x_{S_{i-1}})$. Obviously, the previous attack exploits less information than the new attack for $u=2$, and experiments confirm that for $u=2$ our new attack reduces by a factor greater than 2 the number of queries. (see figure 6.4).

Example Let $(\theta_1, \theta_2) = (0,1)$ and $(x_{M_i}, x_{S_i}, x_{M_{i-1}}, x_{S_{i-1}}) = (1,1,0,1)$. By lemma 6.3(i)

$$\mathbb{P}_\theta(X_{M_i} = 1, X_{S_i} = 1, X_{M_{i-1}} = 0, X_{S_{i-1}} = 1) \tag{6.16}$$

$$
\begin{aligned}
&= \int_0^1 \int_0^1 \int_0^{s_{i+1,0}s_{i+1,1}\frac{p}{R}} \int_0^{s_{i+1,1}^2\frac{p}{R}} \int_0^{s_{i,0}^2\frac{p}{R}} \int_{s_{i,0}s_{i,1}\frac{p}{R}}^1 1\, ds_{i-1,1}ds_{i-1,0}ds_{i,1}ds_{i,0}ds_{i+1,1}ds_{i+1,0} \\
&= \cdots = \int_0^1 \int_0^1 \int_0^{s_{i+1,0}s_{i+1,1}\frac{p}{R}} \int_0^{s_{i+1,1}^2\frac{p}{R}} \left(s_{i,0}^2\frac{p}{R} - s_{i,0}^3 s_{i,1}\left(\frac{p}{R}\right)^2\right) ds_{i,1}ds_{i,0}ds_{i+1,1}ds_{i+1,0} \\
&= \cdots = \int_0^1 \int_0^1 \left(\frac{1}{3}s_{i+1,0}^3 s_{i+1,1}^5\left(\frac{p}{R}\right)^5 - \frac{1}{8}s_{i+1,0}^4 s_{i+1,1}^8\left(\frac{p}{R}\right)^8\right) ds_{i+1,1}ds_{i+1,0} \\
&= \cdots = \frac{1}{3\cdot 4\cdot 6}\left(\frac{p}{R}\right)^5 - \frac{1}{8\cdot 5\cdot 9}\left(\frac{p}{R}\right)^8 = \frac{1}{72}\left(\frac{p}{R}\right)^5 - \frac{1}{360}\left(\frac{p}{R}\right)^8 .
\end{aligned}
$$

Corollary 6.4 *For $u = 2$ by applying the law of total probability on $\mathbb{P}_\theta$ in equation (6.8), the joint probability for ML described in theorem 5.4 is recovered.*

Remark These two approaches are independent and give the same result. Here, we are not interested in the values manipulated by the multiplication and square operations, but only with the necessary and sufficient conditions for the existence of extra-reductions, allowing an analysis of larger dimensions.

6.3 Perfect and Noisy Measurements

The attacker gets access to side channel information about each bit k_i ($l-1 \geq i > 0$) of the exponent k through the noised distribution of the pair of extra-reductions $(X_{M_i}, X_{S_{i-1}})$. The noise consists in two binary random variables $(N_{M_i}, N_{S_{i-1}})$. Additionally, the random variables N_{M_i} and $N_{S_{i-1}}$ are assumed independent and identically distributed (i.i.d.), as is usually the case of measurement noise of different operations in a side channel trace. Namely, we denote by p_{noise} the probability

$$p_{\text{noise}} = \mathbb{P}(N_{M_i} = 1) = \mathbb{P}(N_{S_{i-1}} = 1) \qquad \text{for all } i.$$

Thus, as depicted in figure 6.1, the attacker garners an i.i.d. sequence $(\underline{y_{M_i}}, \underline{y_{S_{i-1}}}) = (y_{M_i}^q, y_{S_{i-1}}^q)_{q=1,\ldots,Q}$, where for each query q and exponent index i, $y_{M_i}^q = x_{M_i}^q \oplus n_{M_i}^q$ and $y_{S_{i-1}}^q = x_{S_{i-1}}^q \oplus n_{S_{i-1}}^q$. This means that X_{M_i} and Y_{M_i} are respectively the input and the output of a binary symmetric channel (BSC) of parameter p_{noise}. Similarly, $X_{S_{i-1}}$ and $Y_{S_{i-1}}$ are also input and output of an independent identical BSC parallel to the first one.

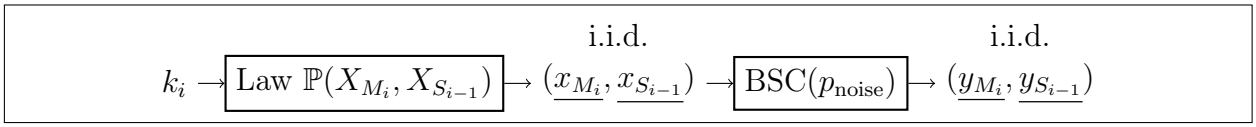


Figure 6.1: Observable leakage corresponding to exponent bit k_i

The probabilities equation (6.8) depend on the unknown ratio p/R. The crucial observation is that the attacker knows the position of all squaring and all multiplications. Lemma 6.6 provides concrete formula, which allow to estimate p/R. Of course, this estimation step is only necessary for RSA with CRT but not for RSA without CRT. We begin with a lemma, which will be needed below.

Lemma 6.5 *It is*

$$\mathbb{E}(X_{M_i}) = \mathbb{P}(X_{M_i} = 1) = \frac{1}{4} \cdot \frac{p}{R} \tag{6.17}$$

$$\mathbb{E}(X_{S_i}) = \mathbb{P}(X_{S_i} = 1) = \frac{1}{3} \cdot \frac{p}{R} \tag{6.18}$$

$$\frac{p}{R} = 3\,\mathbb{E}(X_{S_i}) = 2\,\mathbb{E}(X_{M_i}) + 1.5\,\mathbb{E}(X_{S_i}) \tag{6.19}$$

Proof Since X_{M_i} and X_{S_i} assume values in $\{0, 1\}$ the left-hand equations in equation (6.17) and equation (6.18) are obvious while the right equation follow immediately from equation (6.4) and equation (6.5), respectively. For $k_i = 0$, for instance,

$$\mathbb{P}(X_{S_i} = 1) = \int_0^1 \int_0^1 \int_0^{s_{t+1,0}s_{t+1,1}p/R} 1\, ds_{t+1,1} ds_{t+1,0} ds_{t,1} = \frac{1}{4} \cdot \frac{p}{R}$$

We note that the probabilities equation (6.18) and equation (6.17) were already verified in [Sch00] and, for instance, , respectively, the latter by other mathematical methods. Formula equation (6.19) follows directly from equation (6.17) and equation (6.18).

The ER-values x_{M_i} and x_{S_i} are determined (or more precisely: guessed) on the basis of single-trace template attacks. In particular, their guesses $\widetilde{x}_{M_i}$ and $\widetilde{x}_{S_i}$ might be incorrect with some probability. We denote the corresponding random variables (referring to the guessed ER values) by $\widetilde{X}_{M_i}$ and $\widetilde{X}_{S_i}$. In the following we assume that

$$\begin{aligned} &\mathbb{P}(\widetilde{X}_{M_i} = v \mid X_{M_i} = 1 - v) = \mathbb{P}(\widetilde{X}_{S_i} = v \mid X_{S_i} = 1 - v) = p_{\text{noise}} \\ &\qquad \text{for } t = 0, \ldots, l-1 \text{ and } v = 0, 1\,, \end{aligned} \tag{6.20}$$

and likewise, for the initialization of the registers R_0 and R_1 in algorithm 3.2. In other words: The probability of guessing an ER value incorrectly is $p_{\text{noise}} \geq 0$, independently of the true value. Of course, $p_{\text{noise}} = 0$ characterizes a perfect side channel measurement. Lemma 6.6(iii) is the generalization of equation (6.19) for noisy measurements. As noted in lemma 6.8 this allows the estimation of p/R and p_{noise}.

Lemma 6.6

$$\frac{p}{R} = \frac{12\,\mathbb{E}(\widetilde{X}_{S_i}) - 12\,\mathbb{E}(\widetilde{X}_{M_i})}{1 + 6\,\mathbb{E}(\widetilde{X}_{S_i}) - 8\,\mathbb{E}(\widetilde{X}_{M_i})} \tag{6.21}$$

$$p_{\text{noise}} = 4\,\mathbb{E}(\widetilde{X}_{M_i}) - 3\,\mathbb{E}(\widetilde{X}_{S_i}) \tag{6.22}$$

Proof Since $\widetilde{X}_{S_i}$ is $\{0, 1\}$-valued we obtain

$$\begin{aligned} \mathbb{E}(\widetilde{X}_{S_i}) &= \mathbb{P}(\widetilde{X}_{S_i} = 1) \\ &= \mathbb{P}(\widetilde{X}_{S_i} = 1 \mid X_{S_i} = 1)\mathbb{P}(X_{S_i} = 1) + \mathbb{P}(\widetilde{X}_{S_i} = 1 \mid X_{S_i} = 0)\mathbb{P}(X_{S_i} = 0) \\ &= (1 - p_{\text{noise}})\frac{p}{3R} + p_{\text{noise}}\left(1 - \frac{p}{3R}\right)\,. \end{aligned}$$

and similarly

$$\mathbb{E}(\widetilde{X}_{M_i}) = \mathbb{P}(\widetilde{X}_{M_i} = 1)$$

$$\begin{aligned} &= \mathbb{P}(\widetilde{X}_{M_i} = 1 \mid X_{M_i} = 1)\mathbb{P}(X_{M_i} = 1) + \mathbb{P}(\widetilde{X}_{M_i} = 1 \mid X_{M_i} = 0)\mathbb{P}(X_{M_i} = 0) \\ &= (1 - p_{\text{noise}})\frac{p}{4R} + p_{\text{noise}}\left(1 - \frac{p}{4R}\right) \ . \end{aligned}$$

Solving these equations for (p/R) and p_{noise} yields equation (6.21) and equation (6.22).

In lemma 6.7 below $(e_{M_1}, e_{S_1}, \ldots, e_{M_u}, e_{S_u}) \in \{0,1\}^{2u}$ represents the 'error vector'. The non-zero entries give the positions at which the guessed extra reduction vector $(\widetilde{x}_{M_i}, \widetilde{x}_{S_i}, \ldots, \widetilde{x}_{M_{i-u+1}}, \widetilde{x}_{S_{i-u+1}})$ are incorrect.

Lemma 6.7 *(i)*

$$\begin{aligned} &\mathbb{P}_{\vec{\theta}}(\widetilde{X}_{M_{i-j+1}} = \widetilde{x}_{M_{i-j+1}}, \widetilde{X}_{S_{i-j+1}} = \widetilde{x}_{S_{i-j+1}} \mid j = 1, \ldots, u) = \qquad (6.23) \\ &= \sum_{\substack{0 \le e_{j(M)}, e_{j(S)} \le 1 \\ 1 \le j \le u}} \mathbb{P}_{\vec{\theta}}(X_{M_{i-j+1}} = \widetilde{x}_{M_{i-j+1}} \oplus e_{j(M)}, X_{S_{i-j+1}} = \widetilde{x}_{S_{i-j+1}} \oplus e_{j-i+1(S)} \mid j = 1, \ldots, u) \\ &\quad \times p_{\text{noise}}^{\text{ham}(e_{M_1}, \ldots, e_{S_u})} (1 - p_{\text{noise}})^{2u - \text{ham}(e_{M_1}, \ldots, e_{S_u})} \ . \end{aligned}$$

(ii) For each hypothesis $\vec{\theta} \in \{0,1\}^u$ and each (guessed) extra reduction vector $(\widetilde{x}_{M_i}, \widetilde{x}_{S_i}, \ldots, \widetilde{x}_{M_{i-t+1}}, \widetilde{x}_{S_{i-u+1}})$ we have

$$\mathbb{P}_{\vec{\theta}}(\widetilde{X}_{M_i} = \widetilde{x}_{M_i}, \ldots, \widetilde{X}_{S_{i-u+1}} = \widetilde{x}_{S_{i-u+1}}) = \mathbb{P}_{\vec{1} - \vec{\theta}}(\widetilde{X}_{M_i} = \widetilde{x}_{M_i}, \ldots, \widetilde{X}_{S_{i-u+1}} = \widetilde{x}_{S_{i-u+1}}) \, . \qquad (6.24)$$

Proof The term $p_{\text{noise}}^{\text{ham}(e_{M_1}, \ldots, e_{S_u})} (1 - p_{\text{noise}})^{2u - \text{ham}(e_{M_1}, \ldots, e_{S_u})}$ quantifies the probability for the error vector $(e_{M_1}, e_{S_1}, \ldots, e_{M_u}, e_{S_u})$. This fact and the definition of the conditional probability imply equation (6.23). Assertion (ii) follows immediately from (i) and lemma 6.3(ii), applied to the particular right-hand probabilities in equation (6.23).

The last lemma of this section explains how to estimate the ratio p/R and the probability p_{noise}.

Lemma 6.8 *Assume that the attacker has observed Q side channel traces. Then*

$$\widetilde{\mu}_M = \frac{1}{lQ} \sum_{q=1}^{Q} \sum_{t=0}^{l-1} = \widetilde{x}_{M_i}^q \qquad (6.25)$$

provides an estimator for $\mathbb{E}(\widetilde{X}_{M_i}^q)$ and analogously

$$\widetilde{\mu}_Q = \frac{1}{lQ} \sum_{q=1}^{Q} \sum_{t=0}^{l-1} = \widetilde{x}_{S_i}^q \qquad (6.26)$$

for $\mathbb{E}(\widetilde{X}_{S_i}^q)$. The index q refers to the numbering of the side channel traces.
(ii) Substituting $\widetilde{\mu}_M$ and $\widetilde{\mu}_Q$ for $\mathbb{E}(\widetilde{X}_{M_i}^q)$ and $\mathbb{E}(\widetilde{X}_{S_i}^q)$ into equation (6.21) and equation (6.22) yields estimates $\widetilde{p/R}$ and $\widetilde{p_{\text{noise}}}$.
(iii) For perfect measurements alternatively equation (6.19) might be used to estimate p/R. Compared to the mid-term the right-hand term considers twice as many Montgomery multiplications and thus should provide a more precise estimate.

Proof Straight-forward.

Example [estimation of p/R and p_{noise}] For different exponents of 512 bits-length, we estimate $\widetilde{p/R}$ and $\widetilde{p_{\text{noise}}}$ for two moduli (`RSA-1024-p` and `RSA-1024-q` defined and different value of p_{noise} depending of the number of side channel traces Q. For each values of Q between 1 and 500, we compute $\widetilde{p/R}$ using equation (6.21) and $\widetilde{p_{\text{noise}}}$ using equation (6.22) for the different exponents and the found values are represented using a boxplot (deciles/quartile/median values) on the figure 6.2.

6.4 The optimal decision strategy

Lemma 6.9 provides the optimal decision strategy for the individual decisions, i.e. for guessing the parameter set $\{\vec{\theta}, 1-\vec{\theta}\}$ for the particular u-tuples $(k_i, \ldots, k_{i-u+1})$. The decision strategy exploits the information from the observed (guessed) ER-vectors from Q side channel traces. For $p_{\text{noise}} = 0$ lemma 6.9 describes the situation in case of perfect measurements.

Lemma 6.9 (Maximum likelihood estimator) *Assume that the key k has been selected randomly and that the attacker has no information on the subkey $(k_i, \ldots, k_{i-u+1})$. Let*

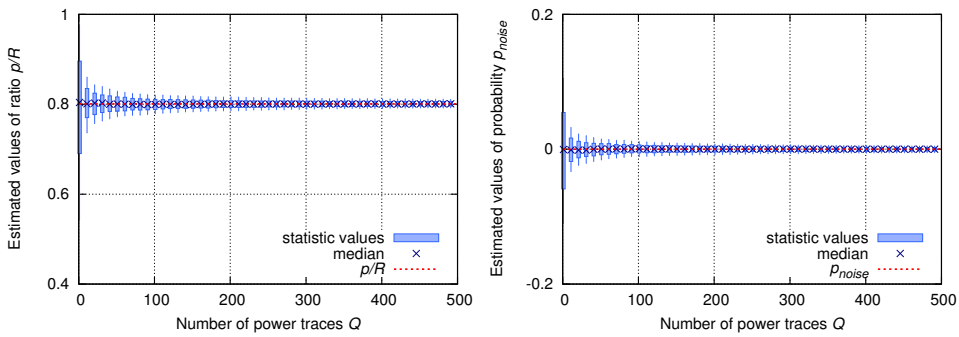
$$\hat{\vec{\theta}} = \underset{\vec{\theta}\in\{0,1\}^u}{\operatorname{argmax}} \prod_{q=1}^{Q} \mathbb{P}_{\vec{\theta}}(\widetilde{X}^q_{M_{i-j+1}} = \widetilde{x}^q_{M_{i-j+1}}, \widetilde{X}^q_{S_{i-j+1}} = \widetilde{x}^q_{S_{i-j+1}} \mid j = 1, \ldots, u)\,. \tag{6.27}$$

(i) $\hat{\vec{\theta}}$ maximizes the right-hand side of equation (6.27) iff $\vec{1} - \hat{\vec{\theta}}$ maximizes the right-hand side of equation (6.27). It thus suffices to compute the right-hand term of equation (6.27) for all $\vec{\theta} \in \{0,1\}^u \mid \theta_u = 0\}$.
(ii) The attacker decides for

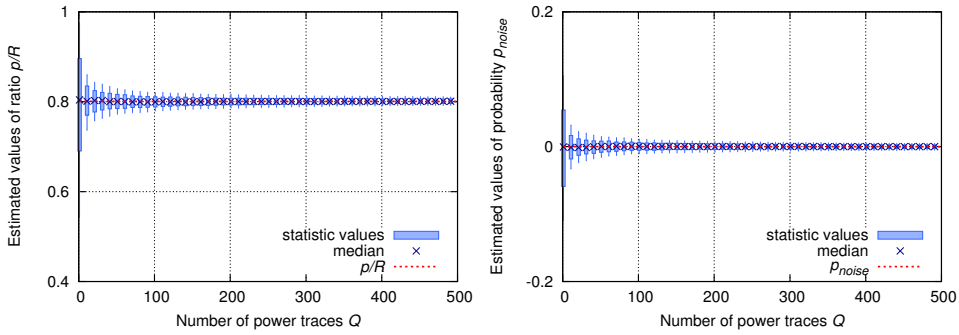
$$(k_i \oplus k_{i-1} = \hat{\theta}_1 \oplus \hat{\theta}_2, \ldots, k_{i-u+2} \oplus k_{i-u+1} = \hat{\theta}_{u-1} \oplus \hat{\theta}_u)\,. \tag{6.28}$$

This is the optimal decision strategy.

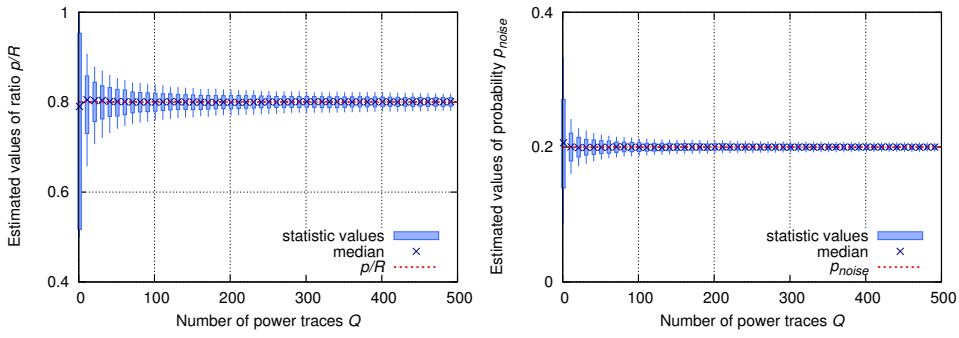
Proof The first assertion of (i) follows from lemma 6.3(ii), and the second is an immediate consequence of the first. With regard to the assumptions on k and the on the subkey $(k_i, \ldots, k_{i-u+1})$ we interpret the unknown sub-key $(k_i, \ldots, k_{i-u+1})$ as a realization of random variable, which is uniformly distributed on $\{0,1\}^u$. Then $(k_i \oplus k_{i-1}, \ldots, k_{i-u+2} \oplus k_{i-u+1})$ may be viewed as a realization of a random variable, which is uniformly distributed on $\{0,1\}^{u-1}$. Further, $(k_i, \ldots, k_{i-u+1}) \in \{\vec{\theta}, \vec{1} - \vec{\theta}\}$ iff $(k_i \oplus k_{i-1} = \theta_i \oplus \theta_{i-1}, \ldots, k_{i-u+2} \oplus k_{i-u+1} = \theta_{i-u+2} \oplus \theta_{i-u+1})$. Hence equation (6.27) yields the maximum likelihood estimator for the transformed sub-key $(k_i \oplus k_{i-1}, \ldots, k_{i-u+2} \oplus k_{i-u+1})$. If we assume that each false decision is equally bad the optimal decision strategy (Bayes strategy against the uniform distribution on $\{0,1\}^{u-1}$) is given by the maximum likelihood estimator, which completes the proof of lemma 6.9.



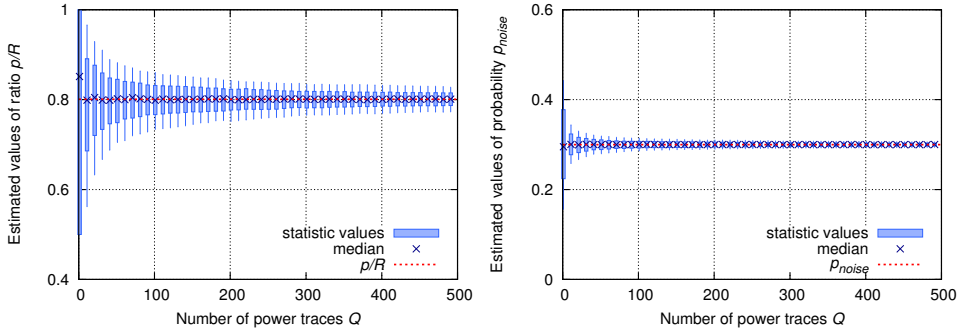
(a) $p/R \simeq 0.800907$ and $p_{\text{noise}} \simeq 0.00$



(b) $p/R \simeq 0.789290$ and $p_{\text{noise}} \simeq 0.00$



(c) $p/R \simeq 0.800907$ and $p_{\text{noise}} \simeq 0.20$



(d) $p/R \simeq 0.789290$ and $p_{\text{noise}} \simeq 0.30$

Figure 6.2: Statistic box-plot to estimated the ratio p/R and the probability p_{noise} in function of side channel traces Q using 1.000 exponents values

Remark
(i) Lemma 6.9 assumes that p/R and p_{noise} are known. Substituting $\widetilde{p/R}$ and $\widetilde{p_{\text{noise}}}$ into equation (6.27) yields estimates for the probabilities
$\mathbb{P}_{\vec{\theta}}(\widetilde{X}_{M_i} = \widetilde{x}_{M_i}, \ldots, \widetilde{X}_{S_{i-u+1}} = \widetilde{x}_{S_{i-u+1}})$.
(ii) In the proof of lemma 6.9 we assume that $(k_i, \ldots, k_{i-u+1})$ is a realization of a uniformly distributed random variable on $\{0,1\}^u$. This assumption may not be justified for $i = l-1, l-2$ and in particular not for $i = 1$ since $k_0 = 1$. Since we are finally only interested in the distribution on $\{0,1\}^{u-1}$ this relaxes the situation.
(iii) In the proof of lemma 6.9 we assume that each false decision is equally bad. This assumption is certainly justified if all transformed sub-keys $(k_i \oplus k_{i-1}, \ldots, k_{i-u+2} \oplus k_{i-u+1})$ are treated independently. If error correction strategies are applied (for $u > 2$) the situation may change.

6.5 Summarize of attack and success rate

The decision strategy in lemma 6.9 is based on the observed extra-reduction for each multiply and square operations for Q calls of the cryptographic operation with a static key k of l-bits length ($k_{l-1} = 1$ as describe in algorithm 3.2). For each u-tuples of random variable $(x_{M_i}, x_{S_i}, \ldots, x_{M_{i-u}}, x_{S_{i-u}}) \in \{0,1\}^u$, the attacker estimated the $\vec{\theta_i}$ value using the maximum likelihood estimator like described in lemma 6.9 using only the probabilities $\mathbb{P}_\theta$. The following algorithm 6.1 permits to retrieve the majority parts of key bits values. The u least significant bit values cannot be determinate[1].

Remark Warning, when the number of queries increase, for each $\vec{\theta}$, the value $T_{\vec{\theta}}$ decreases very quickly (see line 9) and a computer wrongly considers them equal to zero. The algorithm 6.2 permits to correct this computation problem and improved the efficiency of the attack.

[1]The maximal value of u is 5 in our experiments, so we can determinate the least significant bit by a brute force attack. Only $2^5 = 32$ gcd must be required to find the correct exponent value.

Algorithm 6.1: Optimal extra-reduction attack using maximum likelihood estimator

Require: $(\underline{x_{M_i}}, \underline{x_{S_i}}, \ldots, \underline{x_{M_{i-u}}}, \underline{x_{S_{i-u}}})$, a set of Q u-tuples of $(l-u)$ bits

Ensure: A guess key value $\hat{k}$

Attack phase

Determinate the ratio p/R and the probability p_{noise} (or their estimated values using lemma 6.8)
for each $\vec{\theta} \in \{0,1\}^u$ **do**
  Compute the law probabilities $\mathbb{P}_{\vec{\theta}}(X_{M_i}, X_{S_i}, X_{M_{i-1}}, X_{S_{i-1}})$ using the ratio p/R and the p_{noise} by equation (6.23)
end for
for $i = l-2$ **down to** u by step u **do**
  for each $\vec{\theta} \in \{0,1\}^u$ **do**
    $T_{\vec{\theta}} = 1$
    for $q = 1$ **to** Q **do**
      $T_{\vec{\theta}} \leftarrow T_{\vec{\theta}} \times \mathbb{P}_{\vec{\theta}}(X_{M_i} = x^q_{M_i}, X_{S_i} = x^q_{S_i}, X_{M_{i-1}} = x^q_{M_{i-1}}, X_{S_{i-1}} = x^q_{S_{i-1}})$
    end for
  end for
  $\vec{\hat{\theta}}_i = (\hat{\theta}_i(0), \ldots, \hat{\theta}_i(u)) \leftarrow \text{argmax}_{\vec{\theta}}(T_{\vec{\theta}})$ ▷ see lemma 6.9
end for

Computation of the estimated key bit value

$\hat{k}_{l-1} \leftarrow 1$ ▷ by definition of the key
$\hat{k}_{l-2} \leftarrow 0$ or 1 ▷ using the lemma 5.11
for $i = l-2$ **down to** u **do**
  for $j = 0$ **to** u **do**
    $\hat{k}_{i-j} \leftarrow \hat{\theta}_i(j) \oplus \hat{k}_{i-j+1}$ ▷ see equation (6.28)
  end for
  $i \leftarrow i - u$
end for
return $\hat{k} = (\hat{k}_l \hat{k}_{l-1} \hat{k}_{l-2} \ldots \hat{k}_0)_2$ ▷ the u least significant bits are no determinate

Algorithm 6.2: Improved of optimal extra-reduction attack using maximum likelihood estimator

Require: $(\underline{x_{M_i}}, \underline{x_{S_i}}, \ldots, \underline{x_{M_{i-u}}}, \underline{x_{S_{i-u}}})$, a set of Q u-tuples of $(l-u)$ bits

Ensure: A guess key value $\hat{k}$

Attack phase

Determinate the ratio p/R and the probability p_{noise} (or their estimated values using lemma 6.8)
for each $\vec{\theta} \in \{0,1\}^u$ **do**
  Compute the law probabilities $\mathbb{P}_{\vec{\theta}}(X_{M_i}, X_{S_i}, X_{M_{i-1}}, X_{S_{i-1}})$ using the ratio p/R and the p_{noise} by equation (6.23)
end for
for $i = l-2$ **down to** u by step u **do**
  $Accum(X_{M_i}, X_{S_i}, \ldots, X_{M_{i-u}}, X_{S_{i-u}}) \leftarrow \vec{0}$ ▷ $\vec{0} = [0]^{2^{(2\times u)}}$
  for $q = 1$ **to** Q **do**
    $Accum(X_{M_i} = x^q_{M_i}, X_{S_i} = x^q_{S_i}, \ldots, X_{M_{i-u}} = x^q_{M_{i-u}}, X_{S_{i-u}} = x^q_{S_{i-u}}) + +$ ▷ Incrementing by 1
  end for
  for each $\vec{\theta} \in \{0,1\}^u$ **do**
    $T_{\vec{\theta}} \leftarrow 0$
    for each t-uples $(x_{M_i}, x_{S_i}, \ldots, x_{M_{i-u}}, x_{S_{i-u}}) \in \{0,1\}^{2^u}$ **do**
      $T_{\vec{\theta}} \leftarrow T_{\vec{\theta}} + Accum(x_{M_i}, x_{S_i}, \ldots, x_{M_{i-u}}, x_{S_{i-u}}) \times \ln\left(\mathbb{P}_{\vec{\theta}}(x_{M_i}, x_{S_i}, \ldots, x_{M_{i-u}}, x_{S_{i-u}})\right)$
    end for
  end for
  $\hat{\vec{\theta}}_i = (\hat{\theta}_i(0), \ldots, \hat{\theta}_i(u)) \leftarrow \text{argmax}_{\vec{\theta}}(T_{\vec{\theta}})$ ▷ see lemma 6.9
end for

Computation of the estimated key bit value

$\hat{k}_{l-1} \leftarrow 1$ ▷ by definition of the key
$\hat{k}_{l-2} \leftarrow 0$ or 1 ▷ using the lemma 5.11
for $i = l-2$ **down to** u **do**
  for $j = 0$ **to** u **do**
    $\hat{k}_{i-j} \leftarrow \hat{\theta}_i(j) \oplus \hat{k}_{i-j+1}$ ▷ see equation (6.28)
  end for
  $i \leftarrow i - u$
end for
return $\hat{k} = (\hat{k}_l \hat{k}_{l-1} \hat{k}_{l-2} \ldots \hat{k}_0)_2$ ▷ the u least significant bits are no determinate

In order to compare the chapter 5 and this optimize method, we compute the success rate of one bit value for $u = 2$. The two attacks are made using the modulo `RSA-1024-p` defined in section 3.5.3 and the distinguisher is the maximum likelihood for the two attacks. The figure 6.3 represents the evolution of the success rate to retrieve one bit value in function of the number of queries.

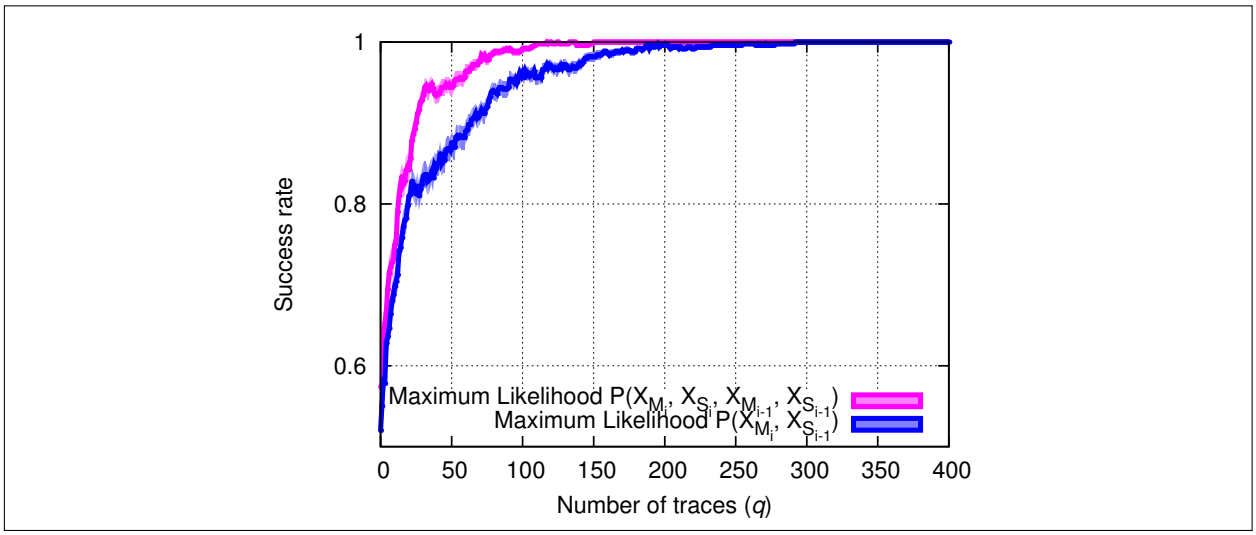


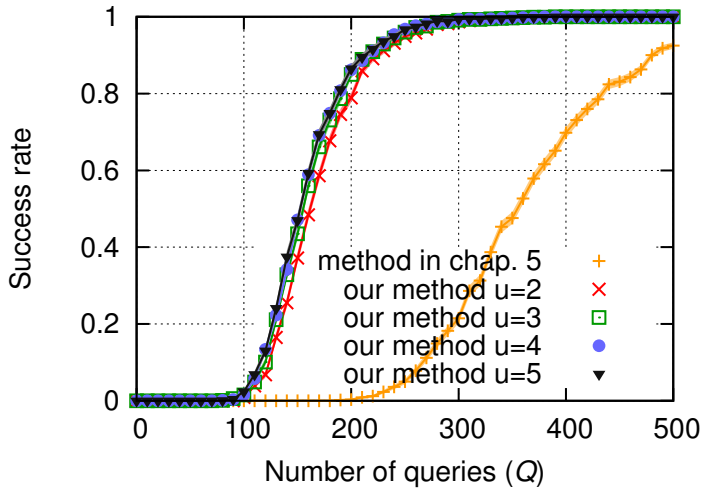
Figure 6.3: Evolution of the success rate of one bit using 500 experiments between the chapter 5 attack version and the optimized version with $u = 2$ without noise in acquisition

The figure 6.3 permits to conclude than the number of queries decreases when we use the optimize method. In order to compare the evolution of the success rate, when the u value increased, we define the success rate of a whole exponent value.

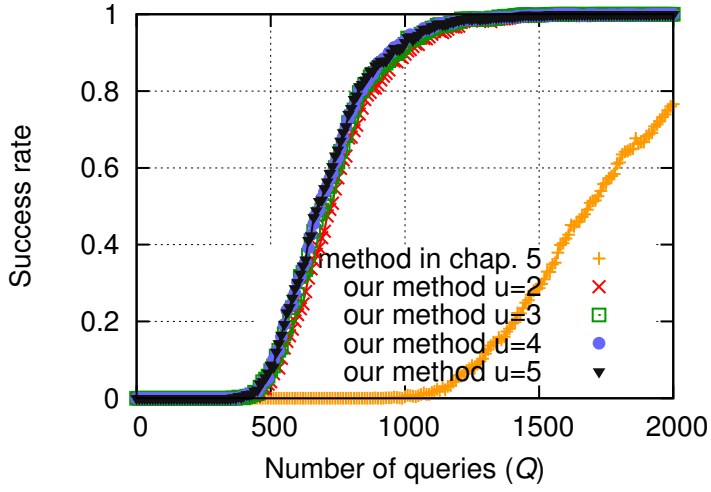
Definition 6.10 (success rate of an attack) *The success rate of an attack is the number of succeed attacks over the number of experiments. The attack succeed when all the key bits of entire exponent are founded. If only one bit is bad, then the attack failed.*

For different exponents of 512-bit length, we estimate the success rate of the attack for the modulo (`RSA-1024-q` defined in section 3.5.3, for different probability p_{noise} and different values of u depending of the number of side channel traces Q. In this part, only the most probable exponent was considered. If the most probable exponent is different with only one bit, then the attack failed. On the figure 6.4, a comparison between the attack described in chapter 5 and our method for u between 2 and 5.

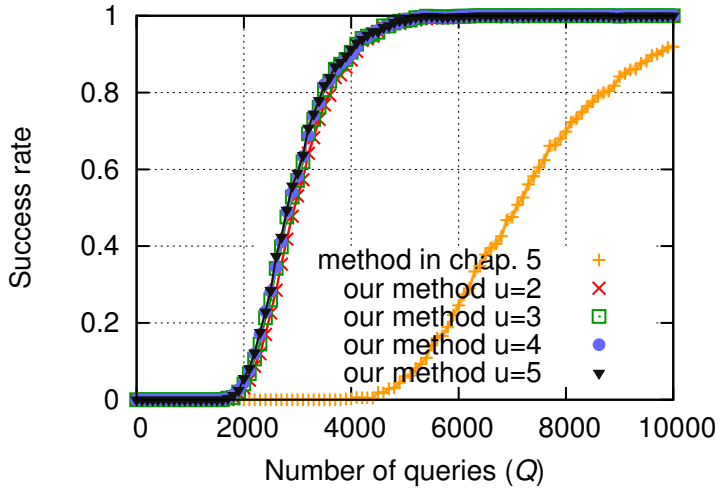
Here one can observe our method for different u values increases significantly the success rate compared to the method described in chapter 5. The number of side channel trace needed to succeed the attack is divided by a factor greater than 2. The gain between the increasing u values is not significant.



(a) $p_{\text{noise}} = 0.00$



(b) $p_{\text{noise}} = 0.10$



(c) $p_{\text{noise}} = 0.20$

Figure 6.4: Success rate for a entire exponent using 1.000 exponents values depending of the number of side channel trace Q without different probability p_{noise}

6.6 Conclusion

The extra-reduction analysis using the dependence of the consecutive operations permits to attack a blinded regular exponentiation algorithm with a limited number of acquisition. This chapter presents the method to analyze more than two iteration loops in the "Montgomery Ladder" algorithm. The same method can be used for the other exponentiation algorithm like "square-and-always-multiply". To resume all the extra-reduction analysis, we can make this table 6.1. The criteria are :

- the differential protection named the message blinded or unknown (see section 3.3.3),
- the simple power analysis protection named the regular protection for the exponentiation algorithm (see definition 3.3),
- the compensation of the extra-reduction by a fake operation named constant time (see example in listing 5.2),
- the differential protection named the exponent blinded (see section 3.3.3),
- the fault and differential protection named modular extension (see section 3.6.2).

	Message Blinded	Regular Algorithm	Constant Time	Exponent Blinded	Modular Extension
`ERA 1` [SKQ01, SW03, ASK05, AS08]	✓ Yes	✗ No	✗ No	✗ No	✗ No
`ERA 2` [Sch15]	✗ No	✗ No	✗ No	✓ Yes	✗ No
`ERA 3` [Sch00, WT01, Sch02]	✓ Yes	✓ No	✗ Yes	✗ No	✗ No
Our ERA Chapters 5-6	✓ Yes	✓ No	✓ Yes	✗ No	✗ No

Table 6.1: Summarize of the extra-reduction analysis published before 2017

Two axes for new research are opening:

- **Exponent blinding.** Applied this method when the exponent in blinded with the classical method. Note that Berzati and al. in [BCG10] show that the exponent blinding is partially ineffective on some bits depending on the chosen modulo. Our new idea is than this bias is sufficient to retrieve the right exponent value using the extra-reduction leakage information.
- **Modular extension.** Applied this method when the modular is randomized by a small random. The formulas of the probabilities $\mathbb{P}_{\vec{\theta}}$ in lemma 6.3 must be adapted. Our new idea is the ratio p/R has an impact for the probability but our hypothesis is that the attack works with an adaptation of this ratio with a "randomized ratio" between each queries.

Chapter 7

Correction of Modular Extension countermeasure for elliptic curves cryptography

This collaborative work with Sylvain Guilley, Martin Moreau, Zakaria Najm, and Pablo Rauzy is presented to PROOFS workshop in 2016[DGM+16]. This work is also accepted at JCEN in 2017[DGM+17].

In this chapter, we present a countermeasure against fault attack exclusively on Elliptic Curve Cryptography. We focus on countermeasures which guarantee the integrity of the computation result, hence covering most existing and future fault attacks. Namely, we study the *modular extension* protection scheme in previously existing and newly contributed variants of the countermeasure on elliptic curve scalar multiplication algorithms. We find that an existing countermeasure is incorrect and we propose new "test-free" variant of the modular extension scheme that fixes it. As we can see on figure 7.1, this countermeasure named Modular Extension Countermeasure (MEC) protects against fault attack and simple side channel attack with unified formula.

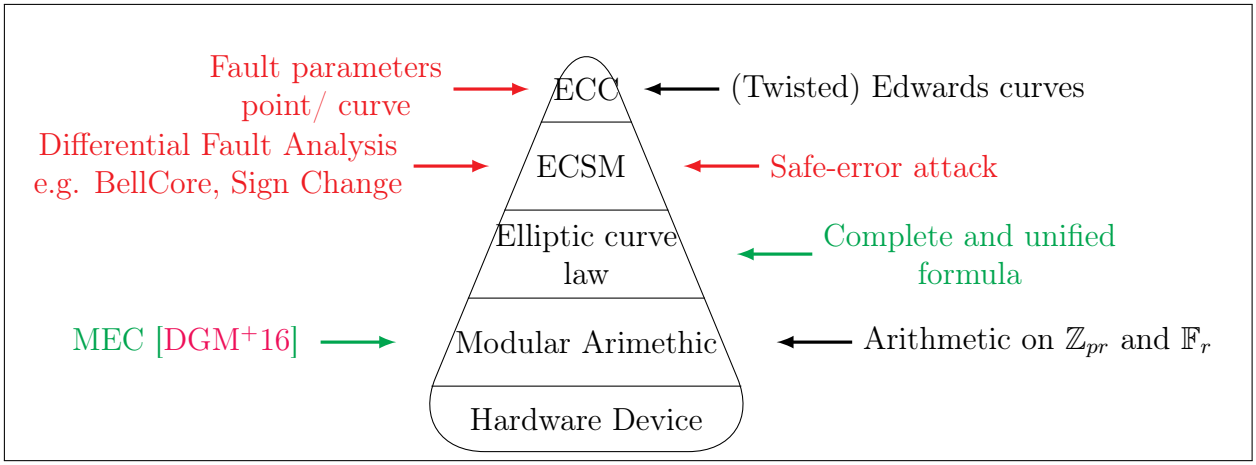


Figure 7.1: Countermeasure protects these fault attack and side channel presented in this chapter

Contents

7.1 Introduction

Properly used cryptography is a key building block for secure information exchange. Thus, implementation-level hacks must be considered seriously in addition to the threat of cyber-attacks. In particular, fault injection attacks target physical implementations of secured devices in order to induce exploitable errors.

Countermeasures against fault injection attacks have been proposed for elliptic curve computations, but they are actually incorrect. For example, in [BOS06], Blömer, Otto, and Seifert (BOS) propose a countermeasure based on the modular extension idea of Shamir for CRT-RSA [Sha99]. The problem is that the modular extension scheme cannot actually be applied as is to Weierstrass elliptic curve, because the tests in the ECDBL and ECADD operations are not true at the same times for the computation in $\mathbb{Z}_{pr}$ and the one in $\mathbb{Z}_r$, which breaks the scheme and will yield false negatives. This behavior can be a *serious security issue* as it reveals information about the inputs.

In 2010 Joye patented [Joy10] essentially the same countermeasure except it uses $\mathbb{Z}_{r^2}$ and $\mathbb{Z}_{pr^2}$ instead of $\mathbb{F}_r$ and $\mathbb{Z}_{pr}$, which does not address the raised issues.

In [BV07], Baek and Vasyltsov (BV) propose a countermeasure based on modular extension and point verification. This countermeasure can be applied only on Weierstrass curve, and the overhead computation is 48% for curve with parameters on 256 bits. The particularity of this countermeasure is that instead of computing a sibling ECSM on a smaller curve $\mathcal{E}(\mathbb{F}_r)$ to compare with its redundant counterpart over $\mathcal{E}(\mathbb{Z}_{pr})$, it only checks whether the point obtained by reducing the result $\mathcal{E}(\mathbb{Z}_{pr})$ modulo r is on the $\mathcal{E}(\mathbb{F}_r)$ curve (i.e., whether it satisfies the curve equations modulo r). Because of that, BV does not suffer from BOS problem. However, the correctness of BV comes with a drawback: indeed, faults may go undetected if they happen before $\mathcal{O}_\mathcal{E}$ (the point at infinity) is reached in the

computation modulo r as the intermediate point quickly tends to $(0:0:0)$ in projective coordinates and stays there until the end.

It is actually possible to get the best of both world: what is needed is BOS approach (i.e., pure modular extension scheme) but without the problematic tests. Luckily, Edwards curves allow to perform ECC without tests thanks to a complete addition law, as it was detailed in proposition 2.21. But before, we will formally analyze the security of the modular extension scheme when the implementation is test-free.

The main idea of this chapter was presented by Marc Joye in FDTC 2013 [Joy13] but also in rump session at CRYPTO'2000, without formally proved this countermeasure. In this chapter, we take advantage of the speed-up record on ECDSA computation using the twisted Edwards curve `Ed25519` [BDL+12] coded with NaCl crypto-library [BLS12]. We propose a new countermeasure against faults injection based on modulus extension with only one "test-free" addition operation using *complete* and *unified* formulas of addition point on Edwards and twisted Edwards curves. This allows for a *synchronized* computation in $\mathbb{F}_p$ and $\mathbb{F}_r$ while computing in $\mathbb{Z}_{pr}$. Our countermeasure is new insofar as we give explicit conditions on the prime r: they happen to be easily met in the case of Edwards curves (see section 7.5.1), whereas they restrict the number of possible r to a little number of values for the popular twisted Edwards curves (see section 7.5.2). The overhead computation of this countermeasure is 28% for Edwards curve and 39% for twisted Edwards curve on 32-bit processors, such as an ARM Cortex-M4.

The rest of the chapter is organized as follows. Section 7.2 is the theoretical analysis of our new countermeasure using the modular extension with test-free. Details of the security bound of this countermeasure are described in section 7.3. The description of our countermeasure to make the modular extension without test in the elliptic curve operation is in section 7.4. Section 7.5 explains the overhead of computation and some examples of our countermeasure.

7.2 Security Analysis of Modular Extension

Definition 7.1 (Fault model) *We consider an attacker who can fault data by randomizing or zeroing any intermediate variable, and fault code instruction by skipping any number of consecutive instructions.*

Remark The three fault models have been described several times in the literature. For example, randomizing faults are discussed in [BDL97], zeroing faults in [Cla07], and instruction skip faults in [MHER14].

Definition 7.2 (Attack order) *We call order of the attack the number of faults (in the sense of definition 7.1) injected during the target execution.*

In the rest of this section, we focus mainly on the resistance to first-order attacks on data.

Definition 7.3 (Secure algorithm) *An algorithm is said secure if it is correct and if it either returns the right result or an* error *constant when faults have been injected, with an overwhelming probability.*

Theorem 7.4 (Security of test-free modular extension) *Test-free algorithms protected using the modular extension technique, are secure as per definition 7.3. In particular, the probability of non-detection is inversely proportional to the security parameter r.*

Proof *Faulted results are polynomials of faults.* The result of an asymmetric cryptography computation can be written as a function of a subset of the intermediate variables, plus some inputs if the intermediate variables do not suffice to finish the computation. We are interested in the expression of the result as a function of the intermediate variables which are the target of a transient or permanent fault injection. We give the formal name $\widehat{x}$ to any faulted variable x. For convenience, we denote them by $\widehat{x_i}$, $1 \leq i \leq t$, where $t \geq 1$ is the number of injected faults. The result consists in additions, subtractions, and multiplications of those formal variables (and inputs). Such expression is a multivariate polynomial. If the inputs are fixed, then the polynomial has only t formal variables. We call it $P(\widehat{x_1}, \ldots, \widehat{x_t})$. For now, let us assume that $t = 1$, i.e., that we face a single fault. Then P is a mono-variate polynomial. Its degree δ is the multiplicative depth of $\widehat{x_1}$ in the result.

A fault is not detected if and only if $P(\widehat{x_1}) = P(x_1) \bmod r$, whereas $P(\widehat{x_1}) \neq P(x_1) \bmod p$. Notice that the latter condition is superfluous insofar since if it is negated then the effect of the fault does not alter the result in $\mathbb{F}_p$.

Non-detection probability is inversely proportional to r. As the faulted variable $\widehat{x_1}$ can take any value in $\mathbb{Z}_{pr}$, the non-detection probability $\mathbb{P}_{\text{n.d.}}$ is given by:

$$\begin{aligned}\mathbb{P}_{\text{n.d.}} &= \frac{1}{pr-1} \times \sum_{\widehat{x_1} \in \mathbb{Z}_{pr} \setminus \{x_1\}} \mathbb{1}_{P(\widehat{x_1}) \,=\, P(x_1) \bmod r} \\ &= \frac{1}{pr-1} \times \Big(-1 + p \sum_{\widehat{x_1}=0}^{r-1} \mathbb{1}_{P(\widehat{x_1}) \,=\, P(x_1) \bmod r} \Big). \end{aligned} \tag{7.1}$$

Here, $\mathbb{1}_{\text{condition}}$ is an indicator function: it is equal to 1 (resp. 0) if the condition is true (resp. false).

Let $\widehat{x_1} \in \mathbb{Z}_r$, if $P(\widehat{x_1}) = P(x_1) \bmod r$, then $\widehat{x_1}$ is a root of the polynomial $\Delta P(\widehat{x_1}) = P(\widehat{x_1}) - P(x_1)$ in $\mathbb{Z}_r$. We denote by $\#\text{roots}(\Delta P)$ the number of roots of ΔP over $\mathbb{Z}_r$. Thus (7.1) computes $(p \times \#\text{roots}(\Delta P) - 1)/(pr-1) \approx \#\text{roots}(\Delta P)/r$.

Study of the proportionality constant. A priori, bounds on this value are broad since $\#\text{roots}(\Delta P)$ can be as high as the degree δ of ΔP in $\mathbb{Z}_r$, i.e., $\min(\delta, r-1)$. However, in practice, ΔP looks like a random polynomial over the finite field $\mathbb{Z}_r$, for several reasons:

- inputs are random numbers in most cryptographic algorithms, such as probabilistic signature schemes,
- the coefficients of ΔP in $\mathbb{Z}_r$ are randomized due to the reduction modulo r.

In such case, the number of roots is very small, despite the possibility of δ being large. See for instance [Leo06] for a proof that the number of roots tends to 1 as $r \to \infty$. Interestingly, random polynomials are still friable (i.e., they are clearly not irreducible) on average, but most factors of degree greater than one happen not to have roots in $\mathbb{Z}_r$.

Thus, we have $\mathbb{P}_{\text{n.d.}} \gtrsim \frac{1}{r}$, meaning that $\mathbb{P}_{\text{n.d.}} \geq \frac{1}{r}$ but is close to $\frac{1}{r}$. A more detailed study of the theoretical upper bound on the number of roots is available in following section 7.3.

The same law applies to multiple faults. In the case of multiple faults ($t > 1$), then the probability of non-detection generalizes to:

$$\begin{aligned}
\mathbb{P}_{\text{n.d.}} &= \tfrac{1}{(pr-1)^t} \times \sum_{\widehat{x_1},\ldots,\widehat{x_t} \in \mathbb{Z}_{pr}\setminus\{x_1\}\times\ldots\times\mathbb{Z}_{pr}\setminus\{x_t\}} \mathbb{1}_{P(\widehat{x_1},\ldots,\widehat{x_t})=P(x_1,\ldots,x_t) \bmod r} \\
&= \tfrac{1}{(pr-1)^t} \times \sum_{\widehat{x_2},\ldots,\widehat{x_t} \in \prod_{i=2}^{t} \mathbb{Z}_{pr}\setminus\{x_i\}} \left[\sum_{\widehat{x_1} \in \mathbb{Z}_{pr}\setminus\{x_1\}} \mathbb{1}_{P(\widehat{x_1},\ldots,\widehat{x_t})=P(x_1,\ldots,x_t) \bmod r} \right] \\
&= \tfrac{1}{(pr-1)^t} \times \sum_{\widehat{x_2},\ldots,\widehat{x_t} \in \prod_{i=2}^{t} \mathbb{Z}_{pr}\setminus\{x_i\}} [p \times \#\text{roots}(\Delta P) - 1] \\
&= \tfrac{1}{(pr-1)^t} \times (pr-1)^{t-1} [p \times \#\text{roots}(\Delta P) - 1] \\
&= \frac{p \times \#\text{roots}(\Delta P) - 1}{pr-1}.
\end{aligned} \tag{7.2}$$

Therefore, the probability not to detect a fault when $t > 1$ is identical to that for $t = 1$. Thus, we also have $\mathbb{P}_{\text{n.d.}} \approx \frac{1}{r}$ in the case of multiple faults of the intermediate variables. Note that this study does not take correlated faults into account. □

7.3 Theoretical Upper-Bound for #roots

It is interesting to study the theoretical upper bound on the number of roots in practical cases. Leont'ev proved in [Leo06] that if P is a random polynomial in $\mathbb{F}_p$ then $\#\text{roots}(P) \sim \text{Poisson}(\lambda = 1)$, i.e., $\mathbb{P}(\#\text{roots}(P) = w) = \frac{1}{ew!}$. In the case of $\Delta P \bmod r$, we know that there is always at least one root, when $\widehat{x_1} = x_1$, so we can rewrite $\Delta P(\widehat{x_1}) = P(\widehat{x_1}) - P(x_1) = R(\widehat{x_1}) \cdot a(\widehat{x_1} - x_1)$, where a is some constant, and R is indeed a random polynomial of degree $r-2$, owing to the modular reduction of ΔP by r. So we know that $\#\text{roots}(\Delta P) = 1 + \#\text{roots}(R)$, hence $\mathbb{P}(\#\text{roots}(\Delta P) = w) = \mathbb{P}(\#\text{roots}(R) = w-1)$, which is 0 if $w = 0$ and $\frac{1}{e(w-1)!}$ otherwise. We want the maximum value of roots w which has a "plausible" probability, let us say that is 2^{-p}, e.g., 2^{-256}. Since the values of a Poisson distribution of parameter $\lambda = 1$ are decreasing, we are looking for w such that: $\mathbb{P}(\#\text{roots}(R) = w-1) = \frac{1}{e(w-1)!} \leq 2^{-256}$. This would suggest that $w \gtrsim 58$.

This result means that $\mathbb{P}_{\text{n.d.}}$ is predicted to be at most $\frac{57}{r}$, with r being at least a 32-bit number, i.e., that $\mathbb{P}_{\text{n.d.}}$ is at maximum $\approx 2^{-26}$, and that this worst-case scenario has a probability of $\approx 2^{-256}$ of happening, in theory.

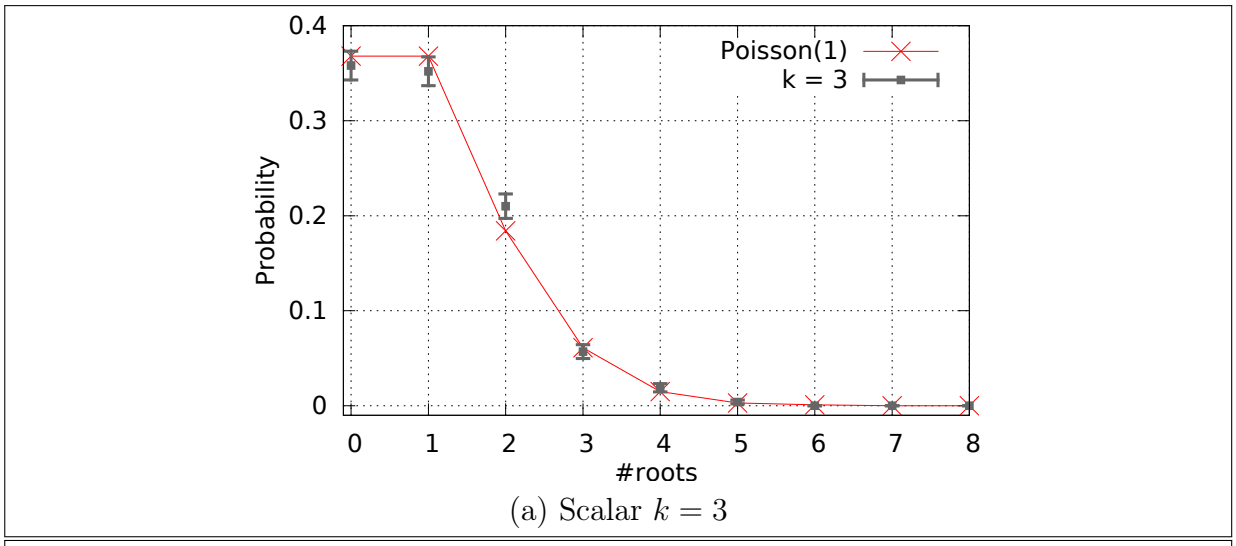
The figure 7.2 shows typical number of roots (obtained with SAGE) for practical cases in ECC, and compares them to the theoretical predictions. In this figure, we chose values of scalar k of the form $2^j - 1$, which maximize the number of operations, and thus the size and degree of the resulting ΔP polynomials. For each value of scalar k, we expressed the polynomial ΔP corresponding to the ECSM $[k]G$, and did so for a thousand random G. We then plotted for $w = 0$ to 8 the number of $[k]G$ for which $\#\text{roots}(\Delta P) = w + 1$ divided by 1000, that is the estimated probability $\hat{\mathbb{P}}(\#\text{roots}(\Delta P) = w+1)$. Let us denote by B the boolean random variable which is equal to one if ΔP has a $(w+1)$ roots, and zero otherwise. Our estimation of $\hat{\mathbb{P}}(\#\text{roots}(\Delta P) = w+1)$ is thus the expectation

of $\frac{1}{1000}\sum_{j=1}^{1000} B_j$. This random variable follows a binomial distribution, of mean $p = \mathbb{P}(\#\text{roots}(\Delta P) = w+1)$ and variance $p(1-p)/1000$. The later values are used for the error bars ($[p - \sqrt{p(1-p)/1000}, p + \sqrt{p(1-p)/1000}]$).

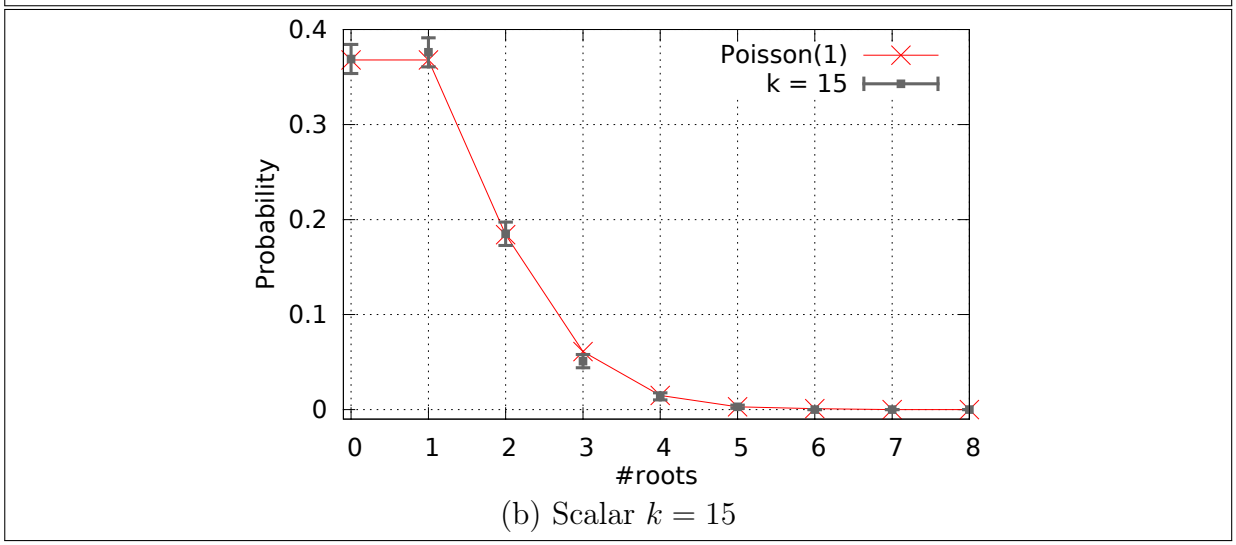
The two graphs in figure 7.2 correspond to two corner-cases:

(a) $k = 3 = (11)_2$: the number of roots is small because the polynomial degree is small (it is 13). (recall that $\#\text{roots}(P)$ cannot exceed the degree of P).

(b) $k = 15 = (1111)_2$: the number of roots is also small, but this times because the result of Leont'ev applies. Indeed, the degree is 7213, thus the polynomial is much more random-looking.



(a) Scalar $k = 3$



(b) Scalar $k = 15$

Figure 7.2: #roots probability for ECSM $[k]G$.

Actually, it is computationally hard to count the roots of polynomials of degree greater than 7213. But it can be checked that the degree of the polynomials is growing exponentially with k. This is represented in figure 7.3, where we see that the degree is about equal to $k^{3.25}$ (of course, when k has a large Hamming weight, as in $(11\ldots1)_2$, the degree is

larger than when k is hollow, as in $(10\ldots0)_2$). In particular, the polynomial ΔP reaches degree 2^{32} (resp. 2^{64}) when k has about 10 (resp. 18) bits. Thus, modulo r (recall equation (7.1)), the polynomial ΔP has maximal degree as long as the fault is injected before the last 10 (resp. 18) elliptic curve operations when r fits on 32 bits (resp. 64 bits).

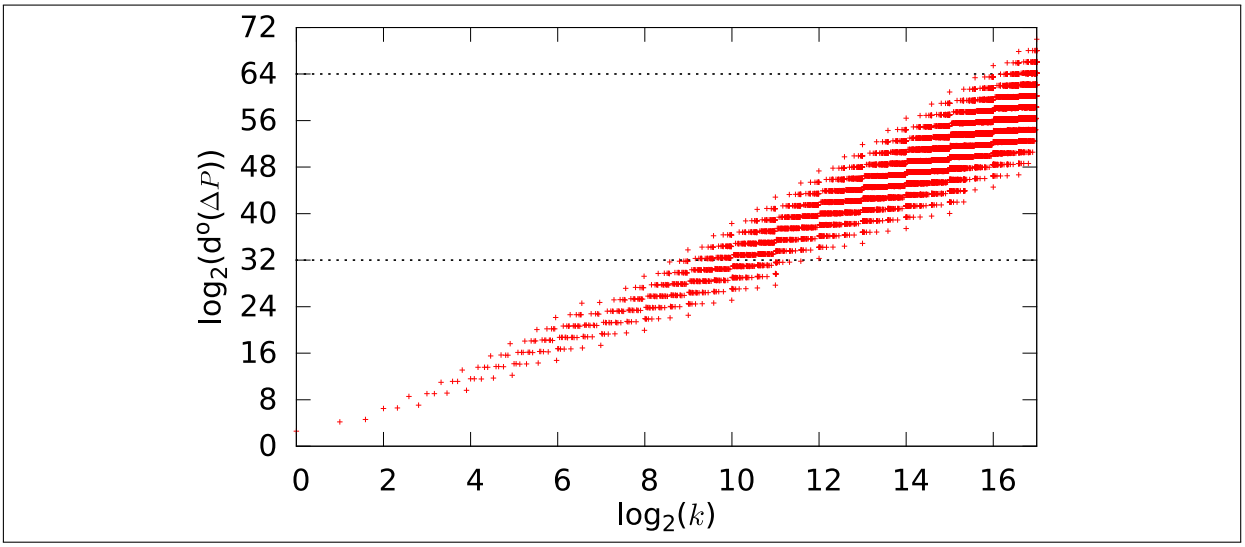


Figure 7.3: Degree of the polynomial ΔP against the value of k (in log-log scale).

7.4 Practical Study

On Edwards curves and twisted Edwards curves, the addition law is *complete*: addition formulas work for all pairs of input points (proposition 2.21-proposition 2.29). In particular, there is no troublesome point at infinity. Another advantage of Edwards curve is the atomicity of the formula doubling and adding and the constant time to protect the classical Side Channel Attack (SCA). The addition law is *unified*, meaning that there is no test to verify if the two input points are equal, opposite or different. To be more efficient, we use the projective coordinates to the addition law named ECADD-complete-unified (see equation (2.25) for Edwards curves - equation (2.29) for twisted Edwards curves).

7.4.1 Scalar Multiplication with the modular extension protection

The ECSM with modular extension protection using complete unified addition formulas is given in algorithm 7.1 for Edwards curves and algorithm 7.2 for twisted Edwards curves. The first phase can compute offline; because find r verifies the lemmas 7.6 and 7.9 is not trivial. The second phase is composed by two ECSM computations online. The first ECSM computation consists in multiplying the point P with the scalar k on the ring $\mathbb{Z}_{pr}$ using the parameters defined later in this section by the proprieties 7.7 or 7.10. The second ECSM computation is the multiplication of the point P' with the scalar k on the *small* curve over $\mathbb{F}_r$ using the parameters defined in the lemmas 7.6 or 7.9. It is worthwhile to note that these two ECSM share the same code (see. algorithm 3.6).

Algorithm 7.1: ECSM with modular extension protection using complete unified addition formulas - Edwards Curves case

Require: $P \in Ed(\mathbb{F}_p)$, $k \in \mathbb{Z}$
Ensure: $Q = [k]P \in Ed(\mathbb{F}_p)$

Offline phase

$\lambda p \leftarrow u_G^2 + v_G^2 - c^2(1 + cu_G^2 v_G^2)$ in $\mathbb{Z}$ ▷ using definition 7.5
repeat
  Choose a random prime $r < p$
  $u'_G \leftarrow u_G \bmod r$
  $v'_G \leftarrow v_G \bmod r$
  $c'^2 \leftarrow c^2 + \lambda p \bmod r$
  $d' \leftarrow \frac{dc^2}{c^2+\lambda p} \bmod r$
until $u'_G \neq 0$ and $v'_G \neq 0$ and $c'd'(1 - c'^4 d') \neq 0$ and c'^2 a square
  ▷ r verifies the lemma 7.6
Determine the small curve $Ed(\mathbb{F}_r)$ with parameters c' and d' , and a point $P'(u'_G, v'_G)$ is on that curve.
Determine the combined curve $Ed(\mathbb{Z}_{pr})$ with parameters $C = CRT(c, c')$ and $D = CRT(d, d')$ ▷ using proposition 7.7

Online phase

$(U_{pr}, V_{pr}, W_{pr}) \leftarrow ECSM(P, k, Ed(\mathbb{Z}_{pr}))$ ▷ without test on the point and on the scalar value (proposition 2.21)
$(U_r, V_r, W_r) \leftarrow ECSM(P', k, Ed(\mathbb{F}_r))$ ▷ without test on the point and on the scalar value (proposition 2.21)
if $(U_{pr} \bmod r, V_{pr} \bmod r, W_{pr} \bmod r) = (U_r, V_r, W_r)$ **then**
  return $normalization(U_{pr} \bmod p, V_{pr} \bmod p, W_{pr} \bmod p)$ ▷ proposition 2.24
else
  return `Error`
end if

Algorithm 7.2: ECSM with modular extension protection using complete unified addition formulas - Twisted Edwards Curves case

Require: $P \in TEd(\mathbb{F}_p)$, $k \in \mathbb{Z}$
Ensure: $Q = [k]P \in TEd(\mathbb{F}_p)$

Offline phase

$\lambda \leftarrow (au_G^2 + v_G^2 - 1 - du_G^2 v_G^2) \div p$ ▷ using definition 7.8
Find all the factor r smaller than p of λ
for each factor r **do**
  $u'_G \leftarrow u_G \bmod r$
  $v'_G \leftarrow v_G \bmod r$
  $a' \leftarrow a \bmod r$
  $d' \leftarrow d \bmod r$
  if $u'_G \neq 0$ and $v'_G \neq 0$ and $a'd'(a' - d') \neq 0$ and a' a square and d' a non-square **then**
    break ▷ r verifies the lemma 7.9
  else
    r does not work
  end if
end for
Determine the small curve $TEd(\mathbb{F}_r)$ with parameters a' and d' , and a point $P'(u'_G, v'_G)$ is on that curve.
Determine the combined curve $TEd(\mathbb{Z}_{pr})$ with parameters $A = a$ and $D = d$ ▷ using proposition 7.10

Online phase

$(U_{pr}, V_{pr}, W_{pr}) \leftarrow ECSM(P, k, TEd(\mathbb{Z}_{pr}))$ ▷ without test on the point and on the scalar value (proposition 2.29)
$(U_r, V_r, W_r) \leftarrow ECSM(P', k, TEd(\mathbb{F}_r))$ ▷ without test on the point and on the scalar value (proposition 2.29)
if $(U_{pr} \bmod r, V_{pr} \bmod r, W_{pr} \bmod r) = (U_r, V_r, W_r)$ **then**
  return *normalization*$(U_{pr} \bmod p, V_{pr} \bmod p, W_{pr} \bmod p)$ ▷ proposition 2.24
else
  return `Error`
end if

7.4.2 Edwards curves

Given an Edwards curve (definition 2.19) over $\mathbb{F}_p$ and a point (u_G, v_G), we define by λ the multiple of p to add when the *point on curve* equation is embedded from $\mathbb{F}_p$ to $\mathbb{Z}$. Formally, we have the following definition and properties.

Definition 7.5 (Parameter λ for Edwards curves) *Given an Edwards elliptic curve of equation (2.19), the parameter λ is the integer satisfying the relationship in $\mathbb{Z}$:*

$$u_G^2 + v_G^2 = c^2(1 + du_G^2v_G^2) + \lambda p. \tag{7.3}$$

Lemma 7.6 *Let p be a prime and an Edwards curve over $\mathbb{F}_p$ as per definition 2.19, characterized by c, d. Let λ as per definition 7.5.*

Let r be a prime number $r < p$, such that $c^2+\lambda p$ is a non-zero square in $\mathbb{F}_r$, $u_G \bmod r \neq 0$ and $v_G \bmod r \neq 0$. The set of points which satisfy $Ed(\mathbb{F}_r) : u^2+v^2 = c'^2(1+d'u^2v^2) \bmod r$ with:

$$\begin{cases} c'^2 = c^2 + \lambda p & \bmod r \\ d' = \dfrac{dc^2}{c^2 + \lambda p} & \bmod r \end{cases} . \tag{7.4}$$

is an Edwards curve, generated by the point $(u'_G, v'_G) = (u_G \bmod r, v_G \bmod r)$.

If the parameters c' and d' satisfy $c'd'(1 - c'^4d') \neq 0$ and d' is not a square in the finite field $\mathbb{F}_r$, then the ECSM computation on this small Edwards curve $Ed(\mathbb{F}_r)$ is complete, i.e., can be computed without point or scalar conditional tests.

Proof Let r a prime number smaller than p. If c, d, p, λ satisfy the conditions given in definitions 2.19 and 7.5, and if $c^2 + \lambda p$ is a non-zero square in $\mathbb{F}_r$, then we have on $\mathbb{Z}$:

$$u_G^2 + v_G^2 = c^2(1 + du_G^2v_G^2) + \lambda p \tag{7.5}$$

$$= (c^2 + \lambda p) + c^2du_G^2v_G^2 \ . \tag{7.6}$$

As by hypothesis $(c^2 + \lambda p) \bmod r \neq 0$, then we have in $\mathbb{F}_r$:

$$u_G^2 + v_G^2 = (c^2 + \lambda p)\left(1 + \frac{c^2d}{c^2 + \lambda p}u_G^2v_G^2\right) \ . \tag{7.7}$$

As in addition $(c^2 + \lambda p)$ is a square in $\mathbb{F}_r$, we have in $\mathbb{F}_r$:

$$u_G^2 + v_G^2 = \left(\sqrt{c^2 + \lambda p}\right)^2\left(1 + \frac{c^2d}{c^2 + \lambda p}u_G^2v_G^2\right) \ . \tag{7.8}$$

By definition 2.19, the parameter $c' = \sqrt{c^2 + \lambda p}$ in $\mathbb{F}_r$ and $d' = \frac{c^2d}{c^2+\lambda p}$ in $\mathbb{F}_r$ with the assumption $c'd'(1 - c'^4d') \neq 0$ are the parameters of the Edwards curve Ed_r and the point $(u'_G, v'_G) = (u_G \bmod r, v_G \bmod r)$ is on the curve $Ed(\mathbb{F}_r)$ by curve construction.

If $u_G \bmod r \neq 0$ and $v_G \bmod r \neq 0$ then the order of the point (u'_G, v'_G) is greater than 4, because the point at infinity and the point of order 2 on $Ed(\mathbb{F}_r)$ have the u-coordinate equal to zero, and the point of order 4 on $Ed(\mathbb{F}_r)$ has the v-coordinate equal to zero.

If d' is not a square in $\mathbb{F}_r$ by contruction of d' value, then the addition law on this *small* curve is unified and complete [BL07]. Thus there is no need for point verification testing when performing additions in an ECSM, and the scalar can be an integer greater than the order of the *small* curve $Ed(\mathbb{F}_r)$. □

For the purpose of the modular extension countermeasure, we extend the notion of Edwards curve to rings. Similar idea can be found in [BOS06, BV07, Joy10](such as $\mathbb{Z}_{pr}$).

Proposition 7.7 *Let an Edwards curve defined on $\mathbb{F}_p$ with the parameters c, d and the point (u_G, v_G). If a random number r verifying the lemma 7.6 can be found to define the Edwards curve $Ed(\mathbb{F}_r)$ with the parameters c', d', then $C = CRT(c, c')$ and $D = CRT(d, d')$ are the parameters of an Edwards elliptic curve over the rings $\mathbb{Z}_{pr}$. This curve parameters permits to detect a fault thanks to the comparison at line 13 in the algorithm 7.1.*

Proof We introduce the following notations:

- We denote by Pt_p with p in index a point named Pt computed on the $Ed(\mathbb{F}_p)$;
- We denote by Pt_r with r in index a point named Pt computed on the $Ed(\mathbb{F}_r)$;
- We denote by Pt_{pr} with pr in index a point named Pt computed on the $Ed(\mathbb{Z}_{pr})$.

The input value of the two ECSMs verify the equality using the projective coordinates, because we have as input (u_G, v_G) for the combined curve and (u'_G, v'_G) for the *small* curve:

$$\begin{cases} U\text{- coordinate:} & u'_G & = & u_G & \mod r, \\ V\text{- coordinate:} & v'_G & = & v_G & \mod r, \\ W\text{- coordinate:} & 1 & = & 1 & \mod r. \end{cases} \tag{7.9}$$

The ECSM computation over the combined curve on the ring extension $\mathbb{Z}_{pr}$ and the *small* curve over finite field $\mathbb{F}_r$ do consist in the same sequence of addition operations (ECADD-complete-unified).

Let P_{pr} and P_r be two points such that $U_{P_r} = U_{P_{pr}} \bmod r, V_{P_r} = V_{P_{pr}} \bmod r, W_{P_r} = W_{P_{pr}} \bmod r$. Let Q_{pr} and Q_r be two points such that $U_{Q_r} = U_{Q_{pr}} \bmod r, V_{Q_r} = V_{Q_{pr}} \bmod r, W_{Q_r} = W_{Q_{pr}} \bmod r$.

We compute R_r the result of ECADD-complete-unified between P_r and Q_r over $Ed(\mathbb{F}_r)$, and R_{pr} the result of ECADD-complete-unified between P_{pr} and Q_{pr} over $Ed(\mathbb{Z}_{pr})$.

The computation of the projective coordinates of R_{pr} is composed by addition, subtraction, multiplication over the ring $\mathbb{Z}_{pr}$ using the projective coordinates of P_{pr} and Q_{pr} and the two curve parameters C and D.

The computation of the projective coordinates of R_{pr} is composed by addition, subtraction, multiplication over the ring $\mathbb{Z}_{pr}$ using the projective coordinates of P_r and Q_r and the two curve parameters c' and d'.

By construction $C = CRT(c, c')$ and $D = CRT(d, d')$, we have $C \bmod r = c'$ and $D \bmod r = d'$, so the projective coordinates of R_{pr} are pairwise equal modulo r with the projective coordinates of R_r.

As the ECADD-complete-unified operation conserves the equality of the point coordinates value modulo r, we conclude that the ECSM computation conserves the equality of the point coordinates value modulo r between the computation over the ring extension and over the finite field $\mathbb{F}_r$. □

7.4.3 Twisted Edwards curves

Like an Edwards curve, given a twisted Edwards curve (definition 2.27) over $\mathbb{F}_p$ and a point (u_G, v_G), we define by λ the multiple of p to add when the *point on curve* equation is plunged from $\mathbb{F}_p$ to $\mathbb{Z}$. Formally,

Definition 7.8 (Parameter λ for twisted Edwards curves) *Given a twisted Edwards elliptic curve of equation (2.26), the parameter λ is the integer satisfying the following relationship in $\mathbb{Z}$:*

$$au_G^2 + v_G^2 = 1 + du_G^2 v_G^2 + \lambda p \ . \tag{7.10}$$

Lemma 7.9 *If a, d, p, λ verify the conditions defined in definition 7.8, then if we can choose a prime factor r of λp such that $u_G \bmod r \neq 0$ and such that the point $(u'_G, v'_G) = (u_G \bmod r, v_G \bmod r)$ generates the curve $TEd(\mathbb{F}_r) : a'u^2 + v^2 = 1 + d'u^2v^2$ where $a' = a \bmod r$ and $d' = d \bmod r$.*

If the parameters a' and d' satisfy $a'd'(a'-d') \neq 0$, a' is a square and d' is a non-square in the finite field $\mathbb{F}_r$, then the ECSM computation on this small *twisted Edwards curve $TEd(\mathbb{F}_r)$ requires no point and scalar tests.*

Proof If a, d, p, λ satisfy the conditions given in definition 7.8, then we have we have on $\mathbb{Z}$:

$$au_G^2 + v_G^2 = 1 + du_G^2 v_G^2 + \lambda p \ . \tag{7.11}$$

Let r a prime factor of λp, then $\lambda p = 0 \bmod r$. Hence we have:

$$\begin{aligned} &(a \bmod r)(u_G \bmod r)^2 + (v_G \bmod r)^2 \\ &= 1 + (d \bmod r)(u_G \bmod r)^2 (v_G \bmod r)^2 \ . \end{aligned} \tag{7.12}$$

By definition 2.27, the parameters $(a', d') = (a \bmod r, d \bmod r)$ with the assumption $a'd'(a' - d') \neq 0$ are the parameters of the twisted Edwards curve TEd_r and the point $(u'_G, v'_G) = (u_G \bmod r, v_G \bmod r)$ is on the curve TEd_r by curve construction.

By definition 2.27, the parameters $(a', d') = (a \bmod r, d \bmod r)$ with the assumption $a'd'(a' - d') \neq 0$ are the parameters of the twisted Edwards curve $TEd(\mathbb{F}_r)$ and the point $(u'_G, v'_G) = (u_G \bmod r, v_G \bmod r)$ is on the curve $TEd(\mathbb{F}_r)$ by curve construction.

If a' is a square and d' is a non-square in $\mathbb{F}_r$, then the addition law on this *small* curve is unified and complete [BL07], which implies that no point verification testing is required for each addition in ECSM, and that the scalar can be an arbitrary integer, for instance greater than the order of the *small* curve. □

For the purpose of the modular extension countermeasure depicted in algorithm 7.2, we extend the notion of twisted Edwards curve to rings (such as $\mathbb{Z}_{pr}$).

Proposition 7.10 *Let a twisted Edwards curve defined on $\mathbb{F}_p$ with the parameters a, d and the point (u_G, v_G). If a random number r verifying the lemma 7.9 can be found to define the twisted Edwards curve $TEd(\mathbb{F}_r)$ with the parameters a', d', then $A = a$ and $D = d$ are the parameters of a twisted Edwards curve over the ring $\mathbb{Z}_{pr}$.*

If $u_G \bmod r \neq 0$ then the point (u'_G, v'_G) is not the point at infinity. So, this point (u'_G, v'_G) is a generator of a non-trivial subgroup of the elliptic curve $TEd(\mathbb{F}_r)$.

This curve parameters permit to detect a fault with the comparison at line 18 in the algorithm 7.2.

Proof The input value of the two ECSM verify the equality using the projective coordinates, because we have as input (u_G, v_G) for the combined curve and (u'_G, v'_G) for the *small* curve, as described previously in equation (7.9).

The ECSM computation over the combined curve on the ring extension $\mathbb{Z}_{pr}$ and the *small* curve over finite field $\mathbb{F}_r$ consist in the same sequence of addition operations (ECADD-complete-unified). Namely, the sequence is given in algorithm 3.6, where TEd is either $TEd(\mathbb{Z}_{pr})$ or $TEd(\mathbb{F}_r)$.

By construction $A = a, D = d$ and we have $a \bmod r = a'$ and $d \bmod r = d'$, so the projective coordinates of R_{pr} are equal modulo r two by two with the projective coordinates of R_r.

As the ECADD-complete-unified operation conserves the equality of the point coordinates value modulo r, we conclude that the ECSM computation conserves the equality of the point coordinates value modulo r between the computation over the ring extension and over the finite field $\mathbb{F}_r$. □

7.4.4 Discussion

About small curve requirements Both for Edwards and twisted Edwards curves, the small curve is of course not a cryptographic-grade curve. Indeed, the modulus r is too small and the curve might have points of low order. However, the small curve is not intended to be the support of a secure cryptographic operation: the computation on this curve actually remains internal to fault-detection-enabled ECSM. That is, the small curve is intended here to carry out exactly the same computation as that done in the curve on the extended ring, in order to enable the integrity verification.

Resistance to some attacks As a general guideline, additional protection against the *common point* attack [Bat14] shall be enforced. This attack is based on curve parameters alteration, with the hope that the obtained curve is weak. Thus, to thwart this attack, the curve parameters shall be tested before and after the computation.

7.5 Performance

Our implementation uses the projective coordinates described in section 2.3.3 or [BL07, section. 4, page 9]. Projective unified addition version takes $10M_p + 1S_p + 1C_p^{c/a} + 1C_p^d + 7A_p$ where M_p is the cost of multiplication, S_p is the cost of square, $C_p^{c/a}$ is the cost of multiplying by c for Edwards curve and by a for twisted Edwards curves, C_p^d is the cost of multiplying by d, and A_p abbreviates addition. The ECSM is the algorithm *add-always left-to-right* like described in algorithm 3.6. The bit-width of the modulus is denoted by N (e.g., $N = 256$ for `Ed25519`). We denote by n' the number of words of the modulus, that is $n = 256/32 = 8$ on 32-bit platforms (or $n = 256/16 = 16$ on 16-bit platforms). We consider that cost of a multiplication of two numbers composed by n words is n^2, cost of a square S_p is $0.8M_p$ and the addition A_p is n. The table 7.1 permits to compare the time of each ECADD-complete-unified, depending of the number of words n.

Curves type		Edwards curve	Twisted Edwards curve
ECADD-complete-unified on $\mathbb{F}_p$		$\mathbf{11.8n^2 + 7n}$	$\mathbf{11.8n^2 + 7n}$
ECADD-complete-unified on $\mathbb{Z}_{pr}$		$11.8n^2 + 30.6n + 18.8$	$12.8n^2 + 32.6n + 29.8$
ECADD-complete-unified on $\mathbb{F}_r$		19.8	19.8
Total cost of the countermeasure		$\mathbf{11.8n^2 + 30.6n + 38.6}$	$\mathbf{12.8n^2 + 32.6n + 49.6}$
Computational overhead with	$n = 8$	$\simeq +28\%$	$\simeq +39\%$
	$n = 16$	$\simeq +13\%$	$\simeq +21\%$

Table 7.1: Theory of the elliptic curve addition cost for Edwards and twisted Edwards curves

7.5.1 Edwards curve example

For our experiment, we generate a Edwards curve on the finite field $\mathbb{F}_{2^{255}-19}$ defined by $u^2 + v^2 = 1 - 6u^2v^2 \bmod 2^{255} - 19$.

Using the proposition [MS10, sec 3.1], this Edwards curve corresponds to an elliptic curve defined by $\mathcal{W}(\mathbb{F}_{2^{255}-19}) : y^2 = x^3 + a_2x^2 + a_4x$ on $\mathbb{F}_{2^{255}-19}$, with $a_2 = -5$ and $a_4 = 49$. The number of elements defined on the curve computed by MAGMA tool [Uni] is:

$$\#Ed(2^{255} - 19) = 2^{255} + 13869417260526501318107114900338184066 0. \tag{7.13}$$

We find a generator point (u_G, v_G) on the Edwards curve with:

$$\begin{cases} u_G &= 53746514586250388770967951861766021561817370662802863797712166095360241234126, \\ v_G &= 19570081233560550597987439135529516381390903225319934175948181057081969418594. \end{cases} \tag{7.14}$$

The co-factor of the curve is 4. For the small curve, we can choose $r = 2147499037$; hence we have $c' = 1800340494, d' = 1430405543, u'_G = 28751952$ and $v'_G = 1290929995$. These parameters verify the lemma 7.6. The probability of fault non-detection is about equal to 2^{-31}.

Remark To generate 500.000 random primes $r < 2^{32}$ verifying the lemma 7.6, using online version on MAGMA tool [Uni], the time is 110.769 seconds. The number of random prime number generated is 1.999.238. The probability that a random prime r meets the requirement of lemma 7.6 is less than 1/4 verified by this experimental part.

7.5.2 Twisted Edwards curve example: `Curve25519` / `Ed25519`

On the finite field $\mathbb{F}_{2^{255}-19}$, the elliptic curve `Curve25519` defined by the equation $y^2 = x^3 + 48662x^2 + x$ is bi-rationally equivalent to the twisted Edwards Curves `Ed25519` defined by equation $-u^2 + v^2 = 1 - \frac{121665}{121666}u^2v^2$ as detailed in example in section 2.3.3.

We find a generator point (u_G, v_G) on the twisted Edwards curve `Ed25519` with:

$$\begin{cases} u_G &= 24727413235106541002554574571675588834622768167397638456726423682521233608 2063, \\ v_G &= 15549675580280190176352668710449542251549572066445060580507079593062643049417. \end{cases} \tag{7.15}$$

The prime factors of λ (recall definition 7.8) smaller, than p are stored in the table 7.2. Actually, there is in λ only one factor larger than p, of length ≈ 900 bits, hence of no practical use—it is indeed more efficient to perform the computation several times or to verify the signature.

Prime factors r	2	3	17	47	78857	843229	159962189299
Length in bit of r	2	2	5	7	16	19	40
r verifies the lemma 7.9	False	False	False	False	**True**	**True**	False

Table 7.2: Prime Factors $< p$ of λ for the generator point (u_G, v_G) given in example (curve `Ed25519` defined in section 7.5.2)

For the small curve like described in table 7.2, we can choose:

1. $r = 78857, a' = 32865, d' = 47471, u'_G = 71670$ and $v'_G = 16752$, or
2. $r = 843229, a' = 839079, d' = 43998, u'_G = 96826$ and $v'_G = 488894$.

These parameters verify the lemma 7.9.

The probability of fault non-detection is about equal to 2^{-16} for the first case and to 2^{-19} for the second case.

Important remark we notice that the small verification field $\mathbb{F}_r$ cannot be chosen at random. Instead, the value of r is highly constrained, as shown in table 7.2. This limitation of the ring extension countermeasure was not previously known.

7.5.3 Comments about results

One can see in table 7.1 that the global time computation increases by 28% or 39% for each addition operation using a 256-bit curve with a 32-bit processor ($n = 8$). The computation overhead decreases when the curve parameters and the security increase. Remarkably, the implementation code is the same for the two ECSM computations. The memory storage requirement is increased by two word registers for each variable.

7.6 Conclusions

It is well known that detecting faults while computing elliptic curve cryptography can be achieved thanks to ring extension. In this paradigm, two entangled computations are carried out in the extended ring, allowing to tightly produce the functional result along with a redundant one, which can be checked independently. However, classical methods fail because the redundant computation evaluated standalone or entangled can be different, owing to some tests being independently evaluated when the elliptic curve formula are not complete. Edwards curves and twisted Edwards curves have complete formula, hence are not concerned with the issue of consistent tests requirement. Still, the application of ring extension involves some technicalities, we discuss in the chapter. Namely, Edwards curves require an adaptation with Chinese reminder theorem of the curve constant parameter. As for twisted Edwards curves, the modulus extension can

only be performed with a factor of λ, which is related both to the curve parameters and to the base point. The outcome is a provable fault detection method for (twisted) Edwards curves, which despite its simplicity, is novel, elegant and effective. This countermeasure should be more effective, because the computation of the small curve is not required.

Chapter 8

Conclusion

Contents

8.1 Conclusion

Side channel attacks have challenged the security level of cryptographic algorithms on embedded devices. Since the 1990s, the developer and designer want to protect against these kinds of attacks. What we have learned during these three years of researches is that protections on large operations are necessary, but not sufficient to protect against further attacks.

Nowadays, we are never immune to sophisticated new attacks like Template attacks. Our works are very concrete, the two attacks presented can be realized in real world. The experimental part of the attacks is made on real acquisition and no only in simulation phase. In our attacks what is specific is that we manage to forge collisions on small operations: propagation of inner carry or extra-reductions. Small operations seem innocuous among millions of others, but they allow us to reveal information that defeats several classical protections. The constant time or the regular implementation helps an attacker to recognize the position of each operation. This result is well known, but in protecting against some attacks, we illustrate once again that protection leads to new attacks.

Another finding of this thesis is that the adaptation of all countermeasures applied on RSA is not necessarily trivial on ECC. This adaptation does not have the same proof of security. In fact, the security level of each countermeasures must be studied for another cryptosystem.

8.2 Perspectives

RSA and ECC are standard industrial cryptography widely deployed. In the near future, other asymmetric algorithms will be standardized hence the side channel and fault attacks stall thus also by studying on these algorithms. The attacks will be adapted against these

new cryptosystems. In order to enhance security against fault and side channel attacks, new protections will have to be implemented and evaluated in the light of the new attacks.

Appendix A

Number theory

In 1640, Pierre de Fermat wrote the theorem wording in a letter, without demonstration. In 1741, Euler proved this theorem [Eul41].

Theorem A.1 (Little Fermat's theorem) *If p is a prime number, then for any integer m we have $m^p = m \mod p$.*

A variant of the Little Fermat's theorem was defined and proved by Euler [Eul63].

Theorem A.2 (Euler's theorem) *Let $\phi(n)$ denote the number of integers in $[1, n]$ co-prime to n. Let m be an integer co-prime to n. We have $a^{\phi(n)} = 1 \mod n$.*

In RSA, $n = p \times q$ with p and q large prime numbers, so we have $\phi(n) = (p-1)(q-1)$.

The Chinese Remainder Theorem was discovered in third century by the Chinese mathematician Sunzi in Sunzi Suanjing.

Theorem A.3 (Chinese Remainder theorem) *Let n_1 and n_2 be co-prime integers. Let m_1 and m_2 be integers. The following system*

$$\begin{cases} x = m_1 \mod n_1 \\ x = m_2 \mod n_2 \end{cases} . \tag{A.1}$$

has an unique solution modulo $n_1 n_2$ which is $a_1 q_1 n_2 + a_2 q_2 n_1$ where q_1 is the inverse of n_1 modulo n_2 and q_2 is the inverse of n_2 modulo n_1.

The RSA-CRT was more efficient compared to RSA-SFM, but required to store private values p and q. The steps of RSA-CRT are:

1. Compute $d_p = d \mod p$ and $c_p = m^{d_p} \mod p$.
2. Compute $d_q = d \mod q$ and $c_q = m^{d_q} \mod q$.
3. Compute $q_{inv} = q^{-1} \mod p$.
4. Recombine c_p and c_q using $q_i nv$ by this following formula :

$$c = (((c_p - c_q) \times q_{inv}) \mod p) * q + c_q \ . \tag{A.2}$$

Remark In private modular exponentiation, we store the values d_p, d_q and q_{inv} to avoid the modular inversion of q by p.

Theorem A.4 (Bézout's identity) *Let a and b be non zero integers and let δ be their greatest common divisor. Then there exist integers u and v such that:*

$$au + bv = \delta \ . \tag{A.3}$$

Extended Euclide algorithm allows to compute the greatest common divisor δ, and the two integers u and v become from (A.3).

Algorithm A.1: Extended Euclide algorithm

Require: a and b non zero integers
Ensure: δ, u, v such as $au + bv = d$

$r_b \leftarrow b$
$r_a \leftarrow a$
$u_1 \leftarrow 1$
$v_1 \leftarrow 0$
$u_2 \leftarrow 0$
$v_2 \leftarrow 1$
while $r_b \neq 0$ **do**
  $q \leftarrow r_a \div r_b$
  $(r_a, r_b) \leftarrow (r_b, r_a \mod r_b)$
  $(u_2, u_1) \leftarrow ((u_1 - (q \times u_2)), u_2)$
  $(v_2, v_1) \leftarrow ((v_1 - (q \times v_2)), v_2)$
end while
$(\delta, u, v) \leftarrow (r_a, u_1, v_1)$
return (δ, u, v)

Remark The modular inversion computation is based on the extended Euclide algorithm. Compute n and e with private keys d, p, q is easier using algorithm A.1. In fact e is $u \mod (p-1)(q-1)$ with u the output of algorithm A.1 with input values $a = d$ and $b = n$. For RSA-CRT, the inversion of q modulo p can be retrieve using algorithm A.1 with input values $a = q$ and $b = p$. The modular inversion equals $u \mod p$.

Appendix B

Parameters of elliptic curves

B.1 NIST curves - P-256

The elliptic curve domain of the American curves standardized by NIST [NIS13] P-256 is the following parameters set (p, a, b, x, y, n, h) :

$p =$ 0xFFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF
$a =$ 0xFFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFC
$b =$ 0x5AC635D8AA3A93E7B3EBBD55769886BC651D06B0CC53B0F63BCE3C3E27D2604B
$x =$ 0x6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A13945D898C296
$y =$ 0x4FE342E2FE1A7F9B8EE7EB4A7C0F9E162BCE33576B315ECECBB6406837BF51F5
$n =$ 0xFFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551
$h =$ 0x1 .

B.2 BSI curves - brainpoolP256r1

The elliptic curve domain of the German curves standardized by BSI [BSI10] brainpoolP256r1 is the following parameters set (p, a, b, x, y, n, h) :

$p =$ 0xA9FB57DBA1EEA9BC3E660A909D838D726E3BF623D52620282013481D1F6E5377
$a =$ 0x7D5A0975FC2C3057EEF67530417AFFE7FB8055C126DC5C6CE94A4B44F330B5D9
$b =$ 0x26DC5C6CE94A4B44F330B5D9BBD77CBF958416295CF7E1CE6BCCDC18FF8C07B6
$x =$ 0x8BD2AEB9CB7E57CB2C4B482FFC81B7AFB9DE27E1E3BD23C23A4453BD9ACE3262
$y =$ 0x547EF835C3DAC4FD97F8461A14611DC9C27745132DED8E545C1D54C72F046997
$n =$ 0xA9FB57DBA1EEA9BC3E660A909D838D718C397AA3B561A6F7901E0E82974856A7
$h =$ 0x1 .

Appendix C

Implementation of elliptic curves in cryptography library

In our experiment in chapter 4, the cryptography library is PolarSSL v1.3.7. The doubling implementation and the mixed addition implementation are in the following algorithms.

Algorithm C.1: Doubling in PolarSSL v1.3.7

Require: (X, Y, Z) Jacobian coordinates of a point
Ensure: (X_R, Y_R, Z_R) Jacobian coordinates corresponding to the doubling of the input point

1: $T_3 \leftarrow X \times X \mod p$
2: $T_2 \leftarrow Y \times Y \mod p$
3: $Y_3 \leftarrow T_2 \times T_2 \mod p$
4: $X_3 \leftarrow X + T_2 \mod p$
5: $X_3 \leftarrow X_3 \times X_3 \mod p$
6: $X_3 \leftarrow X_3 - Y_3 \mod p$
7: $X_3 \leftarrow X_3 - T_3 \mod p$
8: $T_1 \leftarrow X_3 \times 2 \mod p$
9: $Z_3 \leftarrow Z \times Z \mod p$
10: $X_3 \leftarrow Z_3 \times Z_3 \mod p$
11: $T_3 \leftarrow T_3 \times 3 \mod p$
12: $X_3 \leftarrow X_3 \times a \mod p$
13: $T_3 \leftarrow T_3 + X_3 \mod p$
14: $X_3 \leftarrow T_3 \times T_3 \mod p$
15: $X_3 \leftarrow X_3 - T_1 \mod p$
16: $X_3 \leftarrow X_3 - T_1 \mod p$
17: $T_1 \leftarrow T_1 - X_3 \mod p$
18: $T_1 \leftarrow T_1 \times T_3 \mod p$
19: $T_3 \leftarrow Y_3 \times 8 \mod p$
20: $Y_3 \leftarrow T_1 - T_3 \mod p$
21: $T_1 \leftarrow Y + Z \mod p$
22: $T_1 \leftarrow T_1 \times T_1 \mod p$
23: $T_1 \leftarrow T_1 - T_2 \mod p$
24: $Z_3 \leftarrow T_1 - Z_3 \mod p$
25: $X_R \leftarrow X_3$
26: $Y_R \leftarrow Y_3$
27: $Z_R \leftarrow Z_3$

Algorithm C.2: Mixed-add in PolarSSL v1.3.7

Require: (X, Y, Z) Jacobian coordinates of one point, (x, y) affine coordinates of the second point

Ensure: (X_R, Y_R, Z_R) Jacobian coordinates corresponding to the addition result

1: $T_1 \leftarrow Z \times Z \bmod p$
2: $T_2 \leftarrow T_1 \times Z \bmod p$
3: $T_1 \leftarrow T_1 \times x \bmod p$
4: $T_2 \leftarrow T_2 \times y \bmod p$
5: $T_1 \leftarrow T_1 - X \bmod p$
6: $T_2 \leftarrow T_2 - Y \bmod p$
7: **if** $T_1 = 0$ **then**
8: **if** $T_3 = 0$ **then**
9: $R \leftarrow ECDBL(P)$ ▷algorithm C.1
10: **else**
11: $R \leftarrow \mathcal{O}_{\mathcal{E}}$
12: **end if**
13: **end if**
14: $Z_3 \leftarrow Z \times T_1 \bmod p$
15: $T_3 \leftarrow T_1 \times T_1 \bmod p$
16: $T_4 \leftarrow T_3 \times T_1 \bmod p$
17: $T_3 \leftarrow T_3 \times X \bmod p$
18: $T_1 \leftarrow 2 \times T_3 \bmod p$
19: $X_3 \leftarrow T_2 \times T_2 \bmod p$
20: $X_3 \leftarrow X_3 - T_1 \bmod p$
21: $X_3 \leftarrow X_3 - T_4 \bmod p$
22: $T_3 \leftarrow T_3 - X_3 \bmod p$
23: $T_3 \leftarrow T_3 \times T_2 \bmod p$
24: $T_4 \leftarrow T_4 \times Y \bmod p$
25: $Y_3 \leftarrow T_3 - T_4 \bmod p$
26: $X_R \leftarrow X_3$
27: $Y_R \leftarrow Y_3$
28: $Z_R \leftarrow Z_3$

Appendix D

Probability of extra-reductions in consecutive operations for u bit values

This chapter describes the probability law of extra-reductions in consecutive operations computed by the lemma 6.3.

D.1 u=2

Table D.2 and table D.1 contain all probabilities for $u = 2$.

		$(x_{M_{i-1}}, x_{S_{i-1}})$			
		(0,0)	(0,1)	(1,0)	(1,1)
(x_{M_i}, x_{S_i})	(0,0)	$1 - \frac{7}{6}\frac{p}{R} + \frac{1}{3}\left(\frac{p}{R}\right)^2 + \frac{7}{90}\left(\frac{p}{R}\right)^3 + \frac{17}{504}\left(\frac{p}{R}\right)^4 - \frac{11}{336}\left(\frac{p}{R}\right)^5 - \frac{1}{72}\left(\frac{p}{R}\right)^6 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{3}\frac{p}{R} - \frac{5}{24}\left(\frac{p}{R}\right)^2 - \frac{17}{504}\left(\frac{p}{R}\right)^4 + \frac{1}{48}\left(\frac{p}{R}\right)^5 + \frac{1}{72}\left(\frac{p}{R}\right)^6 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{4}\frac{p}{R} - \frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{7}{90}\left(\frac{p}{R}\right)^3 + \frac{1}{72}\left(\frac{p}{R}\right)^4 + \frac{1}{84}\left(\frac{p}{R}\right)^5 + \frac{1}{72}\left(\frac{p}{R}\right)^6 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{1}{72}\left(\frac{p}{R}\right)^4 - \frac{1}{72}\left(\frac{p}{R}\right)^6 + \frac{1}{264}\left(\frac{p}{R}\right)^8$
	(0,1)	$\frac{1}{3}\frac{p}{R} - \frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{1}{20}\left(\frac{p}{R}\right)^3 - \frac{1}{21}\left(\frac{p}{R}\right)^4 + \frac{11}{336}\left(\frac{p}{R}\right)^5 + \frac{1}{72}\left(\frac{p}{R}\right)^6 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{21}\left(\frac{p}{R}\right)^4 - \frac{1}{48}\left(\frac{p}{R}\right)^5 - \frac{1}{72}\left(\frac{p}{R}\right)^6 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{20}\left(\frac{p}{R}\right)^3 - \frac{1}{84}\left(\frac{p}{R}\right)^5 - \frac{1}{72}\left(\frac{p}{R}\right)^6 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{72}\left(\frac{p}{R}\right)^6 - \frac{1}{264}\left(\frac{p}{R}\right)^8$
	(1,0)	$\frac{1}{4}\frac{p}{R} - \frac{5}{24}\left(\frac{p}{R}\right)^2 - \frac{1}{36}\left(\frac{p}{R}\right)^3 + \frac{1}{72}\left(\frac{p}{R}\right)^4 + \frac{11}{336}\left(\frac{p}{R}\right)^5 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{12}\left(\frac{p}{R}\right)^2 - \frac{1}{72}\left(\frac{p}{R}\right)^4 - \frac{1}{48}\left(\frac{p}{R}\right)^5 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{36}\left(\frac{p}{R}\right)^3 - \frac{1}{72}\left(\frac{p}{R}\right)^4 - \frac{1}{84}\left(\frac{p}{R}\right)^5 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{72}\left(\frac{p}{R}\right)^4 - \frac{1}{264}\left(\frac{p}{R}\right)^8$
	(1,1)	$\frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{11}{336}\left(\frac{p}{R}\right)^5 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{48}\left(\frac{p}{R}\right)^5 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{84}\left(\frac{p}{R}\right)^5 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{264}\left(\frac{p}{R}\right)^8$

Table D.1: $\mathbb{P}_\theta(X_{M_i} = x_{M_i}, X_{S_i} = x_{S_i}, X_{M_{i-1}} = x_{M_{i-1}}, X_{S_{i-1}} = x_{S_{i-1}})$ for $\vec{\theta} \in \{(0,0),(1,1)\}$ (corresponds to the case $k_i = k_{i-1}$)

(x_{M_i}, x_{S_i}) \ $(x_{M_{i-1}}, x_{S_{i-1}})$	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	$1 - \frac{7}{6}\frac{p}{R} + \frac{13}{36}\left(\frac{p}{R}\right)^2 + \frac{7}{90}\left(\frac{p}{R}\right)^3 - \frac{1}{240}\left(\frac{p}{R}\right)^4 - \frac{13}{504}\left(\frac{p}{R}\right)^5 - \frac{1}{200}\left(\frac{p}{R}\right)^6 + \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{3}\frac{p}{R} - \frac{17}{72}\left(\frac{p}{R}\right)^2 + \frac{1}{240}\left(\frac{p}{R}\right)^4 + \frac{1}{72}\left(\frac{p}{R}\right)^5 + \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{4}\frac{p}{R} - \frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{7}{90}\left(\frac{p}{R}\right)^3 + \frac{1}{40}\left(\frac{p}{R}\right)^4 + \frac{1}{84}\left(\frac{p}{R}\right)^5 + \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{1}{40}\left(\frac{p}{R}\right)^4 - \frac{1}{200}\left(\frac{p}{R}\right)^6 + \frac{1}{360}\left(\frac{p}{R}\right)^8$
(0,1)	$\frac{1}{3}\frac{p}{R} - \frac{17}{72}\left(\frac{p}{R}\right)^2 - \frac{1}{20}\left(\frac{p}{R}\right)^3 + \frac{1}{40}\left(\frac{p}{R}\right)^4 + \frac{13}{504}\left(\frac{p}{R}\right)^5 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{9}\left(\frac{p}{R}\right)^2 - \frac{1}{40}\left(\frac{p}{R}\right)^4 - \frac{1}{72}\left(\frac{p}{R}\right)^5 + \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{20}\left(\frac{p}{R}\right)^3 - \frac{1}{40}\left(\frac{p}{R}\right)^4 - \frac{1}{84}\left(\frac{p}{R}\right)^5 + \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{40}\left(\frac{p}{R}\right)^4 - \frac{1}{360}\left(\frac{p}{R}\right)^8$
(1,0)	$\frac{1}{4}\frac{p}{R} - \frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{1}{36}\left(\frac{p}{R}\right)^3 - \frac{1}{48}\left(\frac{p}{R}\right)^4 + \frac{13}{504}\left(\frac{p}{R}\right)^5 + \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{48}\left(\frac{p}{R}\right)^4 - \frac{1}{72}\left(\frac{p}{R}\right)^5 - \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{36}\left(\frac{p}{R}\right)^3 - \frac{1}{84}\left(\frac{p}{R}\right)^5 - \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$
(1,1)	$\frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{13}{504}\left(\frac{p}{R}\right)^5 + \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{72}\left(\frac{p}{R}\right)^5 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{84}\left(\frac{p}{R}\right)^5 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{360}\left(\frac{p}{R}\right)^8$

Table D.2: $\mathbb{P}_\theta(X_{M_i} = x_{M_i}, X_{S_i} = x_{S_i}, X_{M_{i-1}} = x_{M_{i-1}}, X_{S_{i-1}} = x_{S_{i-1}})$ for $\vec{\theta} \in \{(0,1),(1,0)\}$ (corresponds to the case $k_i \neq k_{i-1}$)

D.2 u=3

Table D.3, table D.4, table D.5 and table D.6 contain all probabilities for $u = 3$.

$(x_{M_i}, x_{S_i}, x_{M_{i-1}}, x_{S_{i-1}})$ \ $(x_{M_{i-2}}, x_{S_{i-2}})$	(0,0)	(0,1)	(1,0)	(1,1)
(0,0,0,0)	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{4003}{27518400}\frac{p}{R}^{15} + \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{1280}\frac{p}{R}^{12} - \frac{19}{14784}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{576}\frac{p}{R}^{9} + \frac{2669}{388080}\frac{p}{R}^{8} + \frac{1409}{120960}\frac{p}{R}^{7} - \frac{33}{1600}\frac{p}{R}^{6} - \frac{25}{336}\frac{p}{R}^{5} + \frac{157}{60480}\frac{p}{R}^{4} + \frac{7}{720}\frac{p}{R}^{3} + \frac{131}{144}\frac{p}{R}^{2} - \frac{7}{4}\frac{p}{R} + 1$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{11}{280800}\frac{p}{R}^{15} - \frac{1}{3888}\frac{p}{R}^{13} - \frac{1}{1680}\frac{p}{R}^{12} + \frac{19}{14784}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{264}\frac{p}{R}^{8} - \frac{1409}{120960}\frac{p}{R}^{7} + \frac{5}{504}\frac{p}{R}^{6} + \frac{37}{1008}\frac{p}{R}^{5} - \frac{1417}{60480}\frac{p}{R}^{4} + \frac{11}{96}\frac{p}{R}^{3} - \frac{29}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{8640}\frac{p}{R}^{16} - \frac{23}{705600}\frac{p}{R}^{15} - \frac{1}{5376}\frac{p}{R}^{12} + \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{576}\frac{p}{R}^{9} - \frac{1591}{388080}\frac{p}{R}^{8} - \frac{181}{24192}\frac{p}{R}^{7} + \frac{79}{4032}\frac{p}{R}^{6} + \frac{1}{84}\frac{p}{R}^{5} + \frac{43}{2880}\frac{p}{R}^{4} + \frac{19}{720}\frac{p}{R}^{3} - \frac{13}{48}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{8640}\frac{p}{R}^{16} + \frac{1}{7200}\frac{p}{R}^{15} - \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{264}\frac{p}{R}^{8} + \frac{181}{24192}\frac{p}{R}^{7} - \frac{1}{72}\frac{p}{R}^{6} + \frac{1}{576}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{8}\frac{p}{R}^{2}$
(0,0,0,1)	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{8640}\frac{p}{R}^{16} - \frac{4003}{27518400}\frac{p}{R}^{15} - \frac{1}{3888}\frac{p}{R}^{13} - \frac{1}{1280}\frac{p}{R}^{12} + \frac{19}{14784}\frac{p}{R}^{11} - \frac{1}{960}\frac{p}{R}^{9} - \frac{2669}{388080}\frac{p}{R}^{8} - \frac{19}{2160}\frac{p}{R}^{7} + \frac{17}{900}\frac{p}{R}^{6} + \frac{1}{16}\frac{p}{R}^{5} - \frac{3}{112}\frac{p}{R}^{4} - \frac{1}{20}\frac{p}{R}^{3} - \frac{17}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{11}{280800}\frac{p}{R}^{15} + \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{1680}\frac{p}{R}^{12} - \frac{19}{14784}\frac{p}{R}^{11} + \frac{1}{264}\frac{p}{R}^{8} + \frac{19}{2160}\frac{p}{R}^{7} - \frac{1}{72}\frac{p}{R}^{6} - \frac{37}{1008}\frac{p}{R}^{5} + \frac{1}{21}\frac{p}{R}^{4}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{8640}\frac{p}{R}^{16} + \frac{23}{705600}\frac{p}{R}^{15} + \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{9} + \frac{1591}{388080}\frac{p}{R}^{8} + \frac{1}{216}\frac{p}{R}^{7} - \frac{1}{72}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} - \frac{1}{60}\frac{p}{R}^{4} + \frac{1}{20}\frac{p}{R}^{3}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{8640}\frac{p}{R}^{16} - \frac{1}{7200}\frac{p}{R}^{15} + \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{264}\frac{p}{R}^{8} - \frac{1}{216}\frac{p}{R}^{7} + \frac{1}{72}\frac{p}{R}^{6}$
(0,0,1,0)	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} - \frac{313}{1100736}\frac{p}{R}^{15} - \frac{1}{3888}\frac{p}{R}^{13} - \frac{1}{5376}\frac{p}{R}^{12} + \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{1440}\frac{p}{R}^{9} - \frac{401}{48510}\frac{p}{R}^{8} + \frac{283}{40320}\frac{p}{R}^{7} + \frac{97}{14400}\frac{p}{R}^{6} - \frac{5}{112}\frac{p}{R}^{5} + \frac{7}{108}\frac{p}{R}^{4} + \frac{19}{180}\frac{p}{R}^{3} - \frac{5}{24}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{5616}\frac{p}{R}^{15} + \frac{1}{3888}\frac{p}{R}^{13} - \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{264}\frac{p}{R}^{8} + \frac{283}{40320}\frac{p}{R}^{7} + \frac{1}{252}\frac{p}{R}^{6} - \frac{1}{48}\frac{p}{R}^{5} - \frac{43}{1080}\frac{p}{R}^{4} + \frac{1}{12}\frac{p}{R}^{2}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{71}{12936}\frac{p}{R}^{8} + \frac{23}{8064}\frac{p}{R}^{7} - \frac{23}{4032}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} - \frac{1}{72}\frac{p}{R}^{4} + \frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{264}\frac{p}{R}^{8} - \frac{23}{8064}\frac{p}{R}^{7} + \frac{1}{72}\frac{p}{R}^{4}$

(0,0,1,1)	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{313}{1100736}\frac{p}{R}^{15} + \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{1848}\frac{p}{R}^{11} + \frac{401}{48510}\frac{p}{R}^{8} + \frac{1}{240}\frac{p}{R}^{7} - \frac{1}{200}\frac{p}{R}^{6} - \frac{11}{336}\frac{p}{R}^{5} - \frac{1}{40}\frac{p}{R}^{4} + \frac{1}{8}\frac{p}{R}^{2}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{5616}\frac{p}{R}^{15} - \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{264}\frac{p}{R}^{8} - \frac{1}{240}\frac{p}{R}^{7} + \frac{1}{48}\frac{p}{R}^{5}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{9408}\frac{p}{R}^{15} - \frac{1}{5376}\frac{p}{R}^{12} + \frac{1}{1848}\frac{p}{R}^{11} - \frac{71}{12936}\frac{p}{R}^{8} + \frac{1}{84}\frac{p}{R}^{5}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{264}\frac{p}{R}^{8}$
(0,1,0,0)	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{313}{1100736}\frac{p}{R}^{15} - \frac{1}{1680}\frac{p}{R}^{12} + \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{576}\frac{p}{R}^{9} - \frac{163}{17640}\frac{p}{R}^{8} + \frac{163}{15120}\frac{p}{R}^{7} + \frac{1}{24}\frac{p}{R}^{5} + \frac{41}{960}\frac{p}{R}^{4} + \frac{31}{480}\frac{p}{R}^{3} - \frac{31}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{1}{5616}\frac{p}{R}^{15} + \frac{1}{1680}\frac{p}{R}^{12} - \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{163}{15120}\frac{p}{R}^{7} + \frac{1}{252}\frac{p}{R}^{6} - \frac{1}{63}\frac{p}{R}^{5} - \frac{1}{960}\frac{p}{R}^{4} - \frac{1}{12}\frac{p}{R}^{3} + \frac{1}{9}\frac{p}{R}^{2}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{1}{9408}\frac{p}{R}^{15} - \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{576}\frac{p}{R}^{9} + \frac{1}{588}\frac{p}{R}^{8} + \frac{5}{756}\frac{p}{R}^{7} - \frac{1}{252}\frac{p}{R}^{6} - \frac{1}{960}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{12}\frac{p}{R}^{2}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{5}{756}\frac{p}{R}^{7} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{24}\frac{p}{R}^{3}$
(0,1,0,1)	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{313}{1100736}\frac{p}{R}^{15} + \frac{1}{1680}\frac{p}{R}^{12} + \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{9} + \frac{79}{17640}\frac{p}{R}^{8} + \frac{19}{2160}\frac{p}{R}^{7} - \frac{5}{168}\frac{p}{R}^{5} - \frac{1}{24}\frac{p}{R}^{4} + \frac{1}{9}\frac{p}{R}^{2}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{1}{5616}\frac{p}{R}^{15} - \frac{1}{1680}\frac{p}{R}^{12} + \frac{1}{1848}\frac{p}{R}^{11} - \frac{19}{2160}\frac{p}{R}^{7} + \frac{1}{63}\frac{p}{R}^{5}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{960}\frac{p}{R}^{9} - \frac{1}{588}\frac{p}{R}^{8} - \frac{1}{216}\frac{p}{R}^{7} + \frac{1}{60}\frac{p}{R}^{4}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{216}\frac{p}{R}^{7}$
(0,1,1,0)	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{313}{1100736}\frac{p}{R}^{15} - \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{79}{17640}\frac{p}{R}^{8} + \frac{31}{5040}\frac{p}{R}^{7} - \frac{1}{84}\frac{p}{R}^{5} - \frac{1}{24}\frac{p}{R}^{4} + \frac{1}{20}\frac{p}{R}^{3}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{5616}\frac{p}{R}^{15} + \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{31}{5040}\frac{p}{R}^{7} - \frac{1}{252}\frac{p}{R}^{6} + \frac{1}{60}\frac{p}{R}^{4}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{1440}\frac{p}{R}^{9} - \frac{1}{588}\frac{p}{R}^{8} - \frac{1}{504}\frac{p}{R}^{7} + \frac{1}{252}\frac{p}{R}^{6}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{1848}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{504}\frac{p}{R}^{7}$
(0,1,1,1)	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{313}{1100736}\frac{p}{R}^{15} + \frac{1}{1848}\frac{p}{R}^{11} - \frac{79}{17640}\frac{p}{R}^{8} - \frac{1}{240}\frac{p}{R}^{7} + \frac{1}{40}\frac{p}{R}^{4}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{5616}\frac{p}{R}^{15} - \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{240}\frac{p}{R}^{7}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{9408}\frac{p}{R}^{15} - \frac{1}{1848}\frac{p}{R}^{11} + \frac{1}{588}\frac{p}{R}^{8}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{1848}\frac{p}{R}^{11}$
(1,0,0,0)	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{4003}{27518400}\frac{p}{R}^{15} - \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{1280}\frac{p}{R}^{12} - \frac{1}{1344}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{576}\frac{p}{R}^{9} - \frac{1}{720}\frac{p}{R}^{8} + \frac{1}{1152}\frac{p}{R}^{7} + \frac{31}{11200}\frac{p}{R}^{6} + \frac{13}{504}\frac{p}{R}^{5} - \frac{47}{1728}\frac{p}{R}^{4} + \frac{11}{144}\frac{p}{R}^{3} - \frac{13}{48}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{11}{280800}\frac{p}{R}^{15} + \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{1680}\frac{p}{R}^{12} - \frac{1}{1344}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1152}\frac{p}{R}^{7} + \frac{1}{252}\frac{p}{R}^{6} + \frac{11}{1728}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{12}\frac{p}{R}^{2}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{8640}\frac{p}{R}^{16} + \frac{23}{705600}\frac{p}{R}^{15} + \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{576}\frac{p}{R}^{9} - \frac{1}{720}\frac{p}{R}^{8} + \frac{1}{1152}\frac{p}{R}^{7} - \frac{1}{576}\frac{p}{R}^{6} + \frac{1}{64}\frac{p}{R}^{4} - \frac{1}{16}\frac{p}{R}^{3} + \frac{1}{16}\frac{p}{R}^{2}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{8640}\frac{p}{R}^{16} - \frac{1}{7200}\frac{p}{R}^{15} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{1152}\frac{p}{R}^{7} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{32}\frac{p}{R}^{3}$
(1,0,0,1)	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{4003}{27518400}\frac{p}{R}^{15} + \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{1280}\frac{p}{R}^{12} - \frac{1}{1344}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{9} + \frac{1}{720}\frac{p}{R}^{8} - \frac{1}{200}\frac{p}{R}^{6} - \frac{1}{72}\frac{p}{R}^{5} + \frac{1}{48}\frac{p}{R}^{4}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{11}{280800}\frac{p}{R}^{15} - \frac{1}{3888}\frac{p}{R}^{13} - \frac{1}{1680}\frac{p}{R}^{12} + \frac{1}{1344}\frac{p}{R}^{11}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{8640}\frac{p}{R}^{16} - \frac{23}{705600}\frac{p}{R}^{15} - \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{960}\frac{p}{R}^{9} + \frac{1}{720}\frac{p}{R}^{8}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{8640}\frac{p}{R}^{16} + \frac{1}{7200}\frac{p}{R}^{15}$
(1,0,1,0)	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{313}{1100736}\frac{p}{R}^{15} + \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{1152}\frac{p}{R}^{7} - \frac{31}{11200}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} - \frac{1}{108}\frac{p}{R}^{4} + \frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{5616}\frac{p}{R}^{15} - \frac{1}{3888}\frac{p}{R}^{13} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{1152}\frac{p}{R}^{7} - \frac{1}{252}\frac{p}{R}^{6} + \frac{1}{108}\frac{p}{R}^{4}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} + \frac{1}{9408}\frac{p}{R}^{15} - \frac{1}{5376}\frac{p}{R}^{12} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{1440}\frac{p}{R}^{9} - \frac{1}{1152}\frac{p}{R}^{7} + \frac{1}{576}\frac{p}{R}^{6}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1152}\frac{p}{R}^{7}$
(1,0,1,1)	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19} - \frac{313}{1100736}\frac{p}{R}^{15} - \frac{1}{3888}\frac{p}{R}^{13} - \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{200}\frac{p}{R}^{6}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{5616}\frac{p}{R}^{15} + \frac{1}{3888}\frac{p}{R}^{13}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{38016}\frac{p}{R}^{19} - \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{5376}\frac{p}{R}^{12}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{38016}\frac{p}{R}^{19}$
(1,1,0,0)	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{313}{1100736}\frac{p}{R}^{15} + \frac{1}{1680}\frac{p}{R}^{12} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{576}\frac{p}{R}^{9} + \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{252}\frac{p}{R}^{6} - \frac{13}{504}\frac{p}{R}^{5} + \frac{1}{64}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{8}\frac{p}{R}^{2}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{1}{5616}\frac{p}{R}^{15} - \frac{1}{1680}\frac{p}{R}^{12} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{252}\frac{p}{R}^{6} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{24}\frac{p}{R}^{3}$	$-\frac{1}{57024}\frac{p}{R}^{22} + \frac{1}{8640}\frac{p}{R}^{16} + \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{576}\frac{p}{R}^{9} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{32}\frac{p}{R}^{3}$	$\frac{1}{57024}\frac{p}{R}^{22} - \frac{1}{8640}\frac{p}{R}^{16} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{64}\frac{p}{R}^{4}$

(1,1,0,1)	$-\frac{1}{57024}\left(\frac{p}{R}\right)^{22}+\frac{1}{8640}\left(\frac{p}{R}\right)^{16}+\frac{313}{1100736}\left(\frac{p}{R}\right)^{15}-\frac{1}{1680}\left(\frac{p}{R}\right)^{12}-\frac{1}{960}\left(\frac{p}{R}\right)^{9}-\frac{1}{360}\left(\frac{p}{R}\right)^{8}+\frac{1}{72}\left(\frac{p}{R}\right)^{5}$	$\frac{1}{57024}\left(\frac{p}{R}\right)^{22}-\frac{1}{8640}\left(\frac{p}{R}\right)^{16}-\frac{1}{5616}\left(\frac{p}{R}\right)^{15}+\frac{1}{1680}\left(\frac{p}{R}\right)^{12}$	$\frac{1}{57024}\left(\frac{p}{R}\right)^{22}-\frac{1}{8640}\left(\frac{p}{R}\right)^{16}-\frac{1}{9408}\left(\frac{p}{R}\right)^{15}+\frac{1}{960}\left(\frac{p}{R}\right)^{9}$	$-\frac{1}{57024}\left(\frac{p}{R}\right)^{22}+\frac{1}{8640}\left(\frac{p}{R}\right)^{16}$
(1,1,1,0)	$-\frac{1}{57024}\left(\frac{p}{R}\right)^{22}+\frac{313}{1100736}\left(\frac{p}{R}\right)^{15}+\frac{1}{2880}\left(\frac{p}{R}\right)^{10}-\frac{1}{1440}\left(\frac{p}{R}\right)^{9}-\frac{1}{360}\left(\frac{p}{R}\right)^{8}-\frac{1}{252}\left(\frac{p}{R}\right)^{6}+\frac{1}{84}\left(\frac{p}{R}\right)^{5}$	$\frac{1}{57024}\left(\frac{p}{R}\right)^{22}-\frac{1}{5616}\left(\frac{p}{R}\right)^{15}-\frac{1}{2880}\left(\frac{p}{R}\right)^{10}+\frac{1}{252}\left(\frac{p}{R}\right)^{6}$	$\frac{1}{57024}\left(\frac{p}{R}\right)^{22}-\frac{1}{9408}\left(\frac{p}{R}\right)^{15}-\frac{1}{2880}\left(\frac{p}{R}\right)^{10}+\frac{1}{1440}\left(\frac{p}{R}\right)^{9}$	$-\frac{1}{57024}\left(\frac{p}{R}\right)^{22}+\frac{1}{2880}\left(\frac{p}{R}\right)^{10}$
(1,1,1,1)	$\frac{1}{57024}\left(\frac{p}{R}\right)^{22}-\frac{313}{1100736}\left(\frac{p}{R}\right)^{15}+\frac{1}{360}\left(\frac{p}{R}\right)^{8}$	$-\frac{1}{57024}\left(\frac{p}{R}\right)^{22}+\frac{1}{5616}\left(\frac{p}{R}\right)^{15}$	$-\frac{1}{57024}\left(\frac{p}{R}\right)^{22}+\frac{1}{9408}\left(\frac{p}{R}\right)^{15}$	$\frac{1}{57024}\left(\frac{p}{R}\right)^{22}$

Table D.3: Probability of the 6 consecutive operations $X_{M_i}, X_{S_i}, X_{M_{i-1}}, X_{S_{i-1}}, X_{M_{i-2}}, X_{S_{i-2}}$ for $k_i \oplus k_{i-1} = 0$ and $k_{i-1} \oplus k_{i-2} = 0$ (case (a))

$(x_{M_i}, x_{S_i}, x_{M_{i-1}}, x_{S_{i-1}})$	$(x_{M_{i-2}}, x_{S_{i-2}})$ (0,0)	(0,1)	(1,0)	(1,1)
(0,0,0,0)	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{36000}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{101}{423360}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{59}{116480}\frac{p}{R}^{12} + \frac{3403}{3880800}\frac{p}{R}^{11} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{103}{25920}\frac{p}{R}^{9} + \frac{1439}{529200}\frac{p}{R}^{8} - \frac{3349}{302400}\frac{p}{R}^{6} - \frac{253}{5040}\frac{p}{R}^{5} - \frac{3}{320}\frac{p}{R}^{4} - \frac{59}{2160}\frac{p}{R}^{3} + \frac{15}{16}\frac{p}{R}^{2} - \frac{7}{4}\frac{p}{R} + 1$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{36000}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{7560}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{3120}\frac{p}{R}^{12} - \frac{3403}{3880800}\frac{p}{R}^{11} + \frac{1}{1920}\frac{p}{R}^{10} - \frac{29}{12960}\frac{p}{R}^{9} + \frac{1}{2700}\frac{p}{R}^{8} + \frac{1}{2700}\frac{p}{R}^{6} + \frac{1}{80}\frac{p}{R}^{5} - \frac{11}{960}\frac{p}{R}^{4} + \frac{131}{864}\frac{p}{R}^{3} - \frac{31}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{36000}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{5376}\frac{p}{R}^{12} - \frac{47}{79200}\frac{p}{R}^{11} + \frac{1}{1920}\frac{p}{R}^{10} - \frac{7}{2880}\frac{p}{R}^{9} - \frac{109}{35280}\frac{p}{R}^{8} + \frac{1079}{100800}\frac{p}{R}^{6} + \frac{1}{280}\frac{p}{R}^{5} + \frac{5}{192}\frac{p}{R}^{4} + \frac{19}{720}\frac{p}{R}^{3} - \frac{13}{48}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{36000}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{47}{79200}\frac{p}{R}^{11} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{1}{360}\frac{p}{R}^{8} - \frac{1}{200}\frac{p}{R}^{6} + \frac{1}{120}\frac{p}{R}^{5} - \frac{3}{320}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{8}\frac{p}{R}^{2}$
(0,0,0,1)	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{36000}\frac{p}{R}^{17} + \frac{101}{423360}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{3528}\frac{p}{R}^{11} + \frac{1}{1920}\frac{p}{R}^{10} - \frac{17}{5184}\frac{p}{R}^{9} - \frac{23}{3920}\frac{p}{R}^{8} + \frac{1}{2700}\frac{p}{R}^{6} + \frac{193}{5040}\frac{p}{R}^{5} + \frac{11}{240}\frac{p}{R}^{4} - \frac{7}{540}\frac{p}{R}^{3} - \frac{25}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{36000}\frac{p}{R}^{17} - \frac{1}{7560}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{3528}\frac{p}{R}^{11} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{29}{12960}\frac{p}{R}^{9} + \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{216}\frac{p}{R}^{6} - \frac{1}{80}\frac{p}{R}^{5} - \frac{1}{40}\frac{p}{R}^{4} - \frac{1}{27}\frac{p}{R}^{3} + \frac{1}{9}\frac{p}{R}^{2}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{36000}\frac{p}{R}^{17} - \frac{1}{9408}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{5376}\frac{p}{R}^{12} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{1}{576}\frac{p}{R}^{9} + \frac{109}{35280}\frac{p}{R}^{8} - \frac{1}{280}\frac{p}{R}^{5} - \frac{1}{24}\frac{p}{R}^{4} + \frac{1}{20}\frac{p}{R}^{3}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{36000}\frac{p}{R}^{17} + \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{1920}\frac{p}{R}^{10} - \frac{1}{1440}\frac{p}{R}^{9} - \frac{1}{360}\frac{p}{R}^{8} - \frac{1}{120}\frac{p}{R}^{5} + \frac{1}{40}\frac{p}{R}^{4}$
(0,0,1,0)	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{36000}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{101}{423360}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{59}{116480}\frac{p}{R}^{12} - \frac{3403}{3880800}\frac{p}{R}^{11} - \frac{29}{12960}\frac{p}{R}^{9} - \frac{1087}{264600}\frac{p}{R}^{8} + \frac{1583}{100800}\frac{p}{R}^{6} + \frac{19}{504}\frac{p}{R}^{5} + \frac{1}{240}\frac{p}{R}^{4} - \frac{19}{180}\frac{p}{R}^{3} - \frac{1}{8}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{36000}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{7560}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{3120}\frac{p}{R}^{12} + \frac{3403}{3880800}\frac{p}{R}^{11} + \frac{1}{648}\frac{p}{R}^{9} - \frac{1}{2700}\frac{p}{R}^{8} - \frac{1}{200}\frac{p}{R}^{6} - \frac{1}{72}\frac{p}{R}^{5} + \frac{1}{48}\frac{p}{R}^{4}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{36000}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{9408}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{5376}\frac{p}{R}^{12} + \frac{47}{79200}\frac{p}{R}^{11} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{79}{17640}\frac{p}{R}^{8} - \frac{1079}{100800}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} + \frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{36000}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{47}{79200}\frac{p}{R}^{11} - \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{200}\frac{p}{R}^{6}$
(0,0,1,1)	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{36000}\frac{p}{R}^{17} - \frac{101}{423360}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{5376}\frac{p}{R}^{12} + \frac{1}{3528}\frac{p}{R}^{11} + \frac{1}{648}\frac{p}{R}^{9} + \frac{16}{2205}\frac{p}{R}^{8} - \frac{1}{200}\frac{p}{R}^{6} - \frac{13}{504}\frac{p}{R}^{5} - \frac{1}{40}\frac{p}{R}^{4} + \frac{1}{8}\frac{p}{R}^{2}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{36000}\frac{p}{R}^{17} + \frac{1}{7560}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{3528}\frac{p}{R}^{11} - \frac{1}{648}\frac{p}{R}^{9} - \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{72}\frac{p}{R}^{5}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{36000}\frac{p}{R}^{17} + \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{5376}\frac{p}{R}^{12} - \frac{79}{17640}\frac{p}{R}^{8} + \frac{1}{84}\frac{p}{R}^{5}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{36000}\frac{p}{R}^{17} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{360}\frac{p}{R}^{8}$
(0,1,0,0)	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{101}{423360}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{3120}\frac{p}{R}^{12} - \frac{1}{2200}\frac{p}{R}^{11} + \frac{1}{1920}\frac{p}{R}^{10} - \frac{17}{5184}\frac{p}{R}^{9} - \frac{229}{10584}\frac{p}{R}^{8} - \frac{1}{1512}\frac{p}{R}^{6} + \frac{11}{630}\frac{p}{R}^{5} + \frac{5}{192}\frac{p}{R}^{4} + \frac{439}{4320}\frac{p}{R}^{3} - \frac{31}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{7560}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{3120}\frac{p}{R}^{12} + \frac{1}{2200}\frac{p}{R}^{11} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{1}{648}\frac{p}{R}^{9} - \frac{1}{432}\frac{p}{R}^{8} + \frac{1}{216}\frac{p}{R}^{6} + \frac{1}{120}\frac{p}{R}^{5} + \frac{1}{64}\frac{p}{R}^{4} - \frac{13}{108}\frac{p}{R}^{3} + \frac{1}{9}\frac{p}{R}^{2}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{9408}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{2200}\frac{p}{R}^{11} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{1}{576}\frac{p}{R}^{9} + \frac{1}{588}\frac{p}{R}^{8} - \frac{1}{252}\frac{p}{R}^{6} + \frac{1}{120}\frac{p}{R}^{5} - \frac{1}{960}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{12}\frac{p}{R}^{2}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{2200}\frac{p}{R}^{11} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{1}{120}\frac{p}{R}^{5} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{24}\frac{p}{R}^{3}$
(0,1,0,1)	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{101}{423360}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{67}{25920}\frac{p}{R}^{9} + \frac{79}{17640}\frac{p}{R}^{8} + \frac{1}{216}\frac{p}{R}^{6} - \frac{1}{180}\frac{p}{R}^{5} - \frac{1}{24}\frac{p}{R}^{4} - \frac{1}{27}\frac{p}{R}^{3} + \frac{1}{9}\frac{p}{R}^{2}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{7560}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{1920}\frac{p}{R}^{10} - \frac{1}{648}\frac{p}{R}^{9} - \frac{1}{216}\frac{p}{R}^{6} - \frac{1}{120}\frac{p}{R}^{5} + \frac{1}{27}\frac{p}{R}^{3}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{1920}\frac{p}{R}^{10} - \frac{1}{960}\frac{p}{R}^{9} - \frac{1}{588}\frac{p}{R}^{8} - \frac{1}{120}\frac{p}{R}^{5} + \frac{1}{60}\frac{p}{R}^{4}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{1920}\frac{p}{R}^{10} + \frac{1}{120}\frac{p}{R}^{5}$
(0,1,1,0)	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{101}{423360}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{3120}\frac{p}{R}^{12} + \frac{1}{2200}\frac{p}{R}^{11} + \frac{29}{12960}\frac{p}{R}^{9} + \frac{229}{105840}\frac{p}{R}^{8} - \frac{1}{252}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} - \frac{1}{40}\frac{p}{R}^{4} + \frac{1}{20}\frac{p}{R}^{3}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{7560}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{3120}\frac{p}{R}^{12} - \frac{1}{2200}\frac{p}{R}^{11} - \frac{1}{648}\frac{p}{R}^{9} + \frac{1}{432}\frac{p}{R}^{8}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{9408}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{2200}\frac{p}{R}^{11} - \frac{1}{1440}\frac{p}{R}^{9} - \frac{1}{588}\frac{p}{R}^{8} + \frac{1}{252}\frac{p}{R}^{6}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{3960}\frac{p}{R}^{16} - \frac{1}{2200}\frac{p}{R}^{11}$
(0,1,1,1)	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{101}{423360}\frac{p}{R}^{15} + \frac{1}{3960}\frac{p}{R}^{13} - \frac{1}{648}\frac{p}{R}^{9} - \frac{79}{17640}\frac{p}{R}^{8} + \frac{1}{40}\frac{p}{R}^{4}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{7560}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{648}\frac{p}{R}^{9}$	$\frac{1}{72000}\frac{p}{R}^{22} - \frac{1}{9408}\frac{p}{R}^{15} - \frac{1}{3960}\frac{p}{R}^{13} + \frac{1}{588}\frac{p}{R}^{8}$	$-\frac{1}{72000}\frac{p}{R}^{22} + \frac{1}{3960}\frac{p}{R}^{13}$

(1,0,0,0)	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{36000}\frac{p}{R}^{17}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{101}{423360}\frac{p}{R}^{15}-\frac{59}{116480}\frac{p}{R}^{12}-\frac{149}{352800}\frac{p}{R}^{11}+\frac{1}{1920}\frac{p}{R}^{10}-\frac{7}{2880}\frac{p}{R}^{9}-\frac{1}{1800}\frac{p}{R}^{8}+\frac{91}{43200}\frac{p}{R}^{6}+\frac{11}{336}\frac{p}{R}^{5}-\frac{7}{192}\frac{p}{R}^{4}+\frac{11}{144}\frac{p}{R}^{3}-\frac{13}{48}\frac{p}{R}^{2}+\frac{1}{4}\frac{p}{R}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{36000}\frac{p}{R}^{17}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{7560}\frac{p}{R}^{15}+\frac{1}{3120}\frac{p}{R}^{12}+\frac{149}{352800}\frac{p}{R}^{11}-\frac{1}{1920}\frac{p}{R}^{10}+\frac{1}{1440}\frac{p}{R}^{9}-\frac{1}{1200}\frac{p}{R}^{8}+\frac{1}{216}\frac{p}{R}^{6}-\frac{1}{144}\frac{p}{R}^{5}+\frac{1}{64}\frac{p}{R}^{4}-\frac{7}{96}\frac{p}{R}^{3}+\frac{1}{12}\frac{p}{R}^{2}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{36000}\frac{p}{R}^{17}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{9408}\frac{p}{R}^{15}+\frac{1}{5376}\frac{p}{R}^{12}+\frac{1}{7200}\frac{p}{R}^{11}-\frac{1}{1920}\frac{p}{R}^{10}+\frac{7}{2880}\frac{p}{R}^{9}-\frac{1}{720}\frac{p}{R}^{8}-\frac{1}{576}\frac{p}{R}^{6}+\frac{1}{64}\frac{p}{R}^{4}-\frac{1}{16}\frac{p}{R}^{3}+\frac{1}{16}\frac{p}{R}^{2}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{36000}\frac{p}{R}^{17}+\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{7200}\frac{p}{R}^{11}+\frac{1}{1920}\frac{p}{R}^{10}-\frac{1}{1440}\frac{p}{R}^{9}-\frac{1}{64}\frac{p}{R}^{4}+\frac{1}{32}\frac{p}{R}^{3}$
(1,0,0,1)	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{36000}\frac{p}{R}^{17}-\frac{101}{423360}\frac{p}{R}^{15}+\frac{1}{5376}\frac{p}{R}^{12}+\frac{1}{3528}\frac{p}{R}^{11}-\frac{1}{1920}\frac{p}{R}^{10}+\frac{1}{576}\frac{p}{R}^{9}+\frac{1}{720}\frac{p}{R}^{8}-\frac{1}{2700}\frac{p}{R}^{6}-\frac{1}{48}\frac{p}{R}^{5}+\frac{1}{48}\frac{p}{R}^{4}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{36000}\frac{p}{R}^{17}+\frac{1}{7560}\frac{p}{R}^{15}-\frac{1}{3528}\frac{p}{R}^{11}+\frac{1}{1920}\frac{p}{R}^{10}-\frac{1}{1440}\frac{p}{R}^{9}-\frac{1}{216}\frac{p}{R}^{6}+\frac{1}{144}\frac{p}{R}^{5}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{36000}\frac{p}{R}^{17}+\frac{1}{9408}\frac{p}{R}^{15}-\frac{1}{5376}\frac{p}{R}^{12}+\frac{1}{1920}\frac{p}{R}^{10}-\frac{1}{576}\frac{p}{R}^{9}+\frac{1}{720}\frac{p}{R}^{8}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{36000}\frac{p}{R}^{17}-\frac{1}{1920}\frac{p}{R}^{10}+\frac{1}{1440}\frac{p}{R}^{9}$
(1,0,1,0)	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{36000}\frac{p}{R}^{17}-\frac{1}{19200}\frac{p}{R}^{16}+\frac{101}{423360}\frac{p}{R}^{15}+\frac{59}{116480}\frac{p}{R}^{12}+\frac{149}{352800}\frac{p}{R}^{11}+\frac{1}{1440}\frac{p}{R}^{9}+\frac{7}{3600}\frac{p}{R}^{8}-\frac{97}{14400}\frac{p}{R}^{6}-\frac{1}{84}\frac{p}{R}^{5}+\frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{36000}\frac{p}{R}^{17}+\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{7560}\frac{p}{R}^{15}-\frac{1}{3120}\frac{p}{R}^{12}-\frac{149}{352800}\frac{p}{R}^{11}+\frac{1}{1200}\frac{p}{R}^{8}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{36000}\frac{p}{R}^{17}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{9408}\frac{p}{R}^{15}-\frac{1}{5376}\frac{p}{R}^{12}-\frac{1}{7200}\frac{p}{R}^{11}-\frac{1}{1440}\frac{p}{R}^{9}+\frac{1}{576}\frac{p}{R}^{6}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{36000}\frac{p}{R}^{17}-\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{7200}\frac{p}{R}^{11}$
(1,0,1,1)	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{36000}\frac{p}{R}^{17}+\frac{101}{423360}\frac{p}{R}^{15}-\frac{1}{5376}\frac{p}{R}^{12}-\frac{1}{3528}\frac{p}{R}^{11}-\frac{1}{360}\frac{p}{R}^{8}+\frac{1}{200}\frac{p}{R}^{6}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{36000}\frac{p}{R}^{17}-\frac{1}{7560}\frac{p}{R}^{15}+\frac{1}{3528}\frac{p}{R}^{11}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{36000}\frac{p}{R}^{17}-\frac{1}{9408}\frac{p}{R}^{15}+\frac{1}{5376}\frac{p}{R}^{12}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{36000}\frac{p}{R}^{17}$
(1,1,0,0)	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{101}{423360}\frac{p}{R}^{15}+\frac{1}{3120}\frac{p}{R}^{12}-\frac{1}{1920}\frac{p}{R}^{10}+\frac{1}{576}\frac{p}{R}^{9}+\frac{1}{360}\frac{p}{R}^{8}-\frac{1}{216}\frac{p}{R}^{6}+\frac{13}{504}\frac{p}{R}^{5}+\frac{1}{64}\frac{p}{R}^{4}-\frac{7}{96}\frac{p}{R}^{3}+\frac{1}{8}\frac{p}{R}^{2}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{7560}\frac{p}{R}^{15}-\frac{1}{3120}\frac{p}{R}^{12}+\frac{1}{1920}\frac{p}{R}^{10}-\frac{1}{216}\frac{p}{R}^{6}-\frac{1}{64}\frac{p}{R}^{4}+\frac{1}{24}\frac{p}{R}^{3}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{9408}\frac{p}{R}^{15}+\frac{1}{1920}\frac{p}{R}^{10}-\frac{1}{576}\frac{p}{R}^{9}-\frac{1}{64}\frac{p}{R}^{4}+\frac{1}{32}\frac{p}{R}^{3}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{1920}\frac{p}{R}^{10}+\frac{1}{64}\frac{p}{R}^{4}$
(1,1,0,1)	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{101}{423360}\frac{p}{R}^{15}+\frac{1}{1920}\frac{p}{R}^{10}-\frac{1}{960}\frac{p}{R}^{9}-\frac{1}{360}\frac{p}{R}^{8}-\frac{1}{216}\frac{p}{R}^{6}+\frac{1}{72}\frac{p}{R}^{5}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{7560}\frac{p}{R}^{15}-\frac{1}{1920}\frac{p}{R}^{10}+\frac{1}{216}\frac{p}{R}^{6}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{9408}\frac{p}{R}^{15}-\frac{1}{1920}\frac{p}{R}^{10}+\frac{1}{960}\frac{p}{R}^{9}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{1920}\frac{p}{R}^{10}$
(1,1,1,0)	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{101}{423360}\frac{p}{R}^{15}-\frac{1}{3120}\frac{p}{R}^{12}-\frac{1}{1440}\frac{p}{R}^{9}-\frac{1}{360}\frac{p}{R}^{8}+\frac{1}{84}\frac{p}{R}^{5}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{7560}\frac{p}{R}^{15}+\frac{1}{3120}\frac{p}{R}^{12}$	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{9408}\frac{p}{R}^{15}+\frac{1}{1440}\frac{p}{R}^{9}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}$
(1,1,1,1)	$\frac{1}{72000}\frac{p}{R}^{22}-\frac{101}{423360}\frac{p}{R}^{15}+\frac{1}{360}\frac{p}{R}^{8}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{7560}\frac{p}{R}^{15}$	$-\frac{1}{72000}\frac{p}{R}^{22}+\frac{1}{9408}\frac{p}{R}^{15}$	$\frac{1}{72000}\frac{p}{R}^{22}$

Table D.4: Probability of the 6 consecutive operations $X_{M_i}, X_{S_i}, X_{M_{i-1}}, X_{S_{i-1}}, X_{M_{i-2}}, X_{S_{i-2}}$ for $k_i \oplus k_{i-1} = 0$ and $k_{i-1} \oplus k_{i-2} = 1$ (case (b))

$(x_{M_i}, x_{S_i}, x_{M_{i-1}}, x_{S_{i-1}})$	$(x_{M_{i-2}}, x_{S_{i-2}})$ (0,0)	(0,1)	(1,0)	(1,1)
(0,0,0,0)	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{16} + \frac{23}{114912}\frac{p}{R}^{15} + \frac{1}{816}\frac{p}{R}^{13} + \frac{23}{10080}\frac{p}{R}^{12} - \frac{1513}{443520}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{360}\frac{p}{R}^{9} + \frac{93}{24640}\frac{p}{R}^{8} + \frac{209}{24192}\frac{p}{R}^{7} - \frac{17}{576}\frac{p}{R}^{6} - \frac{13}{168}\frac{p}{R}^{5} + \frac{2701}{60480}\frac{p}{R}^{4} + \frac{7}{720}\frac{p}{R}^{3} + \frac{127}{144}\frac{p}{R}^{2} - \frac{7}{4}\frac{p}{R} + 1$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{2736}\frac{p}{R}^{15} - \frac{1}{816}\frac{p}{R}^{13} - \frac{1}{672}\frac{p}{R}^{12} + \frac{1513}{443520}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{264}\frac{p}{R}^{8} - \frac{209}{24192}\frac{p}{R}^{7} + \frac{5}{504}\frac{p}{R}^{6} + \frac{11}{336}\frac{p}{R}^{5} - \frac{1417}{60480}\frac{p}{R}^{4} + \frac{11}{96}\frac{p}{R}^{3} - \frac{29}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2880}\frac{p}{R}^{16} - \frac{65}{114912}\frac{p}{R}^{15} - \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{360}\frac{p}{R}^{9} + \frac{1}{73920}\frac{p}{R}^{8} - \frac{17}{2688}\frac{p}{R}^{7} + \frac{79}{4032}\frac{p}{R}^{6} + \frac{1}{84}\frac{p}{R}^{5} + \frac{31}{2880}\frac{p}{R}^{4} + \frac{19}{720}\frac{p}{R}^{3} - \frac{13}{48}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{1368}\frac{p}{R}^{15} - \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{264}\frac{p}{R}^{8} + \frac{17}{2688}\frac{p}{R}^{7} - \frac{1}{72}\frac{p}{R}^{6} + \frac{1}{576}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{8}\frac{p}{R}^{2}$
(0,0,0,1)	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2880}\frac{p}{R}^{16} - \frac{23}{114912}\frac{p}{R}^{15} - \frac{1}{816}\frac{p}{R}^{13} - \frac{23}{10080}\frac{p}{R}^{12} + \frac{1513}{443520}\frac{p}{R}^{11} - \frac{1}{480}\frac{p}{R}^{9} - \frac{93}{24640}\frac{p}{R}^{8} - \frac{5}{864}\frac{p}{R}^{7} + \frac{1}{36}\frac{p}{R}^{6} + \frac{11}{168}\frac{p}{R}^{5} - \frac{347}{5040}\frac{p}{R}^{4} - \frac{1}{20}\frac{p}{R}^{3} - \frac{5}{24}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{2736}\frac{p}{R}^{15} + \frac{1}{816}\frac{p}{R}^{13} + \frac{1}{672}\frac{p}{R}^{12} - \frac{1513}{443520}\frac{p}{R}^{11} + \frac{1}{264}\frac{p}{R}^{8} + \frac{5}{864}\frac{p}{R}^{7} - \frac{1}{72}\frac{p}{R}^{6} - \frac{11}{336}\frac{p}{R}^{5} + \frac{1}{21}\frac{p}{R}^{4}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{16} + \frac{65}{114912}\frac{p}{R}^{15} + \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{480}\frac{p}{R}^{9} - \frac{1}{73920}\frac{p}{R}^{8} - \frac{1}{72}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} - \frac{1}{80}\frac{p}{R}^{4} + \frac{1}{20}\frac{p}{R}^{3}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{1368}\frac{p}{R}^{15} + \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{264}\frac{p}{R}^{8} - \frac{1}{288}\frac{p}{R}^{7} + \frac{1}{72}\frac{p}{R}^{6}$
(0,0,1,0)	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{61}{114912}\frac{p}{R}^{15} - \frac{1}{816}\frac{p}{R}^{13} - \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{440}\frac{p}{R}^{9} - \frac{41}{4928}\frac{p}{R}^{8} - \frac{125}{24192}\frac{p}{R}^{7} + \frac{1}{64}\frac{p}{R}^{6} + \frac{5}{112}\frac{p}{R}^{5} + \frac{29}{540}\frac{p}{R}^{4} - \frac{19}{180}\frac{p}{R}^{3} - \frac{5}{24}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2736}\frac{p}{R}^{15} + \frac{1}{816}\frac{p}{R}^{13} - \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{264}\frac{p}{R}^{8} + \frac{125}{24192}\frac{p}{R}^{7} - \frac{1}{48}\frac{p}{R}^{5} - \frac{43}{1080}\frac{p}{R}^{4} + \frac{1}{12}\frac{p}{R}^{2}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{67}{14784}\frac{p}{R}^{8} + \frac{23}{8064}\frac{p}{R}^{7} - \frac{23}{4032}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} - \frac{1}{72}\frac{p}{R}^{4} + \frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{264}\frac{p}{R}^{8} - \frac{23}{8064}\frac{p}{R}^{7} + \frac{1}{72}\frac{p}{R}^{4}$
(0,0,1,1)	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{61}{114912}\frac{p}{R}^{15} + \frac{1}{816}\frac{p}{R}^{13} + \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{4224}\frac{p}{R}^{11} + \frac{41}{4928}\frac{p}{R}^{8} + \frac{1}{432}\frac{p}{R}^{7} - \frac{1}{72}\frac{p}{R}^{6} - \frac{11}{336}\frac{p}{R}^{5} - \frac{1}{72}\frac{p}{R}^{4} + \frac{1}{8}\frac{p}{R}^{2}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2736}\frac{p}{R}^{15} - \frac{1}{816}\frac{p}{R}^{13} + \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{264}\frac{p}{R}^{8} - \frac{1}{432}\frac{p}{R}^{7} + \frac{1}{48}\frac{p}{R}^{5}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{4224}\frac{p}{R}^{11} - \frac{67}{14784}\frac{p}{R}^{8} + \frac{1}{84}\frac{p}{R}^{5}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{264}\frac{p}{R}^{8}$
(0,1,0,0)	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2880}\frac{p}{R}^{16} - \frac{23}{114912}\frac{p}{R}^{15} - \frac{1}{816}\frac{p}{R}^{13} - \frac{23}{10080}\frac{p}{R}^{12} + \frac{1}{315}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{360}\frac{p}{R}^{9} + \frac{1}{1320}\frac{p}{R}^{8} - \frac{1}{504}\frac{p}{R}^{7} + \frac{1}{72}\frac{p}{R}^{6} + \frac{11}{336}\frac{p}{R}^{5} - \frac{313}{6720}\frac{p}{R}^{4} + \frac{31}{480}\frac{p}{R}^{3} - \frac{23}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{2736}\frac{p}{R}^{15} + \frac{1}{816}\frac{p}{R}^{13} + \frac{1}{672}\frac{p}{R}^{12} - \frac{1}{315}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{504}\frac{p}{R}^{7} + \frac{1}{252}\frac{p}{R}^{6} - \frac{1}{960}\frac{p}{R}^{4} - \frac{1}{12}\frac{p}{R}^{3} + \frac{1}{9}\frac{p}{R}^{2}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{16} + \frac{65}{114912}\frac{p}{R}^{15} + \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{360}\frac{p}{R}^{9} - \frac{1}{220}\frac{p}{R}^{8} + \frac{1}{504}\frac{p}{R}^{7} - \frac{1}{252}\frac{p}{R}^{6} + \frac{1}{64}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{12}\frac{p}{R}^{2}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{1368}\frac{p}{R}^{15} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{504}\frac{p}{R}^{7} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{24}\frac{p}{R}^{3}$
(0,1,0,1)	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{16} + \frac{23}{114912}\frac{p}{R}^{15} + \frac{1}{816}\frac{p}{R}^{13} + \frac{23}{10080}\frac{p}{R}^{12} - \frac{1}{315}\frac{p}{R}^{11} + \frac{1}{480}\frac{p}{R}^{9} - \frac{1}{1320}\frac{p}{R}^{8} - \frac{1}{72}\frac{p}{R}^{6} - \frac{1}{48}\frac{p}{R}^{5} + \frac{1}{21}\frac{p}{R}^{4}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{2736}\frac{p}{R}^{15} - \frac{1}{816}\frac{p}{R}^{13} + \frac{1}{672}\frac{p}{R}^{12} + \frac{1}{315}\frac{p}{R}^{11}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2880}\frac{p}{R}^{16} - \frac{65}{114912}\frac{p}{R}^{15} - \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{480}\frac{p}{R}^{9} + \frac{1}{220}\frac{p}{R}^{8}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{1368}\frac{p}{R}^{15}$
(0,1,1,0)	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{61}{114912}\frac{p}{R}^{15} + \frac{1}{816}\frac{p}{R}^{13} + \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{p}{1440}\frac{p}{R}^{9} + \frac{1}{264}\frac{p}{R}^{8} + \frac{1}{504}\frac{p}{R}^{7} - \frac{1}{72}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} - \frac{1}{60}\frac{p}{R}^{4} + \frac{1}{20}\frac{p}{R}^{3}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{2736}\frac{p}{R}^{15} - \frac{1}{816}\frac{p}{R}^{13} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{504}\frac{p}{R}^{7} - \frac{1}{252}\frac{p}{R}^{6} + \frac{1}{60}\frac{p}{R}^{4}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{1440}\frac{p}{R}^{9} - \frac{1}{504}\frac{p}{R}^{7} + \frac{1}{252}\frac{p}{R}^{6}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{504}\frac{p}{R}^{7}$
(0,1,1,1)	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19} + \frac{61}{114912}\frac{p}{R}^{15} - \frac{1}{816}\frac{p}{R}^{13} - \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{264}\frac{p}{R}^{8} + \frac{1}{72}\frac{p}{R}^{6}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{2736}\frac{p}{R}^{15} + \frac{1}{816}\frac{p}{R}^{13}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6072}\frac{p}{R}^{19} - \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{1260}\frac{p}{R}^{12}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6072}\frac{p}{R}^{19}$
(1,0,0,0)	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{16} + \frac{61}{114912}\frac{p}{R}^{15} - \frac{1}{672}\frac{p}{R}^{12} + \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{360}\frac{p}{R}^{9} - \frac{67}{14784}\frac{p}{R}^{8} - \frac{23}{3456}\frac{p}{R}^{7} - \frac{1}{448}\frac{p}{R}^{6} + \frac{5}{112}\frac{p}{R}^{5} + \frac{173}{8640}\frac{p}{R}^{4} + \frac{11}{144}\frac{p}{R}^{3} - \frac{17}{48}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{2736}\frac{p}{R}^{15} + \frac{1}{672}\frac{p}{R}^{12} - \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{23}{3456}\frac{p}{R}^{7} + \frac{1}{252}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} + \frac{11}{1728}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{12}\frac{p}{R}^{2}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{360}\frac{p}{R}^{9} + \frac{1}{1344}\frac{p}{R}^{8} + \frac{5}{1152}\frac{p}{R}^{7} - \frac{1}{576}\frac{p}{R}^{6} + \frac{1}{320}\frac{p}{R}^{4} - \frac{1}{16}\frac{p}{R}^{3} + \frac{1}{16}\frac{p}{R}^{2}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{5}{1152}\frac{p}{R}^{7} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{32}\frac{p}{R}^{3}$
(1,0,0,1)	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2880}\frac{p}{R}^{16} - \frac{61}{114912}\frac{p}{R}^{15} + \frac{1}{672}\frac{p}{R}^{12} - \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{480}\frac{p}{R}^{9} + \frac{67}{14784}\frac{p}{R}^{8} + \frac{5}{864}\frac{p}{R}^{7} - \frac{11}{336}\frac{p}{R}^{5} - \frac{19}{720}\frac{p}{R}^{4} + \frac{1}{12}\frac{p}{R}^{2}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{2736}\frac{p}{R}^{15} - \frac{1}{672}\frac{p}{R}^{12} + \frac{1}{4224}\frac{p}{R}^{11} - \frac{5}{864}\frac{p}{R}^{7} + \frac{1}{84}\frac{p}{R}^{5}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{1344}\frac{p}{R}^{8} + \frac{1}{480}\frac{p}{R}^{9} - \frac{1}{288}\frac{p}{R}^{7} + \frac{1}{80}\frac{p}{R}^{4}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{288}\frac{p}{R}^{7}$

(1,0,1,0)	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{61}{114912}\frac{p}{R}^{15} - \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{67}{14784}\frac{p}{R}^{8} + \frac{11}{3456}\frac{p}{R}^{7} + \frac{1}{448}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} - \frac{5}{216}\frac{p}{R}^{4} + \frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2736}\frac{p}{R}^{15} + \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{11}{3456}\frac{p}{R}^{7} - \frac{1}{252}\frac{p}{R}^{6} + \frac{1}{108}\frac{p}{R}^{4}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{1440}\frac{p}{R}^{9} - \frac{1}{1344}\frac{p}{R}^{8} - \frac{1}{1152}\frac{p}{R}^{7} + \frac{1}{576}\frac{p}{R}^{6}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{4224}\frac{p}{R}^{11} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1152}\frac{p}{R}^{7}$
(1,0,1,1)	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{61}{114912}\frac{p}{R}^{15} + \frac{1}{4224}\frac{p}{R}^{11} - \frac{67}{14784}\frac{p}{R}^{8} - \frac{1}{432}\frac{p}{R}^{7} + \frac{1}{72}\frac{p}{R}^{4}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2736}\frac{p}{R}^{15} - \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{432}\frac{p}{R}^{7}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{4224}\frac{p}{R}^{11} + \frac{1}{1344}\frac{p}{R}^{8}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{4224}\frac{p}{R}^{11}$
(1,1,0,0)	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2880}\frac{p}{R}^{16} - \frac{61}{114912}\frac{p}{R}^{15} + \frac{1}{672}\frac{p}{R}^{12} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{360}\frac{p}{R}^{9} + \frac{1}{264}\frac{p}{R}^{8} + \frac{1}{252}\frac{p}{R}^{6} - \frac{11}{336}\frac{p}{R}^{5} + \frac{1}{64}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{8}\frac{p}{R}^{2}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{2736}\frac{p}{R}^{15} - \frac{1}{672}\frac{p}{R}^{12} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{252}\frac{p}{R}^{6} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{24}\frac{p}{R}^{3}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{16} + \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{360}\frac{p}{R}^{9} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{32}\frac{p}{R}^{3}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{64}\frac{p}{R}^{4}$
(1,1,0,1)	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{16} + \frac{61}{114912}\frac{p}{R}^{15} - \frac{1}{672}\frac{p}{R}^{12} - \frac{1}{480}\frac{p}{R}^{9} - \frac{1}{264}\frac{p}{R}^{8} + \frac{1}{48}\frac{p}{R}^{5}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{2736}\frac{p}{R}^{15} + \frac{1}{672}\frac{p}{R}^{12}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2880}\frac{p}{R}^{16} - \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{480}\frac{p}{R}^{9}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{16}$
(1,1,1,0)	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{61}{114912}\frac{p}{R}^{15} + \frac{1}{2880}\frac{p}{R}^{10} - \frac{1}{1440}\frac{p}{R}^{9} - \frac{1}{264}\frac{p}{R}^{8} - \frac{1}{252}\frac{p}{R}^{6} + \frac{1}{84}\frac{p}{R}^{5}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{2736}\frac{p}{R}^{15} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{252}\frac{p}{R}^{6}$	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{2880}\frac{p}{R}^{10} + \frac{1}{1440}\frac{p}{R}^{9}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2880}\frac{p}{R}^{10}$
(1,1,1,1)	$\frac{1}{27456}\frac{p}{R}^{22} - \frac{61}{114912}\frac{p}{R}^{15} + \frac{1}{264}\frac{p}{R}^{8}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{2736}\frac{p}{R}^{15}$	$-\frac{1}{27456}\frac{p}{R}^{22} + \frac{1}{6048}\frac{p}{R}^{15}$	$\frac{1}{27456}\frac{p}{R}^{22}$

Table D.5: Probability of the 6 consecutive operations $X_{M_i}, X_{S_i}, X_{M_{i-1}}, X_{S_{i-1}}, X_{M_{i-2}}, X_{S_{i-2}}$ for $k_i \oplus k_{i-1} = 1$ and $k_{i-1} \oplus k_{i-2} = 0$ (case (c))

$(x_{M_i}, x_{S_i}, x_{M_{i-1}}, x_{S_{i-1}})$	$(x_{M_{i-2}}, x_{S_{i-2}})$ (0,0)	(0,1)	(1,0)	(1,1)
(0,0,0,0)	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{169}{514080}\frac{p}{R}^{15} - \frac{1}{12960}\frac{p}{R}^{13} + \frac{73}{65520}\frac{p}{R}^{12} + \frac{1711}{1029600}\frac{p}{R}^{11} - \frac{1}{960}\frac{p}{R}^{10} + \frac{37}{6600}\frac{p}{R}^{9} - \frac{1277}{3326400}\frac{p}{R}^{8} - \frac{593}{33600}\frac{p}{R}^{6} - \frac{229}{3360}\frac{p}{R}^{5} + \frac{659}{20160}\frac{p}{R}^{4} - \frac{13}{720}\frac{p}{R}^{3} + \frac{131}{144}\frac{p}{R}^{2} - \frac{7}{4}\frac{p}{R} + 1$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{6120}\frac{p}{R}^{15} + \frac{1}{12960}\frac{p}{R}^{13} - \frac{1}{3120}\frac{p}{R}^{12} - \frac{1711}{1029600}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{10} - \frac{7}{2475}\frac{p}{R}^{9} + \frac{1}{2700}\frac{p}{R}^{8} - \frac{7}{3600}\frac{p}{R}^{6} + \frac{79}{3360}\frac{p}{R}^{5} - \frac{11}{960}\frac{p}{R}^{4} + \frac{41}{288}\frac{p}{R}^{3} - \frac{31}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{12960}\frac{p}{R}^{13} - \frac{1}{1260}\frac{p}{R}^{12} - \frac{47}{79200}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{10} - \frac{1}{198}\frac{p}{R}^{9} + \frac{227}{221760}\frac{p}{R}^{8} + \frac{1079}{100800}\frac{p}{R}^{6} + \frac{19}{3360}\frac{p}{R}^{5} + \frac{7}{320}\frac{p}{R}^{4} + \frac{19}{720}\frac{p}{R}^{3} - \frac{13}{48}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{12960}\frac{p}{R}^{13} + \frac{47}{79200}\frac{p}{R}^{11} - \frac{1}{960}\frac{p}{R}^{10} + \frac{1}{440}\frac{p}{R}^{9} + \frac{1}{360}\frac{p}{R}^{8} - \frac{1}{200}\frac{p}{R}^{6} + \frac{1}{160}\frac{p}{R}^{5} - \frac{3}{320}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{8}\frac{p}{R}^{2}$
(0,0,0,1)	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{169}{514080}\frac{p}{R}^{15} + \frac{1}{12960}\frac{p}{R}^{13} - \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{936}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{10} - \frac{389}{79200}\frac{p}{R}^{9} - \frac{613}{221760}\frac{p}{R}^{8} + \frac{1}{144}\frac{p}{R}^{6} + \frac{9}{160}\frac{p}{R}^{5} + \frac{19}{5040}\frac{p}{R}^{4} - \frac{1}{45}\frac{p}{R}^{3} - \frac{23}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{6120}\frac{p}{R}^{15} - \frac{1}{12960}\frac{p}{R}^{13} + \frac{1}{936}\frac{p}{R}^{11} - \frac{1}{960}\frac{p}{R}^{10} + \frac{7}{2475}\frac{p}{R}^{9} + \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{144}\frac{p}{R}^{6} - \frac{79}{3360}\frac{p}{R}^{5} - \frac{1}{40}\frac{p}{R}^{4} - \frac{1}{36}\frac{p}{R}^{3} + \frac{1}{9}\frac{p}{R}^{2}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{12960}\frac{p}{R}^{13} + \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{960}\frac{p}{R}^{10} + \frac{23}{5280}\frac{p}{R}^{9} - \frac{227}{221760}\frac{p}{R}^{8} - \frac{19}{3360}\frac{p}{R}^{5} - \frac{3}{80}\frac{p}{R}^{4} + \frac{1}{20}\frac{p}{R}^{3}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{12960}\frac{p}{R}^{13} + \frac{1}{960}\frac{p}{R}^{10} - \frac{1}{440}\frac{p}{R}^{9} - \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{160}\frac{p}{R}^{5} + \frac{1}{40}\frac{p}{R}^{4}$
(0,0,1,0)	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{169}{514080}\frac{p}{R}^{15} + \frac{1}{12960}\frac{p}{R}^{13} - \frac{73}{65520}\frac{p}{R}^{12} - \frac{1711}{1029600}\frac{p}{R}^{11} - \frac{1}{800}\frac{p}{R}^{9} - \frac{13843}{3326400}\frac{p}{R}^{8} + \frac{2479}{100800}\frac{p}{R}^{6} + \frac{19}{504}\frac{p}{R}^{5} - \frac{1}{144}\frac{p}{R}^{4} - \frac{19}{180}\frac{p}{R}^{3} - \frac{1}{8}\frac{p}{R}^{2} + \frac{1}{4}\frac{p}{R}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{6120}\frac{p}{R}^{15} - \frac{1}{12960}\frac{p}{R}^{13} + \frac{1}{3120}\frac{p}{R}^{12} + \frac{1711}{1029600}\frac{p}{R}^{11} + \frac{1}{1800}\frac{p}{R}^{9} - \frac{1}{2700}\frac{p}{R}^{8} - \frac{1}{200}\frac{p}{R}^{6} - \frac{1}{72}\frac{p}{R}^{5} + \frac{1}{48}\frac{p}{R}^{4}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{12960}\frac{p}{R}^{13} + \frac{1}{1260}\frac{p}{R}^{12} + \frac{47}{79200}\frac{p}{R}^{11} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{71}{20160}\frac{p}{R}^{8} - \frac{1079}{100800}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} + \frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{12960}\frac{p}{R}^{13} - \frac{47}{79200}\frac{p}{R}^{11} - \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{200}\frac{p}{R}^{6}$
(0,0,1,1)	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{169}{514080}\frac{p}{R}^{15} + \frac{1}{12960}\frac{p}{R}^{13} + \frac{1}{936}\frac{p}{R}^{11} + \frac{1}{1800}\frac{p}{R}^{9} + \frac{1621}{221760}\frac{p}{R}^{8} - \frac{1}{72}\frac{p}{R}^{6} - \frac{13}{504}\frac{p}{R}^{5} - \frac{1}{72}\frac{p}{R}^{4} + \frac{1}{8}\frac{p}{R}^{2}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{6120}\frac{p}{R}^{15} + \frac{1}{12960}\frac{p}{R}^{13} - \frac{1}{936}\frac{p}{R}^{11} - \frac{1}{1800}\frac{p}{R}^{9} - \frac{1}{360}\frac{p}{R}^{8} + \frac{1}{72}\frac{p}{R}^{5}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{12960}\frac{p}{R}^{13} - \frac{1}{1260}\frac{p}{R}^{12} - \frac{71}{20160}\frac{p}{R}^{8} + \frac{1}{84}\frac{p}{R}^{5}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{12960}\frac{p}{R}^{13} + \frac{1}{360}\frac{p}{R}^{8}$
(0,1,0,0)	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{169}{514080}\frac{p}{R}^{15} - \frac{73}{65520}\frac{p}{R}^{12} - \frac{49}{32175}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{10} - \frac{1}{198}\frac{p}{R}^{9} + \frac{73}{23760}\frac{p}{R}^{8} + \frac{11}{1008}\frac{p}{R}^{6} + \frac{7}{144}\frac{p}{R}^{5} - \frac{85}{1344}\frac{p}{R}^{4} + \frac{31}{480}\frac{p}{R}^{3} - \frac{23}{72}\frac{p}{R}^{2} + \frac{1}{3}\frac{p}{R}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{6120}\frac{p}{R}^{15} + \frac{1}{3120}\frac{p}{R}^{12} + \frac{49}{32175}\frac{p}{R}^{11} - \frac{1}{960}\frac{p}{R}^{10} + \frac{1}{440}\frac{p}{R}^{9} - \frac{1}{432}\frac{p}{R}^{8} + \frac{1}{144}\frac{p}{R}^{6} - \frac{1}{63}\frac{p}{R}^{5} + \frac{1}{64}\frac{p}{R}^{4} - \frac{1}{12}\frac{p}{R}^{3} + \frac{1}{9}\frac{p}{R}^{2}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{2200}\frac{p}{R}^{11} - \frac{1}{960}\frac{p}{R}^{10} + \frac{1}{198}\frac{p}{R}^{9} - \frac{1}{220}\frac{p}{R}^{8} - \frac{1}{252}\frac{p}{R}^{6} + \frac{1}{64}\frac{p}{R}^{4} - \frac{7}{96}\frac{p}{R}^{3} + \frac{1}{12}\frac{p}{R}^{2}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} - \frac{1}{2200}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{10} - \frac{1}{440}\frac{p}{R}^{9} - \frac{1}{64}\frac{p}{R}^{4} + \frac{1}{24}\frac{p}{R}^{3}$
(0,1,0,1)	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{169}{514080}\frac{p}{R}^{15} + \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{936}\frac{p}{R}^{11} - \frac{1}{960}\frac{p}{R}^{10} + \frac{23}{5280}\frac{p}{R}^{9} - \frac{1}{1320}\frac{p}{R}^{8} - \frac{1}{144}\frac{p}{R}^{6} - \frac{37}{1008}\frac{p}{R}^{5} + \frac{1}{21}\frac{p}{R}^{4}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{6120}\frac{p}{R}^{15} - \frac{1}{936}\frac{p}{R}^{11} + \frac{1}{960}\frac{p}{R}^{10} - \frac{1}{440}\frac{p}{R}^{9} - \frac{1}{144}\frac{p}{R}^{6} + \frac{1}{63}\frac{p}{R}^{5}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{1260}\frac{p}{R}^{12} + \frac{1}{960}\frac{p}{R}^{10} - \frac{23}{5280}\frac{p}{R}^{9} + \frac{1}{220}\frac{p}{R}^{8}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{960}\frac{p}{R}^{10} + \frac{1}{440}\frac{p}{R}^{9}$
(0,1,1,0)	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} + \frac{169}{514080}\frac{p}{R}^{15} + \frac{73}{65520}\frac{p}{R}^{12} + \frac{49}{32175}\frac{p}{R}^{11} + \frac{1}{1440}\frac{p}{R}^{9} + \frac{7}{4752}\frac{p}{R}^{8} - \frac{1}{56}\frac{p}{R}^{6} - \frac{1}{84}\frac{p}{R}^{5} + \frac{1}{20}\frac{p}{R}^{3}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{6120}\frac{p}{R}^{15} - \frac{1}{3120}\frac{p}{R}^{12} - \frac{49}{32175}\frac{p}{R}^{11} + \frac{1}{432}\frac{p}{R}^{8}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{6048}\frac{p}{R}^{15} - \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{2200}\frac{p}{R}^{11} - \frac{1}{1440}\frac{p}{R}^{9} + \frac{1}{252}\frac{p}{R}^{6}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{19200}\frac{p}{R}^{16} + \frac{1}{2200}\frac{p}{R}^{11}$
(0,1,1,1)	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17} + \frac{169}{514080}\frac{p}{R}^{15} - \frac{1}{1260}\frac{p}{R}^{12} - \frac{1}{936}\frac{p}{R}^{11} - \frac{1}{264}\frac{p}{R}^{8} + \frac{1}{72}\frac{p}{R}^{6}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{6120}\frac{p}{R}^{15} + \frac{1}{936}\frac{p}{R}^{11}$	$\frac{1}{51840}\frac{p}{R}^{22} - \frac{1}{6840}\frac{p}{R}^{17} - \frac{1}{6048}\frac{p}{R}^{15} + \frac{1}{1260}\frac{p}{R}^{12}$	$-\frac{1}{51840}\frac{p}{R}^{22} + \frac{1}{6840}\frac{p}{R}^{17}$

(1,0,0,0)	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{169}{514080}\frac{p}{R}^{15}+\frac{1}{12960}\frac{p}{R}^{13}-\frac{1}{3120}\frac{p}{R}^{12}-\frac{1}{7200}\frac{p}{R}^{11}+\frac{1}{960}\frac{p}{R}^{10}-\frac{1}{300}\frac{p}{R}^{9}-\frac{1367}{369600}\frac{p}{R}^{8}-\frac{1}{192}\frac{p}{R}^{6}+\frac{89}{3360}\frac{p}{R}^{5}+\frac{31}{2880}\frac{p}{R}^{4}+\frac{5}{48}\frac{p}{R}^{3}-\frac{17}{48}\frac{p}{R}^{2}+\frac{1}{4}\frac{p}{R}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{6120}\frac{p}{R}^{15}-\frac{1}{12960}\frac{p}{R}^{13}+\frac{1}{3120}\frac{p}{R}^{12}+\frac{1}{7200}\frac{p}{R}^{11}-\frac{1}{960}\frac{p}{R}^{10}+\frac{1}{1800}\frac{p}{R}^{9}-\frac{1}{1200}\frac{p}{R}^{8}+\frac{1}{144}\frac{p}{R}^{6}-\frac{1}{160}\frac{p}{R}^{5}+\frac{1}{64}\frac{p}{R}^{4}-\frac{29}{288}\frac{p}{R}^{3}+\frac{1}{12}\frac{p}{R}^{2}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{6048}\frac{p}{R}^{15}-\frac{1}{12960}\frac{p}{R}^{13}+\frac{1}{7200}\frac{p}{R}^{11}-\frac{1}{960}\frac{p}{R}^{10}+\frac{1}{360}\frac{p}{R}^{9}+\frac{1}{1344}\frac{p}{R}^{8}-\frac{1}{576}\frac{p}{R}^{6}+\frac{1}{160}\frac{p}{R}^{5}+\frac{1}{320}\frac{p}{R}^{4}-\frac{1}{16}\frac{p}{R}^{3}+\frac{1}{16}\frac{p}{R}^{2}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{12960}\frac{p}{R}^{13}-\frac{1}{7200}\frac{p}{R}^{11}+\frac{1}{960}\frac{p}{R}^{10}-\frac{1}{160}\frac{p}{R}^{5}-\frac{1}{64}\frac{p}{R}^{4}+\frac{1}{32}\frac{p}{R}^{3}$
(1,0,0,1)	$-\frac{1}{51840}\frac{p}{R}^{22}-\frac{169}{514080}\frac{p}{R}^{15}-\frac{1}{12960}\frac{p}{R}^{13}-\frac{1}{960}\frac{p}{R}^{10}+\frac{19}{7200}\frac{p}{R}^{9}+\frac{67}{14784}\frac{p}{R}^{8}+\frac{1}{144}\frac{p}{R}^{6}-\frac{7}{480}\frac{p}{R}^{5}-\frac{19}{720}\frac{p}{R}^{4}-\frac{1}{36}\frac{p}{R}^{3}+\frac{1}{12}\frac{p}{R}^{2}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{6120}\frac{p}{R}^{15}+\frac{1}{12960}\frac{p}{R}^{13}+\frac{1}{960}\frac{p}{R}^{10}-\frac{1}{1800}\frac{p}{R}^{9}-\frac{1}{144}\frac{p}{R}^{6}-\frac{1}{160}\frac{p}{R}^{5}+\frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{6048}\frac{p}{R}^{15}+\frac{1}{12960}\frac{p}{R}^{13}+\frac{1}{960}\frac{p}{R}^{10}-\frac{1}{480}\frac{p}{R}^{9}-\frac{1}{1344}\frac{p}{R}^{8}-\frac{1}{160}\frac{p}{R}^{5}+\frac{1}{80}\frac{p}{R}^{4}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{12960}\frac{p}{R}^{13}-\frac{1}{960}\frac{p}{R}^{10}+\frac{1}{160}\frac{p}{R}^{5}$
(1,0,1,0)	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{169}{514080}\frac{p}{R}^{15}+\frac{1}{12960}\frac{p}{R}^{13}+\frac{1}{3120}\frac{p}{R}^{12}+\frac{1}{7200}\frac{p}{R}^{11}+\frac{1}{800}\frac{p}{R}^{9}+\frac{1367}{369600}\frac{p}{R}^{8}-\frac{1}{576}\frac{p}{R}^{6}-\frac{1}{84}\frac{p}{R}^{5}-\frac{1}{72}\frac{p}{R}^{4}+\frac{1}{36}\frac{p}{R}^{3}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{6120}\frac{p}{R}^{15}+\frac{1}{12960}\frac{p}{R}^{13}-\frac{1}{3120}\frac{p}{R}^{12}-\frac{1}{7200}\frac{p}{R}^{11}-\frac{1}{1800}\frac{p}{R}^{9}+\frac{1}{1200}\frac{p}{R}^{8}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{6048}\frac{p}{R}^{15}+\frac{1}{12960}\frac{p}{R}^{13}-\frac{1}{7200}\frac{p}{R}^{11}-\frac{1}{1440}\frac{p}{R}^{9}-\frac{1}{1344}\frac{p}{R}^{8}+\frac{1}{576}\frac{p}{R}^{6}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{12960}\frac{p}{R}^{13}+\frac{1}{7200}\frac{p}{R}^{11}$
(1,0,1,1)	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{169}{514080}\frac{p}{R}^{15}+\frac{1}{12960}\frac{p}{R}^{13}-\frac{1}{1800}\frac{p}{R}^{9}-\frac{67}{14784}\frac{p}{R}^{8}+\frac{1}{72}\frac{p}{R}^{4}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{6120}\frac{p}{R}^{15}-\frac{1}{12960}\frac{p}{R}^{13}+\frac{1}{1800}\frac{p}{R}^{9}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{6048}\frac{p}{R}^{15}-\frac{1}{12960}\frac{p}{R}^{13}+\frac{1}{1344}\frac{p}{R}^{8}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{12960}\frac{p}{R}^{13}$
(1,1,0,0)	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{169}{514080}\frac{p}{R}^{15}+\frac{1}{3120}\frac{p}{R}^{12}-\frac{1}{960}\frac{p}{R}^{10}+\frac{1}{360}\frac{p}{R}^{9}+\frac{1}{264}\frac{p}{R}^{8}+\frac{1}{144}\frac{p}{R}^{6}-\frac{11}{336}\frac{p}{R}^{5}+\frac{1}{64}\frac{p}{R}^{4}-\frac{7}{96}\frac{p}{R}^{3}+\frac{1}{8}\frac{p}{R}^{2}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{6120}\frac{p}{R}^{15}-\frac{1}{3120}\frac{p}{R}^{12}+\frac{1}{960}\frac{p}{R}^{10}-\frac{1}{144}\frac{p}{R}^{6}-\frac{1}{64}\frac{p}{R}^{4}+\frac{1}{24}\frac{p}{R}^{3}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{1}{6048}\frac{p}{R}^{15}+\frac{1}{960}\frac{p}{R}^{10}-\frac{1}{360}\frac{p}{R}^{9}-\frac{1}{64}\frac{p}{R}^{4}+\frac{1}{32}\frac{p}{R}^{3}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{960}\frac{p}{R}^{10}+\frac{1}{64}\frac{p}{R}^{4}$
(1,1,0,1)	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{169}{514080}\frac{p}{R}^{15}+\frac{1}{960}\frac{p}{R}^{10}-\frac{1}{480}\frac{p}{R}^{9}-\frac{1}{264}\frac{p}{R}^{8}-\frac{1}{144}\frac{p}{R}^{6}+\frac{1}{48}\frac{p}{R}^{5}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{6120}\frac{p}{R}^{15}-\frac{1}{960}\frac{p}{R}^{10}+\frac{1}{144}\frac{p}{R}^{6}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{6048}\frac{p}{R}^{15}-\frac{1}{960}\frac{p}{R}^{10}+\frac{1}{480}\frac{p}{R}^{9}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{960}\frac{p}{R}^{10}$
(1,1,1,0)	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}+\frac{169}{514080}\frac{p}{R}^{15}-\frac{1}{3120}\frac{p}{R}^{12}-\frac{1}{1440}\frac{p}{R}^{9}-\frac{1}{264}\frac{p}{R}^{8}+\frac{1}{84}\frac{p}{R}^{5}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{6120}\frac{p}{R}^{15}+\frac{1}{3120}\frac{p}{R}^{12}$	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{1}{19200}\frac{p}{R}^{16}-\frac{1}{6048}\frac{p}{R}^{15}+\frac{1}{1440}\frac{p}{R}^{9}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{19200}\frac{p}{R}^{16}$
(1,1,1,1)	$\frac{1}{51840}\frac{p}{R}^{22}-\frac{169}{514080}\frac{p}{R}^{15}+\frac{1}{264}\frac{p}{R}^{8}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{6120}\frac{p}{R}^{15}$	$-\frac{1}{51840}\frac{p}{R}^{22}+\frac{1}{6048}\frac{p}{R}^{15}$	$\frac{1}{51840}\frac{p}{R}^{22}$

Table D.6: Probability of the 6 consecutive operations $X_{M_i}, X_{S_i}, X_{M_{i-1}}, X_{S_{i-1}}, X_{M_{i-2}}, X_{S_{i-2}}$ for $k_i \oplus k_{i-1} = 1$ and $k_{i-1} \oplus k_{i-2} = 1$ (case (d))

Appendix E

Acronyms

brainpoolP256r1	Standard curve by the BSI
BSC	Binary symmetric Channel
BSI	Bundesamt für Sicherheit in der Informationstechnik
CPA	Correlation Power Attacks
CRT	Chinese remainder theorem
DAA	"Double-and-add-always" algorithm
DPA	Differential Power Attacks
DSCA	Differential Side-channel analysis
ECADD	Addition operation on elliptic curve
ECC	Elliptic Curve Cryptography
ECDBL	Doubling operation on elliptic curve
ECDH	Elliptic Curve Diffie Hellman protocol
ECDSA	Elliptic Curve Digital Standard Signature
ECSM	Elliptic Curve Scalar Multiplication
Ed25519	Name of Bernstein and Lange curve
EM	Electromagnetic emanation
ERA	Extra-reduction analysis
i.i.d.	independent and identically distributed
IOTA	Improvement of Online Template Attack
L2R	Left-to-Right
lsb	less significant bit
LSW	less significant word
mbedTLS	ARMmbed cryptography library available in https://tls.mbed.org/
ML	"Montgomery Ladder " algorithm
MMM	Montgomery modular multiplication
msb	most significant bit
MSW	most significant word
NaCl	Cryptography Library by Bernstein
NIST	National Institute of Standards and Technology
OpenSSL	Cryptography and SSL/TLS library available in https://www.openssl.org/

OTA	Online Template Attack
P-256	Standard curve by the NIST
p.d.f	product density function
PolarSSL	oldest version of mbedTLS
R2L	Right-to-Left
RNG	Random generator function
RSA	Rivest Shamir Adleman cryptography protocol
RSA-CRT	RSA with Chinese Remainder Theorem
RSA-SFM	RSA without Chinese Remainder Theorem
SCA	Side-Channel analysis
SEMA	Simple ElectroMagnetic Attacks
SMA	"Square-and-multiply-always" algorithm
SPA	Simple power analysis
SSCA	Simple side channel analysis

Bibliography

[ABF+02] Christian Aumüller, Peter Bier, Wieland Fischer, Peter Hofreiter, and Jean-Pierre Seifert. Fault attacks on RSA with CRT: concrete results and practical countermeasures. In Jr. et al. [JKP03], pages 260–275.

[ANS96] American National Standards Institute ANSI. Public key cryptography for the financial services industry: Agreement of symmetric algorithm keys using diffie-hellman. Technical report, ANSI-X9.42, 1996.

[ANS99] American National Standards Institute ANSI. Public key cryptography for the financial services industry- the elliptic curve digital signature algorithm (ECDSA). Technical report, ANSI-X9.63, 1999.

[AS08] Onur Aciiçmez and Werner Schindler. A vulnerability in RSA implementations due to instruction cache analysis and its demonstration on openssl. In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, volume 4964 of *Lecture Notes in Computer Science*, pages 256–273. Springer, 2008.

[ASK05] Onur Aciiçmez, Werner Schindler, and Çetin Kaya Koç. Improving Brumley and Boneh timing attack on unprotected SSL implementations. In Vijay Atluri, Catherine Meadows, and Ari Juels, editors, *Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, Alexandria, VA, USA, November 7-11, 2005*, pages 139–146. ACM, 2005.

[AT03] Toru Akishita and Tsuyoshi Takagi. Zero-value point attacks on elliptic curve cryptosystem. In Colin Boyd and Wenbo Mao, editors, *Information Security, 6th International Conference, ISC 2003, Bristol, UK, October 1-3, 2003, Proceedings*, volume 2851 of *Lecture Notes in Computer Science*, pages 218–233. Springer, 2003.

[Bat14] Alberto Battistello. Common points on elliptic curves: The achilles' heel of fault attack countermeasures. In Prouff [Pro14], pages 69–81.

[BBB+13] Pierre Belgarric, Shivam Bhasin, Nicolas Bruneau, Jean-Luc Danger, Nicolas Debande, Sylvain Guilley, Annelie Heuser, Zakaria Najm, and Olivier Rioul. Time-Frequency Analysis for Second-Order Attacks. In Francillon and Rohatgi [FR14], pages 108–122.

[BBJ+08] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. In Serge Vaudenay, editor, *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings*, volume 5023 of *Lecture Notes in Computer Science*, pages 389–405. Springer, 2008.

[BCG10] Alexandre Berzati, Cécile Canovas-Dumas, and Louis Goubin. Public key perturbation of randomized RSA implementations. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 306–319. Springer, 2010.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[BCP+14] Lejla Batina, Lukasz Chmielewski, Louiza Papachristodoulou, Peter Schwabe, and Michael Tunstall. Online template attacks. In Willi Meier and Debdeep Mukhopadhyay, editors, *Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings*, volume 8885 of *Lecture Notes in Computer Science*, pages 21–36. Springer, 2014.

[BDF+14] Gilles Barthe, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, and Jean-Christophe Zapalowicz. Synthesis of Fault Attacks on Cryptographic Implementations. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 1016–1027. ACM, 2014.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In Walter Fumy, editor, *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997.

[BDL+12] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *J. Cryptographic Engineering*, 2(2):77–89, 2012.

[BdSG+14] Johannes Blömer, Ricardo Gomes da Silva, Peter Günther, Juliane Krämer, and Jean-Pierre Seifert. A practical second-order fault attack against a real-world pairing implementation. In Tria and Choi [TC14], pages 123–136.

[Ben14] Josh Benaloh, editor. *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA,*

February 25-28, 2014. Proceedings, volume 8366 of *Lecture Notes in Computer Science*. Springer, 2014.

[BGL14] Johannes Blömer, Peter Günther, and Gennadij Liske. Tampering attacks in pairing-based cryptography. In Tria and Choi [TC14], pages 1–7.

[BJP+15] Aurélie Bauer, Éliane Jaulmes, Emmanuel Prouff, Jean-René Reinhard, and Justine Wild. Horizontal collision correlation attack on elliptic curves - - extended version -. *Cryptography and Communications*, 7(1):91–119, 2015.

[BJPW13] Aurélie Bauer, Éliane Jaulmes, Emmanuel Prouff, and Justine Wild. Horizontal and vertical side-channel attacks against secure RSA implementations. In Ed Dawson, editor, *Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013, San Francisco,CA, USA, February 25-March 1, 2013. Proceedings*, volume 7779 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2013.

[BL] Daniel J. Bernstein and Tanja Lange. Explicit formulas database. http://www.hyperelliptic.org/EFD/.

[BL07] Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 29–50. Springer, 2007.

[BLS12] Daniel J. Bernstein, Tanja Lange, and Peter Schwabe. The security impact of a new cryptographic library. In Alejandro Hevia and Gregory Neven, editors, *Progress in Cryptology - LATINCRYPT 2012 - 2nd International Conference on Cryptology and Information Security in Latin America, Santiago, Chile, October 7-10, 2012. Proceedings*, volume 7533 of *Lecture Notes in Computer Science*, pages 159–176. Springer, 2012.

[BMM00] Ingrid Biehl, Bernd Meyer, and Volker Müller. Differential fault attacks on elliptic curve cryptosystems. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 131–146. Springer, 2000.

[BNP07] Arnaud Boscher, Robert Naciri, and Emmanuel Prouff. CRT RSA algorithm protected against fault attacks. In Damien Sauveron, Constantinos Markantonakis, Angelos Bilas, and Jean-Jacques Quisquater, editors, *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems, First IFIP TC6 / WG 8.8 / WG 11.2 International Workshop, WISTP 2007, Heraklion, Crete, Greece, May 9-11, 2007, Proceedings*, volume 4462 of *Lecture Notes in Computer Science*, pages 229–243. Springer, 2007.

[BOS03] Johannes Blömer, Martin Otto, and Jean-Pierre Seifert. A new CRT-RSA algorithm secure against bellcore attacks. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003, Washington, DC, USA, October 27-30, 2003*, pages 311–320. ACM, 2003.

[BOS06] Johannes Blömer, Martin Otto, and Jean-Pierre Seifert. Sign change fault attacks on elliptic curve cryptosystems. In Luca Breveglieri, Israel Koren, David Naccache, and Jean-Pierre Seifert, editors, *Fault Diagnosis and Tolerance in Cryptography, Third International Workshop, FDTC 2006, Yokohama, Japan, October 10, 2006, Proceedings*, volume 4236 of *Lecture Notes in Computer Science*, pages 36–52. Springer, 2006.

[BSI10] BSI. RFC 5639 - Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. Technical report, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2010.

[BSI17] BSI - Technische Richtlinie. BSI TR-02102-1 (version 2017-01): Kryptographische Verfahren: Empfehlungen und Schlüssellangen, February 8 2017. Section 3.5, entitled "RSA Schlüsselgenerierung".

[BT11] Billy Bob Brumley and Nicola Tuveri. Remote timing attacks are still practical. In Vijay Atluri and Claudia Díaz, editors, *Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings*, volume 6879 of *Lecture Notes in Computer Science*, pages 355–371. Springer, 2011.

[BV07] Yoo-Jin Baek and Ihor Vasyltsov. How to prevent DPA and fault attack in a unified way for ECC scalar multiplication - ring extension method. In Ed Dawson and Duncan S. Wong, editors, *Information Security Practice and Experience, Third International Conference, ISPEC 2007, Hong Kong, China, May 7-9, 2007, Proceedings*, volume 4464 of *Lecture Notes in Computer Science*, pages 225–237. Springer, 2007.

[CFG+10] Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. Horizontal correlation analysis on exponentiation. In Miguel Soriano, Sihan Qing, and Javier López, editors, *Information and Communications Security - 12th International Conference, ICICS 2010, Barcelona, Spain, December 15-17, 2010. Proceedings*, volume 6476 of *Lecture Notes in Computer Science*, pages 46–61. Springer, 2010.

[CFG+12] Christophe Clavier, Benoit Feix, Georges Gagnerot, Christophe Giraud, Mylène Roussellet, and Vincent Verneuil. ROSETTA for single trace analysis. In Steven D. Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, volume 7668 of *Lecture Notes in Computer Science*, pages 140–155. Springer, 2012.

[CFR10] Jean-Christophe Courrège, Benoit Feix, and Mylène Roussellet. Simple power analysis on exponentiation revisited. In Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny, editors, *Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, Passau, Germany, April 14-16, 2010. Proceedings*, volume 6035 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2010.

[CJ05] Mathieu Ciet and Marc Joye. Practical fault countermeasures for chinese remaindering based RSA. In *Fault Diagnosis and Tolerance in Cryptography*, pages 124–131, Friday September 2nd 2005. Edinburgh, Scotland.

[Cla07] Christophe Clavier. Secret external encodings do not prevent transient fault analysis. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 181–194. Springer, 2007.

[Cor99] Jean-Sébastien Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In Koç and Paar [KP99], pages 292–302.

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Jr. et al. [JKP03], pages 13–28.

[DGD+16] Margaux Dugardin, Sylvain Guilley, Jean-Luc Danger, Zakaria Najm, and Olivier Rioul. Correlated extra-reductions defeat blinded regular exponentiation. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2016.

[DGM+16] Margaux Dugardin, Sylvain Guilley, Martin Moreau, Zakaria Najm, and Pablo Rauzy. Using Modular Extension to Provably Protect Edwards Curves Against Fault Attacks. In *PROOFS: Security Proofs for Embedded Systems 2016*, San, United States, August 2016.

[DGM+17] Margaux Dugardin, Sylvain Guilley, Martin Moreau, Zakaria Najm, and Pablo Rauzy. Using modular extension to provably protect edwards curves against fault attacks. *Journal of Cryptographic Engineering*, 2017.

[DGRS09] Emmanuelle Dottax, Christophe Giraud, Matthieu Rivain, and Yannick Sierra. On second-order fault analysis resistance for CRT-RSA implementations. In Olivier Markowitch, Angelos Bilas, Jaap-Henk Hoepman, Chris J. Mitchell, and Jean-Jacques Quisquater, editors, *Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks, Third IFIP WG 11.2 International Workshop, WISTP 2009, Brussels, Belgium, September 1-4, 2009, Proceedings*, volume 5746 of *Lecture Notes in Computer Science*, pages 68–83. Springer, 2009.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[DPN+16] Margaux Dugardin, Louiza Papachristodoulou, Zakaria Najm, Lejla Batina, Jean-Luc Danger, and Sylvain Guilley. Dismantling real-world ECC with horizontal and vertical template attacks. In François-Xavier Standaert and Elisabeth Oswald, editors, *Constructive Side-Channel Analysis and Secure Design - 7th International Workshop, COSADE 2016, Graz, Austria, April 14-15, 2016, Revised Selected Papers*, volume 9689 of *Lecture Notes in Computer Science*, pages 88–108. Springer, 2016.

[Edw07] Edwards, Harold M. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44:393–422, 9 April 2007. DOI: 10.1090/s0273-0979-07-01153-6, ISSN 0002-9904.

[Eul41] Leonhard Euler. Theorematum quorundam ad numeros primos spectantium demonstratio. *Commentarii academiae scientiarum Petropolitanae*, 8:pp. 141–146, 1741.

[Eul63] Leonhard Euler. Theoremata arithmetica nova methodo demonstrata. *Novi Commentarii academiae scientiarum Petropolitanae*, 8:pp. 74–104, 1763.

[FR14] Aurélien Francillon and Pankaj Rohatgi, editors. *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *LNCS*. Springer, 2014.

[FV03] Pierre-Alain Fouque and Frédéric Valette. The doubling attack - *Why Upwards Is Better than Downwards*. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 269–280. Springer, 2003.

[Gir06] Christophe Giraud. An RSA implementation resistant to fault attacks and to simple power analysis. *IEEE Trans. Computers*, 55(9):1116–1120, 2006.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Koç et al. [KNP01], pages 251–261.

[GV12] Aurore Guillevic and Damien Vergnaud. Genus 2 hyperelliptic curve families with explicit jacobian order evaluation and pairing-friendly constructions. In Michel Abdalla and Tanja Lange, editors, *Pairing-Based Cryptography - Pairing 2012 - 5th International Conference, Cologne, Germany, May 16-18, 2012, Revised Selected Papers*, volume 7708 of *Lecture Notes in Computer Science*, pages 234–253. Springer, 2012.

[Has88] Johan Hastad. Solving simultaneous modular equations of low degree. *SIAM Journal on Computing*, 17(2):336–341, 1988.

[HIM+13] Johann Heyszl, Andreas Ibing, Stefan Mangard, Fabrizio De Santis, and Georg Sigl. Clustering algorithms for non-profiled single-execution attacks on exponentiations. In Francillon and Rohatgi [FR14], pages 79–93.

[HKT15] Neil Hanley, HeeSeok Kim, and Michael Tunstall. Exploiting collisions in addition chain-based exponentiation algorithms using a single trace. In Kaisa Nyberg, editor, *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, volume 9048 of *Lecture Notes in Computer Science*, pages 431–448. Springer, 2015.

[HS13] Michael Hutter and Peter Schwabe. Nacl on 8-bit AVR microcontrollers. In Amr Youssef, Abderrahmane Nitaj, and Aboul Ella Hassanien, editors, *Progress in Cryptology - AFRICACRYPT 2013, 6th International Conference on Cryptology in Africa, Cairo, Egypt, June 22-24, 2013. Proceedings*, volume 7918 of *Lecture Notes in Computer Science*, pages 156–172. Springer, 2013.

[HTM11] Neil Hanley, Michael Tunstall, and William P. Marnane. Using templates to distinguish multiplications from squaring operations. *Int. J. Inf. Sec.*, 10(4):255–266, 2011.

[Ins] Riscure Inspector.

[ISO17] ISO/IEC JTC 1/SC 27/WG 3. ISO/IEC CD 20085-1:2017(E). Information technology - Security techniques — Test tool requirements and test tool calibration methods for use in testing non-invasive attack mitigation techniques in cryptographic modules — Part 1: Test tools and techniques, January 25 2017.

[JKP03] Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2003.

[Joy03] Marc Joye. *Elliptic curve cryptosystems and Side Channel Analysis*, volume 4, pages 17–21. ST J. Syst. Res., 2003.

[Joy10] Marc Joye. Fault-resistant calculations on elliptic curves, September 15 2010. EP Patent App. EP20,100,155,001 ; http://www.google.com/patents/EP2228716A1?cl=en.

[Joy13] Marc Joye. Elliptic curve cryptosystems in the presence of faults. In Wieland Fischer and Jörn-Marc Schmidt, editors, *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013*, page 73. IEEE Computer Society, 2013.

[JPY01] Marc Joye, Pascal Paillier, and Sung-Ming Yen. Secure evaluation of modular functions. In R.J. Hwang and C.K. Wu, editors, *International Workshop on Cryptology and Network Security*, pages 227–229, September, 26-28 2001. http://joye.site88.net/papers/JPY01dfa.pdf, Taipei, Taiwan.

[JT01] Marc Joye and Christophe Tymen. Protections against differential analysis for elliptic curve cryptography. In Koç et al. [KNP01], pages 377–390.

[JT12] Marc Joye and Michael Tunstall, editors. *Fault Analysis in Cryptography*. Information Security and Cryptography. Springer, 2012. ISBN: 978-3-642-29655-0; DOI: 10.1007/978-3-642-29656-7.

[JY02] Marc Joye and Sung-Ming Yen. The montgomery powering ladder. In Jr. et al. [JKP03], pages 291–302.

[KFSV11] Dusko Karaklajic, Junfeng Fan, Jörn-Marc Schmidt, and Ingrid Verbauwhede. Low-cost fault detection method for ECC using montgomery powering ladder. In *Design, Automation and Test in Europe, DATE 2011, Grenoble, France, March 14-18, 2011*, pages 1016–1021. IEEE, 2011.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[KKHH11] Sung-Kyoung Kim, Tae Hyun Kim, Dong-Guk Han, and Seokhie Hong. An efficient CRT-RSA algorithm secure against power and fault attacks. *Journal of Systems and Software*, 84(10):1660–1669, 2011.

[KNP01] Çetin Kaya Koç, David Naccache, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*. Springer, 2001.

[Kob87] Neal Koblitz. *Elliptic curve cryptosystems*, volume 48, pages 203–209. Mathematics of Computation, 1987.

[Koc96] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[Koc98] Paul C. Kocher. On certificate revocation and validation. In Rafael Hirschfeld, editor, *Financial Cryptography, Second International Conference, FC'98, Anguilla, British West Indies, February 23-25, 1998, Proceedings*, volume 1465 of *Lecture Notes in Computer Science*, pages 172–177. Springer, 1998.

[KP99] Çetin Kaya Koç and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*. Springer, 1999.

[Leo06] V.K. Leont'ev. Roots of random polynomials over a finite field. *Mathematical Notes*, 80(1-2):300–304, 2006.

[LMV+13] Liran Lerman, Stephane Fernandes Medeiros, Nikita Veshchikov, Cédric Meuter, Gianluca Bontempi, and Olivier Markowitch. Semi-supervised template attack. In Emmanuel Prouff, editor, *Constructive Side-Channel Analysis and Secure Design - 4th International Workshop, COSADE 2013, Paris, France, March 6-8, 2013, Revised Selected Papers*, volume 7864 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2013.

[LPM+14] Ronan Lashermes, Marie Paindavoine, Nadia El Mrabet, Jacques J. A. Fournier, and Louis Goubin. Practical validation of several fault attacks against the miller algorithm. In Tria and Choi [TC14], pages 115–122.

[LRT14] Duc-Phong Le, Matthieu Rivain, and Chik How Tan. On double exponentiation for securing RSA against fault analysis. In Benaloh [Ben14], pages 152–168.

[MBO+05] Elke De Mulder, Pieter Buysschaert, Sıddıka Berna Örs, Peter Delmotte, Bart Preneel, Guy Vandenbosch, and Ingrid Verbauwhede. Electromagnetic Analysis Attack on an FPGA Implementation of an Elliptic Curve Cryptosystem. In *IEEE International Conference on Computer as a tool*, pages 1879–1882, November 2005. Belgrade, Serbia & Montenegro. DOI: 10.1109/EURCON.2005.1630348, http://www.sps.ele.tue.nl/members/m.j.bastiaans/spc/demulder.pdf.

[MDS99] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Power analysis attacks of modular exponentiation in smartcards. In Koç and Paar [KP99], pages 144–157.

[MFGL15] Nadia El Mrabet, Jacques J. A. Fournier, Louis Goubin, and Ronan Lashermes. A survey of fault attacks in pairing based cryptography. *Cryptography and Communications*, 7(1):185–205, 2015.

[MHER14] Nicolas Moro, Karine Heydemann, Emmanuelle Encrenaz, and Bruno Robisson. Formal verification of a software countermeasure against instruction skip attacks. *J. Cryptographic Engineering*, 4(3):145–156, 2014.

[Mil85] Victor S. Miller. Use of elliptic curves in cryptography. In Hugh C. Williams, editor, *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1985.

[MO08] Marcel Medwed and Elisabeth Oswald. Template attacks on ECDSA. In Kyo-Il Chung, Kiwook Sohn, and Moti Yung, editors, *Information Security Applications, 9th International Workshop, WISA 2008, Jeju Island, Korea, September 23-25, 2008, Revised Selected Papers*, volume 5379 of *Lecture Notes in Computer Science*, pages 14–27. Springer, 2008.

[Mon85] Peter L. Montgomery. Modular multiplication without trial division. *Math. Comput.*, 44(170):519–521, April 1985.

[MS10] Dustin Moody and Dan Shumow. Isogenies on edwards curves. https://www.microsoft.com/en-us/research/publication/isogenies-on-edwards-curves/, January 2010.

[MvOV96] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[NIS13] NIST. FIPS publication 186-4 - Digital Signature standard (DSS). Technical report, National Institute of Standards and Technology (NIST), July 2013.

[NNS10] Michael Naehrig, Ruben Niederhagen, and Peter Schwabe. New software speed records for cryptographic pairings. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *Progress in Cryptology - LATINCRYPT 2010, First International Conference on Cryptology and Information Security in Latin America, Puebla, Mexico, August 8-11, 2010, Proceedings*, volume 6212 of *Lecture Notes in Computer Science*, pages 109–123. Springer, 2010.

[ÖBPV08] Siddika Berna Örs, Lejla Batina, Bart Preneel, and Joos Vandewalle. Hardware implementation of an elliptic curve processor over GF(p) with Montgomery modular multiplier. *IJES*, 3(4):229–240, 2008.

[PITM14] Guilherme Perin, Laurent Imbert, Lionel Torres, and Philippe Maurine. Attacking randomized exponentiations using unsupervised learning. In Prouff [Pro14], pages 144–160.

[Pro14] Emmanuel Prouff, editor. *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, volume 8622 of *Lecture Notes in Computer Science*. Springer, 2014.

[Riv11] Matthieu Rivain. Fast and regular algorithms for scalar multiplication over elliptic curves. *IACR Cryptology ePrint Archive*, 2011:338, 2011.

[RO04] Christian Rechberger and Elisabeth Oswald. Practical template attacks. In Chae Hoon Lim and Moti Yung, editors, *Information Security Applications, 5th International Workshop, WISA 2004, Jeju Island, Korea, August 23-25, 2004, Revised Selected Papers*, volume 3325 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2004.

[RSA83] R.L. Rivest, A. Shamir, and L.M. Adleman. Cryptographic communications system and method, September 20 1983. US Patent 4,405,829.

[Sch00] Werner Schindler. A timing attack against RSA with the chinese remainder theorem. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 109–124. Springer, 2000.

[Sch02] Werner Schindler. A combined timing and power attack. In David Naccache and Pascal Paillier, editors, *Public Key Cryptography, 5th International*

Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12-14, 2002, Proceedings, volume 2274 of *Lecture Notes in Computer Science*, pages 263–279. Springer, 2002.

[Sch15] Werner Schindler. Exclusive exponent blinding may not suffice to prevent timing attacks on RSA. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 229–247. Springer, 2015.

[Sha99] A. Shamir. Method and apparatus for protecting public key schemes from timing and fault attacks, November 23 1999. US Patent 5,991,415.

[SKQ01] Werner Schindler, François Koeune, and Jean-Jacques Quisquater. Improving divide and conquer attacks against cryptosystems by better error detection / correction strategies. In Bahram Honary, editor, *Cryptography and Coding, 8th IMA International Conference, Cirencester, UK, December 17-19, 2001, Proceedings*, volume 2260 of *Lecture Notes in Computer Science*, pages 245–267. Springer, 2001.

[SST04] Hisayoshi Sato, Daniel Schepers, and Tsuyoshi Takagi. Exact analysis of montgomery multiplication. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 290–304. Springer, 2004.

[SW03] Werner Schindler and Colin D. Walter. More detail for a combined timing and power attack against implementations of RSA. In Kenneth G. Paterson, editor, *Cryptography and Coding, 9th IMA International Conference, Cirencester, UK, December 16-18, 2003, Proceedings*, volume 2898 of *Lecture Notes in Computer Science*, pages 245–263. Springer, 2003.

[TC14] Assia Tria and Dooho Choi, editors. *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014, Busan, South Korea, September 23, 2014*. IEEE Computer Society, 2014.

[TK01] Sergios Theodoridis and Konstantinos Koutroumbas. Pattern recognition and neural networks. In Georgios Paliouras, Vangelis Karkaletsis, and Constantine D. Spyropoulos, editors, *Machine Learning and Its Applications, Advanced Lectures*, volume 2049 of *Lecture Notes in Computer Science*, pages 169–195. Springer, 2001.

[Uni] University of Sydney. Magma Computational Algebra System. http://magma.maths.usyd.edu.au/magma/, Accessed on 2014-08-22.

[Ver12] Vincent Verneuil. *Elliptic curve cryptography and security of embedded devices*. Theses, Université de Bordeaux, June 2012.

[Vig08] David Vigilant. RSA with CRT: A new cost-effective solution to thwart fault attacks. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 130–145. Springer, 2008.

[Wag04] David Wagner. Cryptanalysis of a provably secure CRT-RSA algorithm. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *ACM Conference on Computer and Communications Security*, pages 92–97. ACM, 2004.

[Wer02] Schindler Werner. Optimized timing attacks against public key cryptosystems. *Statistics & Risk Modeling*, 20(1-4):191–210, 2002.

[WO11] Carolyn Whitnall and Elisabeth Oswald. A Comprehensive Evaluation of Mutual Information Analysis Using a Fair Evaluation Framework. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 316–334. Springer, 2011.

[WOS14] Carolyn Whitnall, Elisabeth Oswald, and François-Xavier Standaert. The Myth of Generic DPA ... and the Magic of Learning. In Benaloh [Ben14], pages 183–205.

[WT01] Colin D. Walter and Susan Thompson. Distinguishing exponent digits by observing modular subtractions. In David Naccache, editor, *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, volume 2020 of *Lecture Notes in Computer Science*, pages 192–207. Springer, 2001.

[WvWM11] Marc F. Witteman, Jasper G. J. van Woudenberg, and Federico Menarini. Defeating RSA multiply-always and message blinding countermeasures. In Aggelos Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 77–88. Springer, 2011.

List of Figures

List of Tables

List of Algorithms

Amélioration d'attaques par canaux auxiliaires sur la cryptographie asymétrique

Margaux Dugardin

RESUME : Depuis les années 90, les attaques par canaux auxiliaires ont remis en cause le niveau de sécurité des algorithmes cryptographiques sur des composants embarqués. En effet, tout composant électronique produit des émanations physiques, telles que le rayonnement électromagnétique, la consommation de courant ou encore le temps d'exécution du calcul. Or il se trouve que ces émanations portent de l'information sur l'évolution de l'état interne. On parle donc de canal auxiliaire, car celui-ci permet à un attaquant avisé de retrouver des secrets cachés dans le composant par l'analyse de la « fuite » involontaire.

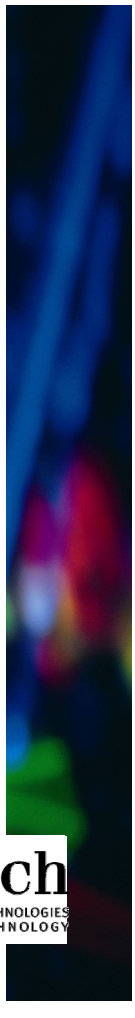
Cette thèse présente d'une part deux nouvelles attaques ciblant la multiplication modulaire permettant d'attaquer des algorithmes cryptographiques protégés et d'autre part une démonstration formelle du niveau de sécurité d'une contre-mesure. La première attaque vise la multiplication scalaire sur les courbes elliptiques implémentée de façon régulière avec un masquage du scalaire. Cette attaque utilise une unique acquisition sur le composant visé et quelques acquisitions sur un composant similaire pour retrouver le scalaire entier. Une fuite horizontale durant la multiplication de grands nombres a été découverte et permet la détection et la correction d'erreurs afin de retrouver tous les bits du scalaire. La seconde attaque exploite une fuite due à la soustraction conditionnelle finale dans la multiplication modulaire de Montgomery. Une étude statistique de ces soustractions permet de remonter à l'enchainement des multiplications ce qui met en échec un algorithme régulier dont les données d'entrée sont inconnues et masquées. Pour finir, nous avons prouvé formellement le niveau de sécurité de la contre-mesure contre les attaques par fautes du premier ordre nommée extension modulaire appliquée aux courbes elliptiques.

MOTS-CLEFS : attaque par canaux auxiliaires, cryptographie asymétrique, cryptographie sur courbes elliptiques, RSA

ABSTRACT : Since the 1990s, side channel attacks have challenged the security level of cryptographic algorithms on embedded devices. Indeed, each electronic component produces physical emanations, such as the electromagnetic radiation, the power consumption or the execution time. Besides, these emanations reveal some information on the internal state of the computation. A wise attacker can retrieve secret data in the embedded device using the analyzes of the involuntary "leakage", that is side channel attacks. This thesis focuses on the security evaluation of asymmetric cryptographic algorithm such as RSA and ECC. In these algorithms, the main leakages are observed on the modular multiplication.

This thesis presents two attacks targeting the modular multiplication in protected algorithms, and a formal demonstration of security level of a countermeasure named modular extension. A first attack is against scalar multiplication on elliptic curve implemented with a regular algorithm and scalar blinding. This attack uses a unique acquisition on the targeted device and few acquisitions on another similar device to retrieve the whole scalar. A horizontal leakage during the modular multiplication over large numbers allows to detect and correct easily an error bit in the scalar. A second attack exploits the final subtraction at the end of Montgomery modular multiplication. By studying the dependency of consecutive multiplications, we can exploit the information of presence or absence of final subtraction in order to defeat two protections : regular algorithm and blinding input values. Finally, we prove formally the security level of modular extension against first order fault attacks applied on elliptic curves cryptography.

KEY-WORDS : Side channel attack, asymmetric cryptography, Elliptic Curve Cryptography, RSA



TELECOM ParisTech

EDiTE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY