Pseudo-random generators and pseudo-random functions : cryptanalysis and complexity measures

Thierry Mefenza Nountu

THÈSE DE DOCTORAT EN COTUTELLE

de l'Université de recherche Paris Sciences et Lettres
PSL Research University et de l'Université de Yaoundé 1

Préparée à l'École normale supérieure et à l'université de Yaoundé 1

Pseudo-Random Generators and Pseudo-Random Functions : Cryptanalysis and Complexity Measures

École doctorale n°386
Sciences Mathématiques de Paris Centre

Spécialité : Informatique

Soutenue par **Thierry MEFENZA NOUNTU**
le 28 novembre 2017

Dirigée par
Damien VERGNAUD
et **Marcel TONGA**

COMPOSITION DU JURY

M. LAGUILLAUMIE Fabien
Université Lyon 1
Président du jury

M. VERGNAUD Damien
Université Pierre et Marie Curie
Institut Universitaire de France
Directeur de thèse

M. TONGA Marcel
Université Yaoundé 1
Directeur de thèse

M. DUQUESNE Sylvain
Université Rennes 1
Rapporteur

M. RENAULT Guénaël
ANSSI
Rapporteur

M. FERREIRA ABDALLA Michel
CNRS, DI/ENS, PSL Research University
Membre de jury

Mme. IONICA Sorina
Université de Picardie
Membre du jury

Pseudo-Random Generators and Pseudo-Random Functions: Cryptanalysis and Complexity Measures

Thierry Mefenza Nountu

Thèse de doctorat dirigée par
Damien Vergnaud et Marcel Tonga

Abstract

Randomness is a key ingredient in cryptography. For instance, random numbers are used to generate keys, for encryption and to produce nonces. They are generated by pseudo-random generators and pseudo-random functions whose constructions are based on problems which are assumed to be difficult. In this thesis, we study some complexity measures of the Naor-Reingold and Dodis-Yampolskiy pseudo-random functions and study the security of some pseudo-random generators (the linear congruential generator and the power generator on elliptic curves) and some pairing-based signatures based on *exponent-inversion* framework.

We show that the Dodis-Yampolskiy pseudo-random functions is uniformly distributed and that a low-degree or low-weight multivariate polynomial cannot interpolate the Naor-Reingold and Dodis-Yampolskiy pseudo-random functions over finite fields and over elliptic curves. The contrary would be disastrous since it would break the security of these functions and of problems on which they are based. We also show that the linear congruential generator and the power generator on elliptic curves are insecure if too many bits are output at each iteration.

Practical implementations of cryptosystems often suffer from critical information leakage through side-channels. This can be the case when computing the exponentiation in order to compute the output of the Dodis-Yampolskiy pseudo-random function and more generally in well-known pairing-based signatures (Sakai-Kasahara signatures, Boneh-Boyen signatures and Gentry signatures) based on the *exponent-inversion* framework. We present lattice-based polynomial-time (heuristic) algorithms that recover the signer's secret in the pairing-based signatures when used to sign several messages under the assumption that blocks of consecutive bits of the exponents are known by the attacker.

Résumé

L'aléatoire est un ingrédient clé en cryptographie. Par exemple, les nombres aléatoires sont utilisés pour générer des clés, pour le chiffrement et pour produire des nonces. Ces nombres sont générés par des générateurs pseudo-aléatoires et des fonctions pseudo-aléatoires dont les constructions sont basées sur des problèmes qui sont supposés difficiles. Dans cette thèse, nous étudions certaines mesures de complexité des fonctions pseudo-aléatoires de Naor-Reingold et Dodis-Yampolskiy et étudions la sécurité de certains générateurs pseudo-aléatoires (le générateur linéaire congruentiel et le générateur puissance basés sur les courbes elliptiques) et de certaines signatures à base de couplage basées sur le paradigme d'inversion.

Nous montrons que la fonction pseudo-aléatoire de Dodis-Yampolskiy est uniformément distribué et qu'un polynôme multivarié de petit dégré ou de petit poids ne peut pas interpoler les fonctions pseudo-aléatoires de Naor-Reingold et de Dodis-Yampolskiy définies sur un corps fini ou une courbe elliptique. Le contraire serait désastreux car un tel polynôme casserait la sécurité de ces fonctions et des problèmes sur lesquels elles sont basées. Nous montrons aussi que le générateur linéaire congruentiel et le générateur puissance basés sur les courbes elliptiques sont prédictibles si trop de bits sont sortis à chaque itération.

Les implémentations pratiques de cryptosystèmes souffrent souvent de fuites critiques d'informations à travers des attaques par canaux cachés. Ceci peut être le cas lors du calcul de l'exponentiation afin de calculer la sortie de la fonction pseudo-aléatoire de Dodis-Yampolskiy et plus généralement le calcul des signatures dans certains schémas de signatures bien connus à base de couplage (signatures de Sakai-Kasahara, Boneh-Boyen et Gentry) basées sur le paradigme d'inversion. Nous présentons des algorithmes (heuristiques) en temps polynomial à base des réseaux qui retrouvent le secret de celui qui signe le message dans ces trois schémas de signatures lorsque plusieurs messages sont signés sous l'hypothèse que des blocs consécutifs de bits des exposants sont connus de l'adversaire.

Acknowledgments

La réalisation de cette thèse n'aurait pas été possible sans le support inconditionnel de certaines personnes dont je ne peux décrire ici en quelques mots le rôle capital qu'elles ont pu jouer pour me permettre d'arriver à cette fin. Ainsi, j'adresse de tout mon coeur mes remerciements aux personnes qui m'ont aidées, encouragées, motivées afin de permettre la réalisation de cette thèse et je m'excuse d'avance auprès de ceux que j'ai oubliés.

Je remercie très chaleureusement mes directeurs de thèse Damien Vergnaud et Marcel Tonga d'avoir accepté d'encadrer cette thèse. Pendant ces trois années vous m'avez initié à la recherche, guidé dans mon travail, appris a être autonome, encouragé quand j'étais moins optimiste, énormément conseillé et je vous en suis très reconnaissant pour tout cela. Je voudrais remercier particulièrement Damien qui en plus de l'encadrement, a toujours été là pour m'aider lorsque j'avais des problèmes de visa et qui malgré parfois le fait que je sois au Cameroun, n'hésitait pas à travailler avec moi par Skype. Je le remercie également pour son aide pour l'obtention du financement qui m'a permis de terminer cette thèse dans des meilleures conditions.

J'adresse mes sincères remerciements à David Pointcheval qui m'a accueilli dans l'équipe Crypto de l'ENS et mis à ma disposition tout le matériel nécessaire pour la réalisation de ce travail.

Je remercie de tout coeur Sylvain Duquesne qui m'a mis en contact avec Damien et qui m'a invité en France pour la première fois en 2013 afin de discuter d'un éventuel sujet de thèse avec Damien.

Rapporter une thèse n'est pas toujours une chose facile car cela nécessite de la volonté, du temps et de la disponibilité. Ainsi, je voudrais remercier mes deux rapporteurs Sylvain Duquesne et Guénaël Rénault pour avoir lu attentivement et commenté cette thèse. Je témoigne aussi ma gratitude envers les autres membres de jury Michel Abdalla, Guillaume Laguillaumie et Sorina Ionica pour avoir accepté de faire partir de ce jury de thèse.

Je voudrais remercier tous les membres de l'équipe Crypto de l'ENS, avec qui j'ai discuté et passé des moments très agréables lors des pauses cafés ou des évènements au laboratoire : Michel Abdalla, Balthazar Bauer, Sonia Belaïd, Fabrice Benhamouda, Florian Bourse, Céline Chevalier, Jérémy Chotard, Mario Cornejo, Edouard Dufour Sans, Angelo De Caro, Geoffroy Couteau, Rafaël Del Pino, Aurélien Dupin, Pierre-Alain Dupont, Pooya Farshim, Houda Ferradi, Pierre-Alain Fouque, Georg Fuchsbauer, Romain Gay, Dahmun Goudarzi, Aurore Guillevic, Chloé Hébant, Julia Hesse, Duong Hieu Phan, Louiza Khati, Pierrick Méaux, Michele Minelli, David Naccache, Anca Nitulescu, Michele Orrù, Alain Passelègue, Thomas Peters, David Pointcheval, Thomas Prest, Răzvan Roşie, Mélissa Rossi , Sylvain Ruhault, Adrian Thillard, Bogdan Ursu, Damien Vergnaud et Hoeteck Wee.

Je remercie également l'équipe administrative du DI et le SPI, en particulier Lise-Marie Bivard, Isabelle Delais, Joëlle Isnard, Sophie Jaudon, Valérie Mongiat, Ludovic Ricardou et Benoît Speziari pour leur disponibilité, leur amabilité et leur professionnalisme quand j'avais besoin de quoi que ce soit.

Je voudrais remercier tous les menbres de l'équipe ERAL du département de Mathéma-

tiques de l'université de Yaoundé 1 pour leurs collaborations et encouragements et plus particulièrement : Etienne Assongmo, Hortense Boudjou, Serges Feukoua, Emmanuel Fouotsa, Alexandre Fotué, Albert Kadji, Maurice Kianpi, Blaise Koguep, Celestin Lélé, Cristophe Mouaha, Celestin Ndjeya, Bertrand Nguefack, Celestin Nkuimi, Amina Pecha, Hervé Tallé, Yannick Tenkeu, Daniel Tieudjo, Roméo Tioffo et Marcel Tonga, Sulamithe Tsakou.

Je remercie également tous les membres du pôle PRMAIS qui m'ont toujours encouragés et motivés, et spécialement son responsable Tony Ezome pour tous les soutiens financiers sans lesquels la réalisation de cette thèse n'aurait pas été possible.

Je remercie très sincèrement mes parents qui ont toujours été là depuis ma première journée d'école. Je remercie aussi toute ma famille ainsi que tous mes amis pour leurs soutiens, leurs conseils et encouragements.

Enfin je rends grâce et honneur à la source intarissable de bonté, mon seigneur et sauveur Jésus-Christ qui a toujours été présent dans les bons comme dans les mauvais moments durant cette thèse. Je te remercie de m'avoir accordé la santé nécessaire pour la réalisation de ce travail et de m'avoir permis d'arriver jusqu'à ce niveau. Je sais que je peux toujours compter sur toi dans la suite de ma vie car par ta grande miséricorde tu seras toujours avec moi et tu ne m'abandonneras jamais.

Contents

Abstract iii

Résumé v

Acknowledgments vii

1. Introduction 1
- 1.1. Pseudo-random generators . . . 5
  - 1.1.1. Constructions . . . 6
  - 1.1.2. Applications . . . 9
- 1.2. Pseudo-random functions . . . 9
  - 1.2.1. Constructions . . . 10
  - 1.2.2. Applications of pseudo-random functions . . . 11
- 1.3. Our results . . . 12
  - 1.3.1. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Functions 12
  - 1.3.2. Distribution and Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function . . . 12
  - 1.3.3. Inferring a Linear Congruential Generator and a Power Generator on Elliptic Curves . . . 13
  - 1.3.4. Lattice Attacks on Pairing-Based Signatures . . . 13
- 1.4. Organization . . . 14

I. Complexity Measures of Pseudo-Random Functions 15

2. Preliminaries 17
- 2.1. Notation . . . 18
- 2.2. Finite fields . . . 18
- 2.3. Elliptic Curves . . . 19
  - 2.3.1. Definition and addition law . . . 19
  - 2.3.2. Division polynomials of elliptic curves . . . 20
  - 2.3.3. Summation polynomials . . . 22
- 2.4. Exponential Sums . . . 22
  - 2.4.1. Finite Fields and Exponential Sums . . . 23
  - 2.4.2. Elliptic Curves and Exponential Sums . . . 25
- 2.5. Polynomial Approximation of the Discrete Logarithm . . . 26

3. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Functions 29
- 3.1. Naor-Reingold pseudo-random function . . . 31
- 3.2. Auxiliary results . . . 32

3.3. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Function over Finite Fields . . . 32
3.3.1. Polynomial Interpolation with variable secret key . . . 32
3.3.2. Polynomial Interpolation with fixed secret key . . . 38
3.4. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Function over Elliptic Curves . . . 42
3.4.1. Polynomial Interpolation with fixed secret key . . . 42
3.4.2. Polynomial Interpolation with variable secret key . . . 48

4. Distribution and Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function 51
4.1. Distribution of the Dodis-Yampolskiy Pseudo-Random Functions . . . 53
4.1.1. Distribution of the Dodis-Yampolskiy Pseudo-Random Function over Finite Fields . . . 53
4.1.2. Distribution of the Dodis-Yampolskiy Pseudo-Random Function over Elliptic Curves . . . 55
4.2. Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function over Finite Fields . . . 56
4.3. Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function over Elliptic Curves . . . 57

II. Lattice-Based Cryptanalysis of Pseudo-Random Generators and Signatures 61

5. Preliminaries 63
5.1. **Coppersmith's methods** . . . 64
5.1.1. **First method** . . . 64
5.1.2. **Second method** . . . 66
5.2. **Analytic Combinatorics** . . . 67
5.2.1. Introduction . . . 68
5.2.2. Combinatorial Classes, Sizes, and Parameters . . . 68
5.2.3. Counting the Elements: Generating Functions . . . 69
5.2.4. Counting the Parameters of the Elements: Bivariate Generating Functions 70
5.2.5. Counting the Parameters of the Elements up to a Certain Size . . . 71
5.2.6. Asymptotic Values: Transfer Theorem . . . 72
5.3. Some useful applications of the technique . . . 72
5.3.1. Counting the Bounds for the Monomials (Useful Examples) . . . 72
5.3.2. Counting the Bounds for the Polynomials . . . 74

6. Inferring a Linear Congruential Generator and a Power Generator on Elliptic Curves 77
6.1. Linear Congruential Generator and Power Generator on Elliptic Curves . . . 78
6.2. Predicting EC-LCG Sequences for Known Composer . . . 78
6.3. Predicting EC-LCG Sequences for Unknown Composer . . . 90
6.3.1. Complexity of the attack . . . 92
6.4. **Predicting the Elliptic curve power generator** . . . 94

7. Lattice Attacks on Pairing-Based Signatures **99**
7.1. Sakai-Kasahara, Boneh-Boyen and Gentry's Pairing-Based Signatures Schemes 101
7.2. Lattice Attack On Gentry Signatures . . . 102
7.2.1. Gentry Signatures . . . 102
7.2.2. Description of the Attack . . . 103
7.2.3. Experimental Results . . . 106
7.3. Concrete Attack Examples against Gentry signatures . . . 107
7.4. Lattice Attack on Boneh-Boyen Signatures . . . 109
7.4.1. Boneh-Boyen Signatures . . . 109
7.4.2. Description of the Attack . . . 109
7.4.3. Experimental results . . . 111
7.5. Lattice Attack on Sakai-Kasahara Signatures . . . 111
7.5.1. Sakai-Kasahara Signatures . . . 111
7.5.2. Description of the Attack . . . 112
7.5.3. Experimental results . . . 113

8. Conclusion and Open Questions **115**
8.1. Conclusion . . . 115
8.2. Open questions . . . 116

Bibliography **119**

Chapter 1.

Introduction

Cryptography can be defined as the practice and study of techniques for secure communication in the presence of third parties called adversaries or eavesdroppers. Today, cryptography is very present in our daily life: emails, credit cards, electronic banking, online shopping, secure network communications, authentications, etc. The main goals of cryptography is to ensure privacy, integrity and authenticity. Suppose two people usually called Alice and Bob want to communicate through an insecure channel in the presence of an adversary called Eve. If Alice is the sender (the person who sends the message also called the plaintext) and Bob the receiver (the person who receives the message), then providing privacy means that the message sent by Alice to Bob should be hidden from Eve. Providing authenticity or integrity means that one wants Bob, upon receiving a communication supposed to be from Alice, to have a way of assuring itself that it really did originate from Alice and was not sent by Eve, or modified along the road by Eve. To achieve these goals (privacy, authenticity and integrity), Alice and Bob are supplied with a set of algorithms also called a protocol or a scheme or a cryptosystem. There is an algorithm for the sender to run also called the encryption algorithm that allow him to encrypt the message he wants to send (the encrypted message is also called the ciphertext) and an algorithm for the receiver that allows him to decrypt the ciphertext and get the message together if possible with an associated message telling him whether or not to regard it as authentic. These algorithms depends on some cryptographic keys (the key used to encrypt a message must be known to the sender and the one used to decrypt must be known to the receiver). The modern cryptography can be divided into two areas of study:

- Secret-key or symmetric cryptography: secret-key cryptography refers to encryption and decryption algorithms in which both the sender and the receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976.
- Public-key or asymmetric cryptography: A significant disadvantage of symmetric cryptography is the key management necessary to use them securely. Specifically, each distinct pair of communicating parties must share a different key. The number of keys required increases as the square of the number of network members, which very quickly requires complex key management schemes to keep them all consistent and secret. The difficulty of securely establishing a secret key between two communicating parties, when a secure channel does not already exist between them is then a considerable practical obstacle for cryptography users in the real world. The idea of public-key cryptography

is due to Diffie and Hellman in 1976 [DH76]. In the public-key setting, a party possesses a pair of keys: a public key also denote pk (which is used by a sender to send a message to the party) and an associated secret key also denoted sk (which is used by the party to decrypt all the messages encrypted with his public key). A party's public key is made publicly known and bound to its identity. For example, a party's public key might be published in a phone book. A party's secret key is kept secret. The computation of one key (the private key) is computationally infeasible from the other (the public key), even though they are necessarily related. Most public-key algorithms involve operations such as modular multiplication and exponentiation in groups, which are much more computationally expensive than the operations use in most secret-key algorithms. So in practice, a fast symmetric-key protocol is used to secure communications and public-key protocols to share the secret key used.

The following question can then be addressed:

What is a secure protocol?

Perfect security. Intuitively, a secure protocol is one for which an encrypted message remains well hidden even after seeing its encryption. In other words, an encryption algorithm is perfectly secure if a ciphertext produced using it provides no information about the plaintext without knowledge of the key. That is, it cannot be broken even when the adversary has unlimited computing power. The adversary simply does not have enough information to break the encryption scheme. The one-time pad (given a n-bit string message m and a n-bit string key k, the ciphertext is simply $c = m \oplus k$, where $\oplus$ denotes the bit-Xor) is known to achieve the perfect security. However, the one-time pad is not very practical, in the sense that the key must be as long as the message: if Alice wants to send a 10 GB file to Bob, they must already share a 10 GB key. Unfortunately, this cannot be avoided since it is proven that any perfectly secure cipher must have a key space at least as large as its message space. This fact provides the motivation for developing a definition of security for which the message remains well hidden but in the presence of limited computing power adversaries, and which allows one to encrypt long messages using short keys. Many other notions of security (for instance the semantic security) were then introduced and allow us to build secure schemes that use reasonably short keys.

Computational-complexity theory By considering weaker notions of security, modern cryptography now introduces adversaries who have limited computing power (probabilistic polynomial time adversaries). For practice purposes, symmetric and asymmetric primitives have security as long as adversaries do not have too much computing time. These primitives are breakable in principle but not in practice and are most often based on the computational complexity of hard problems, often from number theory. Among the hard problems we have:

- Discrete logarithm problem (DLP). Let g be a generator of a group $\mathbb{G}$ (denoted multiplicatively) of order q. Given g, h, the discrete logarithm problem is to find a such that $h = g^a$. The DLP difficulty depends on the choice of the group G. In cryptography, two interesting choices for $\mathbb{G}$ are a subgroup of the multiplicative group of a (prime) finite field and a subgroup of the points of an elliptic curve defined over a finite field. The use of elliptic curves in cryptography was suggested independently by Neal Koblitz [N K87] in 1987 and by Victor S. Miller [Mil86] in 1985. Elliptic curves

are increasingly used to instantiate public-key cryptography protocols, for example encryption, digital signatures, pseudo-random generators, key agreement and other tasks. They are also used in several integer factorization and primality testing algorithms that have applications in cryptography. The DLP takes sub-exponential time in $\mathbb{F}_p$ and is even harder in the present state (with exponential time) in elliptic curves $E(\mathbb{F}_p)$ (we recall that if E is an elliptic curve over the finite field $\mathbb{F}_p$, P and Q are points in $E(\mathbb{F}_p)$, the elliptic curve discrete logarithm problem consists in finding $n \in \mathbb{N}$ such that $Q = [n]P$). The size of the elliptic curve determines the difficulty of the problem. The primary benefit promised by elliptic curve cryptography is a smaller key size, reducing storage and transmission requirements i.e. that an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key. For example a 256-bit elliptic curve public key should provide the same level of security as a 3072-bit RSA public key. The last record in solving the discrete logarithm problem (16 June 2016), using the number field sieve, is the computation of a discrete logarithm modulo a 232-digit prime which roughly corresponds to the factoring of a 768-bits safe prime. The hardness of the DLP is the foundation of several cryptographic systems (e.g. Diffie-Hellman key agreement [DH76], ElGamal encryption and signature [ElG85] or the Schnorr signature[Sch90]).

- Diffie-Hellman assumptions. Given a cyclic group $\mathbb{G}$ (denoted multiplicatively) generated by some element g, the *computational Diffie-Hellman assumption* states that it is difficult to compute the element g^{xy} from known elements g^x and g^y (for x and y picked uniformly at random between 1 and the order of $\mathbb{G}$). This assumption is the basis of the Diffie-Hellman key exchange [DH76] and the most efficient means known to solve this computational problem is to solve the standard discrete logarithm problem in $\mathbb{G}$. Unfortunately, even the computational Diffie-Hellman assumption by itself is generally not sufficient to assess the security of protocols proposed and used in cryptography. Cryptographers have then proposed much stronger assumptions in order to analyze the security of cryptosystems. For instance, the *decision Diffie-Hellman assumption* [Bon98] states that given a cyclic group $\mathbb{G}$ given some elements g, g^x and g^y, no efficient algorithm can distinguish between g^{xy} and an element picked uniformly at random in $\mathbb{G}$. This assumption has been used to prove the security of many cryptographic protocols, most notably the ElGamal [ElG85] and Cramer-Shoup cryptosystems [CS98] and in numerous important cryptographic applications.

- Diffie-Hellman inversions assumptions. Given a cyclic group $\mathbb{G}$ generated by some element g, and the tuple $(g, g^x, \ldots, g^{x^q})$, the q-Diffie-Hellman inversions assumption ([DY05] and the references there in) states that it is difficult to compute the element $g^{1/x}$. This assumption is non-standard and Cheon [Che10] proved that it is stronger than the classical discrete logarithm assumption in $\mathbb{G}$. A much stronger assumption called the q-decisionnal bilinear Diffie-Hellman inversions assumption [DY05; SK03; BB04a] was proposed. Given a bilinear cyclic group $\mathbb{G}$ generated by some element g (that is the group action in $\mathbb{G}$ is efficiently computable, there exists a multiplicative group $\mathbb{G}_1$ and an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ and $e(g, g) \neq 1_{\mathbb{G}_1}$), and the tuple $(g, g^x, \ldots, g^{x^q})$, the q-decisionnal bilinear Diffie-Hellman inversions assumption states that it is difficult to distinguish $e(g, g)$ from a random element. This assumption was shown in [DY05] to be difficult in the generic group model and it was

used in [SK03; BB04a; Gen06] to construct secure identity based encryption schemes.

- Integer factorization problem (FACT). Given a positive integer N, the problem is to find its prime factors, i.e., find the pairwise distinct primes p_i and positive integer powers e_i such that $N = p_1^{e_1} ... p_n^{e_n}$. It is generally believed that the most difficult setting for the factorization problem is when $N = pq$ is the product of only two primes p and q of the same large size since the difficulty of the factorization problem is nonuniform (i.e factoring integers N whose second largest prime factor is bounded by a polynomial in $\log N$ can be performed in polynomial time). It is straightforward to compute $N = pq$ in $O((\log N)^2)$ time and (presumedly) hard to invert this operation when p and q are pairwise distinct primes chosen randomly.

- The RSA Problem . Given (N, e, c) where $c \in \mathbb{Z}_N$, e an integer and $N = pq$, the RSA problem is to find x such that $c = x^e \bmod N$. The RSA scheme relies on the difficulty for solving equations of the form $x^e = c \bmod N$, where e, c, and N are known and x is an arbitrary number. In other words, the security of RSA relies on the assumption that it is difficult to compute e-th roots modulo N.

  The RSA Problem is clearly no harder than the factorization problem since an adversary who can factor N can also compute the private key (p, q, d) from the public key (N, e), where d is an integer satisfying the equation $ed = 1 \bmod (p-1)(q-1)$. However, so far there are no proofs that the converse is true meaning that the RSA problem is *apparently* as difficult as factoring: Whether an algorithm for solving the RSA Problem can be efficiently converted into an integer factoring algorithm is an important open problem. However, Boneh and Venkatesan [BV98] have given evidence that such a reduction is unlikely to exist when the public exponent is very small, such as $e = 3$ or 17.

  It is important to know which parameter sizes to choose when the RSA problem serves as the foundation of a cryptosystem. The current record for factoring general integers was announced on December 12 in 2009, by a team including researchers from CWI, EPFL, INRIA and NTT. The consortium factored the RSA-768 (232-digit number) using the number field sieve (NFS) [KAF+10]. This effort required the equivalent of almost 2000 computing years on a single core 2.2 GHz AMD Opteron. Now the NIST's recommendation is that future systems should use RSA keys with a minimum size of 3072 bits. In 1994, Peter Shor [Sho97] introduced a quantum algorithm solving FACT in polynomial time.

These problems are unproven but are assumed to be difficult. And based on these problems, many cryptographic primitives and schemes with provable security have been proposed. These schemes and many other applications require random numbers or bits which are produced by random number and random bit generators, RNGs and RBGs. In cryptography, these generators are employed to:

- produce secret keys: computer cryptography uses integers for keys. The elementary method to read encrypted data without actually decrypting it is a brute force attack (i.e. simply attempting every number, up to the maximum length of the key). Therefore, it is important to use a sufficiently long key bit length that is unpredictable in order to make the brute force attack impractical.

- encrypt messages: probabilistic encryption is particularly important when using public-key cryptography. Suppose that the adversary observes a ciphertext, and suspects that the plaintext is either "YES" or "NO", or has some information about the plaintext. When a deterministic encryption algorithm is used, the adversary can simply try encrypting each of his guesses under the receiver's public key, and compare each result to the target ciphertext. To prevent this attack, public key encryption schemes must incorporate an element of randomness, ensuring that each plaintext maps into one of a large number of possible ciphertexts.

- produce nonces: a nonce is an arbitrary number used only once in a cryptographic communication. They are used in some authentication protocols, hashing protocols and encryption schemes. Nonces should be unpredictable to ensure the security of the underline protocols.

Random numbers can be generated by true random numbers generators (TRNGs). TRNGs measure some physical phenomenon that is expected to be random and then compensates for possible biases in the measurement process. Example sources include measuring atmospheric noise, thermal noise, and other external electromagnetic and quantum phenomena. For example, cosmic background radiation or radioactive decay or the noise of a semiconductor diode as measured over short timescales represent sources of natural entropy. However TRNGs are often biased, this means for example that on average their output might contain more ones than zeros and therefore does not correspond to a uniformly distributed random variable. This effect can be balanced by different means, but this post-processing reduces the number of useful bits as well as the efficiency of the generator. Another problem is that some TRNGs are very expensive or need at least an extra hardware device. In addition, these generators are often too slow for the intended applications in cryptography. As a solution, cryptographers proposed methods to efficiently generating numbers that look random for adversaries with limited power of computations (probabilistic polynomial time adversaries) by using little or no randomness. Among these methods we have pseudo-random generators and pseudo-random functions.

1.1. Pseudo-random generators

Recall that for the one-time pad to be perfectly secure, the key should be as long as the message. However, we would like to use a key that is much shorter. So the idea is to instead use a short ℓ-bit *seed* s as the encryption key, where ℓ is much smaller than L (L being the bit length of the message) and to stretch this seed into a longer, L-bit string that is used to encrypt the message and decrypt the ciphertext. The seed s is stretched using some efficient, deterministic algorithm G that maps ℓ-bit strings to L-bit strings. Thus, the key space for this modified one-time pad is $\{0,1\}^\ell$, while the message and ciphertext spaces are $\{0,1\}^L$. For $s \in \{0,1\}^\ell$ and $m, c \in \{0,1\}^L$, encryption and decryption are defined as follows: $E(s,m) := G(s) \oplus m$ and $D(s,c) := G(s) \oplus c$. This new scheme is called a stream cipher, and the function G is called a pseudo-random generator. A pseudo-random generator, or PRG for short, is an efficient deterministic algorithm G that, given as input a seed $s \in S$, computes an output $r = G(s) \in R$, where S, R are finite spaces. Our definition of security for a PRG captures the intuitive notion that if s is chosen at random from S and r is chosen at random from R, then no efficient adversary can effectively tell the difference between $G(s)$

and r: the two are computationally indistinguishable.
More formally, let $G : S \to R$ be a function. We say that G is a secure pseudo-random generator if G is efficiently computable and if for any polynomial-time adversary A, its advantage in the following algorithmic game is negligible. The game starts by picking a random challenge bit b and returns either $G(s)$, for a randomly chosen $s \in S$ if $b = 0$ or r, for a randomly chosen $r \in R$ if $b = 1$. Finally, the adversary outputs a bit b', and its advantage is defined by $2Pr[b = b'] - 1$. Here a polynomial time adversary A is simply a non-uniform probabilistic polynomial-time oracle Turing machine. The adversary is polynomial in the input length bit and with a single bit output. It is not known if cryptographically secure pseudo-random generators exist. Proving that they exist is difficult since their existence implies $P \neq NP$, which is widely believed but a famously open problem. The existence of cryptographically secure pseudorandom generators is widely believed as well and they are necessary for many applications in cryptography. It is well-known by combining the results of [BM84; JL99] that:

Theorem 1.1.1. *Cryptographically secure pseudo-random generators exist if and only if one-way functions exist.*

1.1.1. Constructions

Many constructions of pseudo-random generators were proposed, some based on unproven but believed hard problems and these pseudo-random generators were proved to be secure if the underlined problem is difficult. Others constructions were proposed based on difficult problems but have not yet been proved to be secure if the underlined problem is difficult.

1.1.1.1. Proven constructions

- The Blum Blum Shub PRG. This PRG was proposed in 1986 by Blum, Blum and Shub [LS86]. The following sequence (x_n) is defined:

  $$x_{n+1} = x_n^2 \bmod N,$$

  where $N = pq$ is the product of two large primes p and q. At each step of the algorithm, some intermediate output which is part of the final output is derived from x_{n+1}: the intermediate output is commonly either the bit parity of x_{n+1} or one or more of the least significant bits of x_{n+1}. The final output is the concatenation of all the intermediate outputs. For instance if one bit is derived at each step, then if L is the bit length of the output of the generator, we will need L steps to actually compute the desired number of bits. The seed $x_0 \neq 1$ should be an integer that is coprime to N (i.e. p and q are not factors of x_0). Its security relies on the computational difficulty of solving the quadratic residuosity problem which states that given integers a and $N = pq$, it is difficult to decide whether a is a quadratic residue modulo N or not. When the primes are chosen appropriately, and $O(\log \log N)$ least significant bits of each x_n are output, then if N is sufficiently large, distinguishing the output bits from random should be at least as difficult as solving the Quadratic residuosity problem modulo N.

- The Blum-Micali PRG. It is due to Blum and Micali [BM84] and its security relies on the discrete logarithm problem. Let p be an odd prime, and let g be a primitive root

modulo p. Let x_0 be a seed, and let:

$$x_{n+1} = g^{x_n} \bmod p,$$

where the exponent x_n is seen as an integer in $\{0, \ldots, p-1\}$. The nth intermediate output of the algorithm is 1 if $x_n < \frac{p-1}{2}$ and 0 otherwise. The final output is the concatenation of all the intermediate outputs.

- Dual-EC PRG. This generator is based on elliptic curve and was released by the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA) in 2006. The basic Dual-EC algorithm works as follows; given a public elliptic curve E and two public points P and Q on E, the following sequences are defined:

$$s_{i+1} = x(s_i P), \quad r_{i+1} = x(s_{i+1} Q) \quad i \in \mathbb{N},$$

where $x(A)$ denotes the abscissa of a point A, and s_0 is the seed. For each $i \geq 1$, some output is derived from r_i: for instance, for a n-bit string s_0, the n least significant bits of r_i could be computed at each step. The output of the generator is then the concatenation of the different outputs at each stage. Its security is based on the elliptic curve discrete logarithm problem. After its publication, it was criticized by experts for its poor design (it is very slower, its outputs are biased, it outputs too many bits and it is mathematically guaranteed to have a skeleton key d (with $P = dQ$) that makes the output entirely predictable to anyone in possession of the key). In 2007 Bruce Schneier wrote an article about Dual-EC in Wired Magazine and a pair of Microsoft researchers Dan Shumow and Niels Ferguson announced at the Crypto rump session in August 2007 that there was a possibility of a back door (secret knowledge that lets you predict outputs) in Dual-EC. By the end of 2007, for the public cryptographic community, Dual-EC was dead and gone. In 2013, The New York Times reported that documents in their possession appear to confirm that the backdoor was real and had been deliberately inserted by the NSA. NIST's list of DRBG (Deterministic Random Bit Generator) validations showed that Dual EC was provided in dozens of commercial cryptographic software libraries. Dual-EC was even the default pseudorandom number generator in RSA Security's BSafe library. How could an algorithm so criticized by the cryptographic community be present in widely used implementations? A partial explanation surfaced in December 2013, when Reuters reported that NSA paid RSA 10 million dollars in a deal that set Dual-EC as the preferred, or default, method for number generation in the BSafe software. In April 21, 2014, NIST withdrew Dual-EC from its draft guidance on random number generators.

1.1.1.2. Unproven constructions

We have some constructions based on elliptic curves which have not yet been proved to be secure but are widely used in practice:

- the elliptic curve linear congruential generator (EC-LCG). Given a modulus m and a multiplier a relatively prime to m, an increment b and a seed x_0, the linear congruential generator (LCG) is the sequence x_n defined by:

$$x_{n+1} = a x_n + b \bmod m.$$

The elements a, b and m can be secret or not and in order to increase the resistance of the LCG, Knuth [Knu85] proposed to output only most significant bits of each x_n. In 1989, Boyar [Boy89] showed that one can recover the seed of this generator in the bit-size of m if few least significant bits of each x_i are discarded when a, b and m are secret. Frieze et al [FHK+88] infer the LCG when few most significant bits of each x_n are output for known m and a. Joux and Stern [JS98] proposed a lattice attack on the LCG when few most significant bits of each x_n are output even for secret m and a. Their attack constructs an appropriate lattice from known information and computes the shortest vector in that lattice. Due to the insecurity of the LCG, Hallgren [S H94] proposed in 1994 an elliptic curve analogue of the linear congruential generator known as the EC-LCG. Let E be an elliptic curve defined over a prime finite field $\mathbb{F}_p$ and $G \in E(\mathbb{F}_p)$, the EC-LCG is a sequence U_n of points defined by the relation:

$$U_n = U_{n-1} \oplus G = nG \oplus U_0, \quad n \in \mathbb{N}$$

where $U_0 \in E(\mathbb{F}_p)$ is the initial value or seed. We refer to G as the *composer* of the generator. One can notice that if two consecutive values U_n, U_{n+1} of the generator are revealed, it is easy to find U_0 and G. So only the most significant bits of each coordinate of U_n are output, $n \in \mathbb{N}$ in the hope that this makes the resulting output sequence difficult to predict. In the cryptography setting, the initial value U_0 and the constants G, a and b may be kept secret. Gutierrez and Ibeas [GI07] consider two cases: the case where the *composer* G is known and a, b are kept secret and the case where the *composer* G is unknown and a, b are kept secret. In the first case, they showed that the EC-LCG is insecure if a proportion of at most $1/6$ of the least significant bits of two consecutive values of the sequence is hidden. When the *composer* is unknown, they showed heuristically that the EC-LCG is insecure if a proportion of at most $1/46$ of the least significant bits of three consecutive values of the sequence is hidden. Their result is based on a lattice basis reduction attack, using a certain linearization technique. In some sense, their technique can be seen as a special case of the problem of finding small solutions of multivariate polynomial congruences by using only linear relations on them. The Coppersmith's methods also tackle the problem of finding small solutions of multivariate polynomial congruences by taking products of the polynomials. Gutierrez and Ibeas due to the special structure of the polynomials involved claimed that "the Coppersmith's methods does not seem to provide any advantages", and that "It may be very hard to give any precise rigorous or even convincing heuristic analysis of this approach". We tackle this issue in this thesis.

- the elliptic curve power generator (EC-PG). Given a modulus m, an integer e and a seed v_0, the power generator (PG) is the sequence v_n defined by:

$$v_{n+1} = v_n^e \bmod m.$$

The integer e which is typically small ($e = 3$ or $e = 2$) is known and m which can be a prime number or an RSA modulus (the products of two prime numbers) can be secret or not. In order to increase the resistance of the PG, only some most significant bits of each v_n are output at each iteration. For $e = 2$ and a prime m of bit-size n, Blackburn et al [SS06] showed that if $2/3$n bits are output at each iteration, then they are able to recover the seed. Steinfeld et al [RW06] , uses Coppersmith's methods and generalized

this bound to $\frac{e}{e+1}n$. Herrmann and May [HM09] improved this bound to $\frac{e-1}{e}n$ by using Coppersmith's methods with a new technique called the unravelled linearization. In 2005, Lange and Shparlinski [LS05] proposed an elliptic curve analogue of the PG called the elliptic curve power generator (EC-PG). For a positive integer $e > 1$ and a point $V_0 \in E(\mathbb{F}_p)$ of order ℓ with $gcd(e; \ell) = 1$, the EC-PG is a sequence V_n of pseudo-random numbers defined by the relation:

$$V_n = eV_{n-1} = e^n V_0, \quad n \in \mathbb{N}$$

where $V_0 \in E(\mathbb{F}_p)$ is the initial value or seed. At each step, the algorithm outputs some most significant bits of each coordinate of V_n. To the best our knowledge, no result on the cryptanalysis of the EC-PG are known.

In Chapter 6, we show that the EC-PG and the EC-LCG are insecure if too many bits are output at each iteration. Some secure PRGs constructions collect new inputs in addition to the seed and produce outputs that depend on the previous inputs. The designs of such secure PRGs are based on some cryptographic primitives such as ciphers and hash functions and on special-purposes designs (for instance the Yarrow algorithm, Fortuna algorithm, ANSI X9.17 standard etc.).

1.1.2. Applications

Pseudo-random generators have numerous applications in cryptography. As mentioned, pseudo-random generators are used to construct stream ciphers. They may also be used to construct symmetric-key schemes (where a large number of messages can be safely encrypted under the same key), for key generation, for nonces and signature schemes.

1.2. Pseudo-random functions

Suppose Alice wishes to authenticate herself to Bob, by proving she knows a secret that they share. With a PRG, they could proceed as follows. They both seed a PRG with the shared secret. Bob picks and sends to Alice some random number i, and Alice proves she knows the share secret by responding with the ith random number generated by the PRG. But this solution requires state, and they both have to compute i random numbers. Instead, we would like random access to the sequence. This is the intuition behind pseudo-random functions: Bob gives to Alice some random i, and Alice returns $F_K(i)$, where F_K is indistinguishable from a random function. The notion of pseudo-random function family generalizes the notion of a pseudorandom generator.
In cryptography, a pseudo-random function family is a collection of functions (that can be evaluated efficiently using a secret-key) with the property that an adversary cannot efficiently observe any significant difference between the input-output behavior of a random instance of the family or that of a random function.

More formally, we consider collections of functions $\{F_n : \mathcal{K}_n \times \mathcal{D}_n \to \mathcal{R}_n\}_{n\in\mathbb{N}}$ that can be evaluated by a (deterministic) polynomial-time Turing Machine. We define an adversary as a (non-uniform) probabilistic polynomial-time oracle Turing machine with either access to:

- an oracle implementing a function $F : \mathcal{D}_n \to \mathcal{R}_n$ defined by picking uniformly at random a secret-key $k \in \mathcal{K}_n$ such that $F(m) = F_n(k, m)$ for any $m \in \mathcal{D}_n$;

- or an oracle simulating a truly random function $F : \mathcal{D}_n \to \mathcal{R}_n$ (*i.e.* whose outputs are sampled uniformly and independently at random).

This adversary can decide which queries to make to the oracle, perhaps based on answers received to previous queries and eventually, it outputs a single bit (which is its decision as to which function the oracle is implementing). The *advantage* of the adversary is the function of n defined as the difference of the probabilities (taken over the random choices made by the adversary and the oracle) that the adversary outputs 1 in the two cases. A collection of functions $\{F_n : \mathcal{K}_n \times \mathcal{D}_n \to \mathcal{R}_n\}_{n\in\mathbb{N}}$ is a pseudo-random function family if and only if no polynomial time adversary with advantage asymptotically larger than the inverse of a polynomial exists.
By combining the results of Goldreich, Goldwasser and Micali [GGM84] and Goldreich and Levin [GL89], the following result is known:

Theorem 1.2.1. *Pseudo-random functions exist if and only if one-way functions exist.*

1.2.1. Constructions

1.2.1.1. Pseudo-random functions from pseudo-random generators

Pseudo-random generators can be used to construct pseudo-random functions by the construction proposed by Goldreich, Goldwasser and Micali [GGM84]. Let $G : \{0,1\}^s \to \{0,1\}^{2s}$ be a PRG. Define G_0, G_1 to be the left and right halves of G, so that $G(x) = G_0(x)||G_1(x)$, for $x \in \{0,1\}^s$, where $||$ denotes the concatenation of $G_0(x)$ and $G_1(x)$. For any secret key $k \in \{0,1\}^s$, define $F_k : \{0,1\}^n \to \{0,1\}^s$ by

$$F_k(x_1 \dots x_n) = G_{x_n}(G_{x_{n-1}}(\dots(G_{x_2}(G_{x_1}(k))))).$$

If G is a pseudo-random generator, then $F : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^s$ is a pseudo-random function.

1.2.1.2. Number-theoretic constructions

The Goldreich, Goldwasser and Micali construction is inefficient so many direct constructions were proposed based on some hard problems.

- Naor-Reingold PRF. In 1997, Naor and Reingold [NR97; NR04] proposed a (candidate) pseudo-random function family which takes inputs in $\{0,1\}^n$ (for some parameter n) and outputs an element in some (multiplicatively written) group $\mathbb{G}$ of prime order ℓ with generator g. The secret key is an n-dimensional vector $\boldsymbol{a} = (a_1, \dots, a_n) \in ((\mathbb{Z}/\ell\mathbb{Z})^*)^n$ and the Naor-Reingold function is defined as:

$$\begin{array}{rccl} f_{\boldsymbol{a}} & : & \{0,1\}^n & \longrightarrow & \mathbb{G} \\ & & (x_1, \dots, x_n) & \longmapsto & f_{\boldsymbol{a}}(x_1, \dots, x_n) = g^{\prod_{i=1}^{n} a_i^{x_i} \bmod \ell} \end{array}$$

  The evaluation of $f_{\boldsymbol{a}}$ is thus efficient since it consists only in n modular multiplications in $\mathbb{Z}/\ell\mathbb{Z}$ and one modular exponentiation in $\mathbb{G}$. It is shown in [NR97; NR04] that the Naor-Reingold function is pseudo-random provided that certain standard cryptographic assumptions about the hardness of breaking the Decision Diffie-Hellman assumption holds. In cryptography, two interesting choices for $\mathbb{G}$ are a subgroup of the multiplicative

group of a (prime) finite field and a subgroup of the points of an elliptic curve defined over a finite field. To lighten the notation, given an n-dimensional vector $\boldsymbol{a} = (a_1, \ldots, a_n) \in ((\mathbb{Z}/\ell\mathbb{Z})^*)^n$ and a variable x that will denote indifferently an n-bit string $(x_1, \ldots, x_n) \in \{0,1\}^n$ or an integer $x \in \{0, 1, \ldots, 2^n - 1\}$ (which implicitly defines $(x_1, \ldots, x_n) \in \{0,1\}^n$ the bit representation of x with extra leading zeros if necessary), we denote $\boldsymbol{a}^x$ the element in $\mathbb{F}_\ell$ defined by $\boldsymbol{a}^x = a_1^{x_1} \cdots a_n^{x_n} \bmod \ell$. With this notation, the Naor-Reingold function is simply defined by $f_{\boldsymbol{a}}(x) = g^{\boldsymbol{a}^x}$. Since proving that the Decision Diffie-Hellman assumption holds seems currently to be out of reach, several number-theoretic properties and complexity measures have been studied for the Naor-Reingold pseudo-random functions over finite fields as well as over elliptic curves: distribution (see [LSW14; Shp00b] and references therein), linear complexity (see [CGS10; GGI11; Shp00a; SS01]) and non-linear complexity (see [BGLS00]). These results are incomparable but they all support the assumption of the pseudo-randomness of the Naor-Reingold function.

- Dodis-Yampolskiy PRF. In 2005, Dodis and Yampolskiy [DY05] proposed an efficient pseudo-random function family which takes inputs in $\{1, \ldots, d\}$ (for some parameter $d \in \mathbb{N}$) and outputs an element in a group $\mathbb{G}$ (multiplicatively written) of prime order t with generator g. The secret key is a scalar $x \in \mathbb{Z}_t^*$ and the pseudo-random function is defined by:

$$\begin{array}{rccl} V_x & : & \{1, \ldots, d\} & \longrightarrow & \mathbb{G} \\ & & m & \longmapsto & V_x(m) = g^{\frac{1}{x+m}} \quad \text{if } x + m \neq 0 \bmod t \text{ and } 1_{\mathbb{G}} \text{ otherwise.} \end{array}$$

  The Dodis-Yampolskiy pseudo-random function family has found numerous applications in cryptography (e.g., for compact e-cash [CHL05] or anonymous authentication [CHK+06]). Dodis and Yampolskiy showed that their construction is a verifiable random function (that is a pseudo-random function that provides a non-interactively verifiable proof for the correctness if its output) and has some very attractive security properties, provided that some assumption about the hardness of breaking the so-called *Decision Diffie-Hellman Inversion* problem holds in $\mathbb{G}$ [DY05]. In practice, two interesting choices for the group $\mathbb{G}$ are a subgroup of the multiplicative group of any finite field (in particular, for the so-called *verifiable* Dodis-Yampolskiy pseudo-random function in groups equipped with a bilinear map [DY05]) or a subgroup of points of an elliptic curve defined over a prime finite field. Very few results supporting the Decision Diffie-Hellman Inversion assumption hardness were proven (contrary to the Naor-Reingold pseudo-random function family [NR04] for which numerous results are known, e.g. distribution [LSW14], linear complexity [GGI11] and non-linear complexity [BGLS00]).

1.2.2. Applications of pseudo-random functions

Pseudo-random functions have many applications in cryptography:

- they can be used for secret-key encryption as follows: Given a PRF F, pick random r, then for a secret key k and a message m, the ciphertext is $E_k(m) = (F_k(r) \oplus m, r)$. If F is a PRF, then E is semantically secure.
- they can be used to construct a secure block cipher by using the Luby-Rackoff construction.

- they can be used as message authentication codes (MACs) [Gol04, Chapter 1]: $MAC_k(m) = F_k(m)$.
- they can also be used for key-exchange.

1.3. Our results

1.3.1. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Functions

The polynomial interpolation is a question which is well studied for cryptographic believed hard functions to support their hardness. For breaking a hard function, it would be sufficient to have an easy multivariate polynomial f (namely a polynomial of low degree and low weight (the number of non-zero coefficients of the polynomial) which is efficiently computable) and which from some known information can approximate the function. For instance, for the Computational Diffie-Hellman assumption, given an element g of order ℓ such a polynomial could satisfy the relation:

$$f(g^x, g^y) = g^{xy}, \quad \text{for all } (x, y) \in S,$$

for a large subset $S \subseteq \{0, \ldots, \ell - 1\}^2$. Lower bounds on the degree or weight of polynomials interpolating the discrete logarithm problem (see [CS00]) or the Computational and Decision Diffie-Hellman assumption (see [MS01; Win01; KW04; Shp03] and references therein) are known. In order to break the security of the Naor-Reingold function, it would be sufficient to have a k-variate polynomial f over a finite field (of low degree or low weight) with $k \geq 1$ which reveals information on the functions values that is a k-variate polynomial f satisfying: $(f(g^{\boldsymbol{a}^{x^1}}, \ldots, g^{\boldsymbol{a}^{x^k}}) = g^{\boldsymbol{a}^{x^{k+1}}}$, for all $\boldsymbol{a} = (a_1, \ldots, a_n) \in S$ for a large subset $S \subseteq (\mathbb{F}_\ell^*)^n$, and for some known values $x^1, \ldots, x^{k+1} \in \{0, \cdots, 2^n - 1\}$) or $(f(g^{\boldsymbol{a}^x}, g^{\boldsymbol{a}^{x+t_1}}, \ldots, g^{\boldsymbol{a}^{x+t_{k-1}}}) = g^{\boldsymbol{a}^{x+t_k}}$ for many integers $x \in \{0, 1, \ldots, 2^n - 1\}$, and for some known values $t_1, \ldots, t_k$ and for some known secret key $\boldsymbol{a}$). We refer the first case to *the polynomial interpolation with variable secret key* and the second case to *the polynomial interpolation with fixed secret key*. Our first constribution [MV17c; MV17b] is that low weight or degree k-variate polynomial cannot reveal information on the functions values. We consider the settings of a finite field and an elliptic curve and in both cases, we obtain lower bounds on the degree of polynomials interpolating the Naor-Reingold function with a fixed secret key and variable secret key.

1.3.2. Distribution and Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function

The second contribution [MV16] of this thesis is about the distribution of the Dodis-Yampolskiy pseudo-random function over finite fields and over elliptic curves and the lower bounds on the degree of polynomials which interpolate these functions.
We prove that for almost all values of parameters, the Dodis-Yampolskiy pseudo-random function produces a uniformly distributed sequence. Our result is based on some recent bounds on character sums with exponential functions. We also prove that a low-degree univariate polynomial cannot reveal the secret key x when evaluated at $V_x(m)$ (for some integer $m \in \{1, \ldots, d\}$) for all x. These results can be regarded as first complexity lower bounds on the pseudo-randomness of the Dodis-Yampolskiy function families.

1.3.3. Inferring a Linear Congruential Generator and a Power Generator on Elliptic Curves

As a third contribution [Mef16], we analyze the security of the elliptic curve linear congruential generator and of the elliptic curve power generator. We infer the EC-LCG sequence and the EC-PG sequence using Coppersmith's method for calculating the small roots of multivariate polynomials modulo an integer. In the case where the *composer* is known, we showed that the EC-LCG is insecure if a proportion of at most $1/5$ of the least significant bits of two consecutive values U_0 and U_1 of the sequence is hidden. This improves the previous bound $1/6$ of Gutierrez and Ibeas [GI07]. We further improve this result by considering several consecutive values of the sequence. We showed that the EC-LCG is insecure if a proportion of at most $3/11$ of the least significant bits of these values is hidden. To prevent the attacks of [GI07], one could output only the most significant bits of the abscissa of consecutive multiple values U_{kn} (for some fixed integer k) of the sequence. We consider this setting and use summation polynomials to infer the EC-LCG. These polynomials were used to solve elliptic curve discrete logarithm problem and we use it in this thesis to infer the EC-LCG when the attacks of [GI07] cannot work. We then showed that the EC-LCG is insecure if a proportion of at most $1/8$ of the least significant bits of two values $X(U_0)$ and $X(U_k)$ is hidden, where $X(P)$ denotes the abscissa of the point P on the curve. We further improve this result by considering several values U_{kn}, $n \in \mathbb{N}$ of the sequence. We showed that the EC-LCG is insecure if a proportion of at most $1/4$ of the least significant bits of the abscissa of these values is hidden. In the case where the *composer* is unknown, we showed that the EC-LCG is insecure if a proportion of at most $1/24$ of the least significant bits of two consecutive values U_0 and U_1 of the sequence is hidden. This improves the previous bound $1/46$ of Gutierrez and Ibeas [GI07]. We further improve this result by considering sufficiently many consecutive values of the sequence. We showed that the EC-LCG is insecure if a proportion of at most $1/8$ of the least significant bits of these values is hidden. Finally, we also showed that the EC-PG is insecure if a proportion of at most $1/2e^2$ of the least significant bits of the abscissa of two consecutive values V_0 and V_1 of the sequence is hidden. We improve this bound by considering several consecutive values of the sequence and we showed that the EC-PG is insecure if a proportion of at most $1/e^2$ of the least significant bits of the abscissa of these values is hidden. To our knowledge such a result is not known in the literature for the EC-PG.

1.3.4. Lattice Attacks on Pairing-Based Signatures

The pairing-based signature schemes are very well-suited for resource-limited devices since they produce short signatures and their generation involves only one scalar multiplication on an elliptic curve. In the recent years, theoretical attacks against elliptic curves have shown little improvements whereas *side-channel attacks* became a major threat against elliptic curves implementations [Koc96; KJJ99]. These attacks are based on information gained from the physical leakage of a cryptosystem implementation (such as timing information, power consumption or electromagnetic leaks). For public-key cryptography on embedded systems, the core operation is usually group exponentiation – or scalar multiplication on elliptic curves – which is a sequence of group operations derived from the private-key that may reveal secret bits to an attacker (on an unprotected implementation). This can be the case when computing the exponent in order to compute the output of the Dodis-Yampolskiy pseudo-random function and more generally in well-known pairing-based signatures (Sakai-

Kasahara signatures [SK03], Boneh-Boyen signatures [BB04a] and Gentry signatures [Gen06]) based on the *exponent-inversion* framework. Our last contribution [MV17a] is concerned with lattice attacks on these well-known Pairing-Based signatures and our approach is similar to lattice attacks [HS01; NS02; NS03] combined with template attacks [MHMP13] that were proposed against the standardized signature scheme DSA and ECDSA. We present lattice-based polynomial-time (heuristic) algorithms that recover the signer's secret in popular pairing-based signatures when used to sign several messages under the assumption that blocks of consecutive bits of the corresponding exponents are known by the attacker. Our techniques relies upon Coppersmith's methods and apply to all signatures in the so-called *exponent-inversion* framework in the standard security model (*i.e.* Boneh-Boyen and Gentry signatures) as well as in the random oracle model (*i.e.* Sakai-Kasahara signatures). The efficiency of our (heuristic) attacks has been validated experimentally.

1.4. Organization

This thesis is divided in two parts. The first part deals with some complexities measures of Naor-Reingold and Dodis-Yampolskiy Pseudo-Random Function and the second part deals with lattice attacks on pseudo-random generators and on pairing-based signatures based on *exponent-inversion* framework. The first part includes Chapters 2 , 3 and 4. Chapter 2 introduces some mathematical notions used throughout this thesis. Chapter 3 study the polynomial interpolation of the Naor-Reingold pseudo-random functions. Chapter 4 is about the distribution and polynomial interpolation of the Dodis-Yampolskiy pseudo-random function. The second part includes Chapters 5, 6 and 7. Chapter 5 provides short descriptions of Coppersmith's methods and some analytic combinatorics to ease the methods. Chapter 6 presents the attacks on the linear congruential generator and the power generator on elliptic curves and Chapter 7 deals with lattice attacks on some pairing-based signatures. Finally, Chapter 8 concludes this thesis and raises some open questions.

Part I.

Complexity Measures of Pseudo-Random Functions

Chapter 2.

Preliminaries

In this Chapter, we introduce the notation used throughout this manuscript. We recall some results on finite fields and elliptic curves that we use in Chapters 3 and 4. We also provide explicit upper-bounds for exponential sums with consecutive modular roots over a finite field and for analogous exponential sums over elliptic curves. These bounds will help us to study the distribution of the Dodis-Yampolskiy pseudo-random function in Chapter 4. We conclude this Chapter by recalling some known lower bounds on the polynomial interpolation on the discrete logarithm modulo a prime number p since the techniques will help us study the polynomial interpolation of the pseudo-random functions we consider in Chapters 3 and 4.

Contents

2.1. Notation **18**
2.2. Finite fields **18**
2.3. Elliptic Curves **19**
2.3.1. Definition and addition law 19
2.3.2. Division polynomials of elliptic curves 20
2.3.3. Summation polynomials 22
2.4. Exponential Sums **22**
2.4.1. Finite Fields and Exponential Sums 23
2.4.2. Elliptic Curves and Exponential Sums 25
2.5. Polynomial Approximation of the Discrete Logarithm **26**

2.1. Notation

In this section, we recall the general notations that we use throughout this thesis. We denote by $\mathbb{Z}$ the set of integers and by $\mathbb{N}$ the set of non-negative integers. If z is a positive real number, $\log z$ denotes its binary logarithm and $|z|$ its absolute value. If X, Y are two real numbers, $X = O(Y)$ and $X \ll Y$ denote that $|X| \leq cY$, where c is a positive constant. If S is a finite set, $|S|$ or $\sharp S$ denotes its size. $\mathbb{F}_q$ denotes a finite field of q elements and if q is a prime number, then the elements of $\mathbb{F}_q$ are identified with the set of integers $\{0, \cdots, q-1\}$. $\mathbb{F}_q^*$ denotes the multiplicative group of $\mathbb{F}_q$ that is the set of the invertible elements of $\mathbb{F}_q$ which is of size $q-1$. If $\mathbb{F}_q$ is a finite field, $\overline{\mathbb{F}_q}$ denotes the algebraic closure of $\mathbb{F}_q$. For a positive integer $m \geq 2$, $(\mathbb{Z}_m, +, \cdot)$ or $\mathbb{Z}_m$ denotes the ring of integers modulo m which can be identified with the set of integers $\{0, \ldots, m-1\}$ and by $\mathbb{Z}_m^*$ the set of the invertible elements of $\mathbb{Z}_m$ which consists of integers $k \in \mathbb{Z}_m$, with $\gcd(k, m) = 1$. If R is a ring, then $R[X_1, \ldots, X_n]$ denotes the ring of multivariate polynomials with n indeterminates with coefficients in R. When $n = 1$, $R[X]$ denotes the ring of univariate polynomials with coefficients in R. If $f \in R[X_1, \ldots, X_n]$, then $w(f)$ denotes its *weight* (or sparsity) that is the number of its non-zero coefficients and $\deg(f)$ its degree.
For a real z, we use the notation $e(z) = exp(2\pi iz)$ and $e_m(z) = exp(2\pi iz/m)$

2.2. Finite fields

In this section, we collect some statements about finite fields that we will need throughout this thesis. The following lemma gives a lower bound of the weight of a univariate polynomial and we will need it in the next two chapters.

Lemma 2.2.1 ([LW02])**.** *Let $\gamma \in \mathbb{F}_p$ be an element of order ℓ and $F(X) \in \mathbb{F}_p[X]$ be a non-zero polynomial of degree at most $\ell - 1$ with at least b zeros of the form γ^x with $0 \leq x \leq \ell - 1$. The weight of $F(X)$ satisfies*

$$w(F) \geq \frac{\ell}{\ell - b}$$

We will use the two following lemmas (see [KW04]) in Chapter 3 where Lemma 2.2.3 is a generalization of Lemma 2.2.1.

Lemma 2.2.2. *Let D be an integral domain, $n \in \mathbb{N}$ and $f \in D[X_1, \ldots, X_n]$ a polynomial of total degree d with at least N zeros in S^n. If f is not the zero polynomial, then we have*

$$d \geq \frac{N}{|S|^{n-1}}.$$

Lemma 2.2.3. *Let $\gamma \in \mathbb{F}_q$ be an element of order d, G the group generated by γ, n a positive integer, and $f \in \mathbb{F}_q[X_1, \ldots, X_n]$ be a nonzero polynomial of local degree at most $d-1$ in each variable with at least N zeros in G^n. Then for the weight $w(f)$ of f, we have :*

$$w(f) \geq \frac{d^n}{d^n - N}.$$

2.3. Elliptic Curves

In this section, we collect some results on elliptic curves. Later we will consider the settings where the pseudo-random functions in Chapters 3 and 4 are defined over an elliptic curve over a prime finite field and a pseudorandom number generator defined over an elliptic curve in Chapter 6.

2.3.1. Definition and addition law

Let $p > 3$ be an odd prime number, an elliptic curve E defined over $\mathbb{F}_p$ (for more details on elliptic curves, see [BSS99; Was08]) is a rational curve given by the following Weierstrass equation

$$y^2 = x^3 + Ax + B, \quad A, B \in \mathbb{F}_p, \quad 4A^3 + 27B^2 \neq 0.$$

The set $E(\mathbb{F}_p)$ of the points of the curve defined over $\mathbb{F}_p$ (including the special point O at infinity) has a group structure (denoted additively) with an appropriate composition rule $\oplus$ where O is the neutral element. Let $E/\mathbb{F}_p : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_p$. For two points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$, with $P, Q \neq O$, the point $R = P \oplus Q$ is geometrically obtained as follows:

- Draw the line L through P and Q or (the tangent to the curve E at P if $P = Q$)
- L intersects E in a third point R'
- Reflect R' across the x-axis to obtain R.

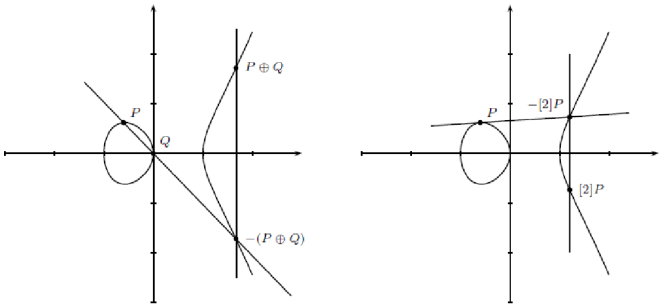

Figure 2.1.: *Adding two points on an elliptic curve over* $\mathbb{R}$

The formulas for x_R and y_R with $P \oplus Q = R = (x_R, y_R)$ are given as follows:

- If $x_P \neq x_Q$, then

$$x_R = m^2 - x_P - x_Q, \quad y_R = m(x_P - x_R) - y_P, \quad \text{where } m = \frac{y_Q - y_P}{x_Q - x_P} \tag{2.1}$$

- If $x_P = x_Q$ but $y_P \neq y_Q$, then $R = O$
- If $P = Q$ and $y_P \neq 0$, then

$$x_R = m^2 - 2x_P, \quad y_R = m(x_P - x_R) - y_P, \quad \text{where } m = \frac{3x_Q^2 + a}{2y_P}$$

- If $P = Q$ and $y_P = 0$, then $R = O$.

Given a P point of a curve E with prime order ℓ (with $\ell \mid \mid E(\mathbb{F}_p)|$), we denote $[r]P$ the scalar multiplication, i.e. the adding of the point P to itself r times:

$$[r]P = \underbrace{P \oplus \cdots \oplus P}_{r \text{ times}}$$

(and $[r]P = -([-r]P)$ for $r \leq 0$).

2.3.2. Division polynomials of elliptic curves

We recall some basic facts on division polynomials of elliptic curves (see [Was08], Section 3.2). They provide a way to calculate multiples of points on elliptic curves. The *division polynomials* $\psi_m(X, Y) \in \mathbb{F}_p[X, Y]/(Y^2 - X^3 - AX - B)$, $m \geq 0$, are recursively defined by:

$$\begin{aligned}
\psi_0 &= 0 \\
\psi_1 &= 1 \\
\psi_2 &= 2Y \\
\psi_3 &= 3X^4 + 6AX^2 + 12BX - A^2 \\
\psi_4 &= 4Y(X^6 + 5AX^4 + 20BX^3 - 5A^2X^2 - 4ABX - 8B^2 - A^3) \\
\psi_{2m+1} &= \psi_m + 2\psi_m^3 - \psi_{m-1}\psi_{m+1}^3\,, \quad m \geq 2 \\
\psi_{2m} &= \psi_m(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2)/\psi_2\,, \quad m \geq 3,
\end{aligned}$$

where ψ_m is an abbreviation for $\psi_m(X,Y)$. If m is odd, then $\psi_m(X,Y) \in \mathbb{F}_p[X]$ is univariate and if m is even then $\psi_m(X,Y) \in \psi_2(X,Y)\mathbb{F}_p[X] = 2Y\mathbb{F}_p[X]$. Therefore, as $\psi_2^2(X,Y) = 4(X^3 + AX + B)$, we have $\psi_m^2(X,Y) \in \mathbb{F}_p[X]$ and $\psi_{m-1}(X,Y)\psi_{m+1}(X,Y) \in \mathbb{F}_p[X]$. In particular, we may write $\psi_{2m+1}(X)$ and $\psi_m^2(X)$.

As mentioned above, the division polynomials can be used to calculate multiples of a point on the elliptic curve E. Let $P = (x,y) \in E$ with $P \neq O$, then the coordinates of $[m]P$ if $[m]P \neq O$ are given by

$$[m]P = \left(\frac{\theta_m(x)}{\psi_m^2(x)}, \frac{\omega_m(x,y)}{\psi_m^3(x,y)}\right),$$

where $\theta_m(X) = X\psi_m^2 - \psi_{m-1}\psi_{m+1}$ and $\omega_m(X,Y) = (4Y)^{-1}(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2)$. The zeros of the denominator $\psi_m^2(X)$ are exactly the first coordinates of the non-trivial m-torsion points, i.e, the points $Q = (x,y) \in \overline{\mathbb{F}_p}^2 \setminus \{O\}$ on E with $[m]Q = O$. Note, that these points occur in pairs $Q = (x,y)$ and $-Q = (x,-y)$, which coincide only if $2Q = O$, i.e, if x is a zero of $\psi_2^2(X)$.

We recall that the group of m-torsion points $E[m]$, for an elliptic curve E defined over a field of characteristic p, is isomorphic to $(\mathbb{Z}/m\mathbb{Z})^2$ if $p \nmid m$ and to a proper subgroup of $(\mathbb{Z}/m\mathbb{Z})^2$ if $p \mid m$. If m is a power of p then $E[m]$ is either isomorphic to $(\mathbb{Z}/m\mathbb{Z})$ or to $\{O\}$. Accordingly, the degree of $\psi_m^2(X)$ is $m^2 - 1$ if $p \mid m$ and strictly less than $m^2 - 1$ otherwise. By induction one can show that $\theta_m(X) \in \mathbb{F}_p[X]$ is monic of degree m^2.

In the next two chapters, we will make use of the two following technical lemmas.

Lemma 2.3.1. *Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve over $\mathbb{F}_p$ with $A \neq 0$ and $B \neq 0$. Let $F(X) \in \mathbb{F}_p[X]$ be a non-constant polynomial with $F(X) \neq X$ and $\deg(F) < p$. Then there exists $\alpha \in \overline{\mathbb{F}_p}$ such that $\psi_2^2(F(\alpha)) = 0$ and $\psi_2^2(\alpha) \neq 0$.*

Proof. There are exactly three distinct zeros $\alpha_1, \alpha_2, \alpha_3 \in \overline{\mathbb{F}_p}$ of $\psi_2^2(X) = 4(X^3 + AX + B)$. For all index $i \in \{1,2,3\}$, there exists at least one $\beta_i \in \overline{\mathbb{F}_p}$ such that $F(\beta_i) = \alpha_i$, because F is not a constant polynomial. Since for all $i,j \in \{1,2,3\}$, $i \neq j$, we have $\alpha_i \neq \alpha_j$, then the system $F(X) = \alpha_i$ and $F(X) = \alpha_j$ has no solution. It follows that the polynomial $\psi_2^2(F(X))$ has at least three different zeros.

Let $d < p$ denote the degree of F and let us suppose that there does not exist $\alpha \in \overline{\mathbb{F}_p}$ such that $\psi_2^2(F(\alpha)) = 0$ and $\psi_2^2(\alpha) \neq 0$. Then we have that $\psi_2^2(F(X))$ has exactly three zeros which are the zeros of $\psi_2^2(X)$. If $d = 1$, putting $F(X) = aX + b$, we obtain that the polynomials $X^3 + AX + B$ and $a^3X^3 + 3a^2bX^2 + (3ab^2 + aA)X + b^3 + Ab + B$ have exactly the same three zeros. We then have $3a^2b = 0$ and $a \neq 0$. Thus $b = 0$, and if we suppose $A \neq 0$ and $B \neq 0$, we have $a = 1$ which is impossible since $F(X) \neq X$. If $d \geq 2$, for all $i \in \{1,2,3\}$, the equation $F(X) = \alpha_i$ has exactly one solution γ_i of multiplicity d which is one of $\{\alpha_1, \alpha_2, \alpha_3\}$. Then γ_1 and γ_2 are the zeros of the $(d-1)$-derivative of $F(X)$ which is of degree 1 and this is impossible because $\gamma_1 \neq \gamma_2$. Hence in all cases, we obtain a contradiction. So there exists $\alpha \in \overline{\mathbb{F}_p}$ such that: $\psi_2^2(F(\alpha)) = 0$ and $\psi_2^2(\alpha) \neq 0$. □

Lemma 2.3.2. *Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve over $\mathbb{F}_p$. Let $k = 2^i$ for $i > 0$ an integer. Let $F(X) \in \mathbb{F}_p[X]$ be a non-constant polynomial with $\deg(F) \geq 2$. Then there exists $\alpha \in \overline{\mathbb{F}_p}$ such that $\psi_k^2(F(\alpha)) = 0$ and $\psi_k^2(\alpha) \neq 0$.*

Proof. The univariate polynomial $\psi_k^2(X)$ has at least $k^2/2$ distinct zeros because $p \nmid k$. For all α such that $\psi_k^2(\alpha) = 0$, there exists at least one $\beta \in \overline{\mathbb{F}_p}$ such that $F(\beta) = \alpha$ and two such

roots β (corresponding to two different α) are different. Let us suppose that there does not exist $\alpha \in \overline{\mathbb{F}_p}$ such that $\psi_k^2(F(\alpha)) = 0$ and $\psi_k^2(\alpha) \neq 0$. Since $\deg(F) \geq 2$, it follows that the equation $F(X) = \alpha$, for some α zero of $\psi_k^2(X)$ has at least two different solutions using the ideas of the later proof. Thus we obtain a contradiction and the desired result follows. □

2.3.3. Summation polynomials

Index calculus algorithm [Adl79] is used to solve the discrete logarithm problem (DLP) over finite fields in sub-exponential time. For the elliptic curve discrete logarithm problem (EC-DLP), only exponential time algorithms were known to solve the problem. The idea of using index calculus to solve the EC-DLP was proposed by Semaev [Sem04] and Gaudry [Gau09]. They proposed to decompose points by computing the zeroes of summation polynomials (which were first introduced in 2004 by Semaev [Sem04]). For some curves over extension finite fields, Gröbner basis were used to compute zeroes of such polynomials or their generalizations and sub-exponential algorithms solving the EC-DLP were obtained (see [Gau09; Die11; JV12; FPPR12]). For elliptic curves over prime fields or binary fields of prime extension degree, [Sem15] proposed improved algorithms solving the EC-DLP which compute the zeroes of summation polynomials by solving a system of boolean equations. In this thesis, we use summation polynomials in another context, namely to infer the elliptic curve linear congruential generator in Chapter 6. Below, we recall the summation polynomials (see [Sem04] for details and proofs). Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over a finite field $\mathbb{F}_q$ (where q is a power of a prime number $p \neq 2, 3$). For $n \in \mathbb{N}$, $n \geq 2$, the n^{th} summation polynomial $f_n(X_1, X_2, \ldots, X_n) \in \mathbb{F}_q[X_1, \ldots, X_n]$ for E which is related to the arithmetic operation on E has the following property:
$f_n(x_1, \ldots, x_n) = 0$, $x_i \in \overline{\mathbb{F}_q}$ if and only if (there exists $y_1, \ldots, y_n \in \overline{\mathbb{F}_q}$ such that $(x_1, y_1), \ldots, (x_n, y_n) \in E$ and $(x_1, y_1) \oplus \cdots \oplus (x_n, y_n) = O$), where O is the point at infinity.

Lemma 2.3.3 ([Sem04])**.** *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over a finite field $\mathbb{F}_q$ (where q is a power of a prime number $p \neq 2, 3$)). The summation polynomials for E are given as follows:*

$$\begin{gathered} f_2(X_1, X_2) = X_1 - X_2 \\ f_3(X_1, X_2, X_3) = (X_1 - X_2)^2 X_3^2 - 2\left((X_1 + X_2)(X_1X_2 + a) + 2b\right) X_3 + \\ (X_1X_2 - a)^2 - 4b(X_1 + X_2) \\ f_n(X_1, \ldots, X_n) = Res_X(f_{n-k}(X_1, \ldots, X_{n-k-1}, X), f_{k+2}(X_{n-k}, \ldots, X_n, X)), \quad n \geq 4 \end{gathered}$$

where the last equality holds for any constant k with $1 \leq k \leq n - 3$. The n^{th} summation polynomial f_n is an irreducible symmetric polynomial of degree 2^{n-2} in each variable X_i for any $n \geq 2$.

2.4. Exponential Sums

In this section, we collect some statements about exponential sums over finite fields and elliptic curves that we use to study he distribution of the Dodis-Yampolskiy pseudo-random function in Chapter 4. We provide explicit upper bounds for exponential sums with consecutive modular roots over a finite field and for analogous exponential sums over elliptic curves [Shp09a; OS11]. The bound for exponential sums with consecutive modular roots over a general finite field is easily derived from [Shp09a] and may be of independent interest.

2.4.1. Finite Fields and Exponential Sums

Let p be an odd prime number, $r \geq 1$ an integer. Let $g \in \mathbb{F}_{p^r}^*$ of order t, and ψ be a non-trivial character of $\mathbb{F}_{p^r}$. For $a \in \mathbb{F}_{p^r}^*$ and $b \in \mathbb{Z}_t$, we define the sum:

$$S_{a,b} = \sum_{n \in \mathbb{Z}_t^*} \psi(ag^{1/n}) e_t(bn).$$

In the following lemmas, the implied constants in the symbols "$\ll$" may occasionally depend on the integer parameters k, ℓ but are absolute otherwise.

In [BS08] Bourgain and Shparlinski proved, when $r = 1$, that for any $\epsilon > 0$, there exists $\delta > 0$ such that for $t \geq p^\epsilon$, we have the bound $S_{a,b} \ll t^{1-\delta}$. Shparlinski [Shp09a](Theorem 3.1) gave an explicit form of this result (again when $r = 1$) for relatively large values of t; in the case $t = p^{1+o(1)}$, it takes the form $S_{a,b} \ll t^{127/128+o(1)}$. Using Shparlinski's methods, we generalize this bound on $S_{a,b}$ for any $r \geq 1$.

Proposition 2.4.1. *For any integers $k \geq 2$, $\ell \geq 1$ we have for $t \geq q^{1/2}(\log q)^2$:*

$$S_{a,b} \leq t^{1-\alpha_{k,\ell}} q^{\beta_{k,\ell}+o(1)},$$

where $\alpha_{k,\ell} = \frac{1}{2(2k+\ell)} - \frac{1}{4k\ell}$ *and* $\beta_{k,\ell} = \frac{1}{4(2k+\ell)}$.

The proof follows the one of [Shp09a]). In the proof of the Proposition 2.4.1, we use the two following lemmas.

Lemma 2.4.2 is the classical Weil bound for exponential sums which can be found in [Wei48; NW01].

Lemma 2.4.2. *Let $F(x)$ be a non constant polynomial in $\mathbb{F}_q[x]$ such that $F(x) \neq h(x)^p - h(x)$ for any $h(x) \in \overline{\mathbb{F}_q}(x)$, where q is a power of p. We have*

$$\left| \sum_{x \in \mathbb{F}_q} \psi(F(x)) \right| \leq (\deg(F) - 1) q^{1/2}$$

We then deduce the following simple lemma:

Lemma 2.4.3. *For any pairwise distinct positive integers $1 \leq r_1, \ldots, r_v \leq R$, we have*

$$\max_{\substack{(a_1,\ldots,a_v) \in \mathbb{F}_{p^r}^v \\ (a_1,\ldots,a_v) \neq (0,\ldots,0)}} \left| \sum_{n=1}^{t} \psi\left(\sum_{i=1}^{v} a_i g^{r_i n} \right) \right| \leq R q^{1/2}.$$

Proof. Let $s = (q-1)/t$. We have $g = \theta^s$, where θ is a primitive root in $\mathbb{F}_q$ and

$$\begin{aligned} \sum_{n=1}^{t} \psi\left(\sum_{i=1}^{v} a_i g^{r_i n} \right) = \sum_{n=1}^{t} \psi\left(\sum_{i=1}^{v} a_i \theta^{s r_i n} \right) &= \frac{1}{s} \sum_{n=1}^{q-1} \psi\left(\sum_{i=1}^{v} a_i \theta^{s r_i n} \right) \\ &= \frac{1}{s} \left(\sum_{x \in \mathbb{F}_q} \psi\left(\sum_{i=1}^{v} a_i x^{s r_i} \right) - 1 \right) \end{aligned}$$

Applying Lemma 2.4.2 with the polynomial $F(x) = \sum_{i=1}^{v} a_i x^{s r_i}$, we obtain :

$$\max_{\substack{(a_1,\ldots,a_v) \in \mathbb{F}_{p^r}^v \\ (a_1,\ldots,a_v) \neq (0,\ldots,0)}} \left| \sum_{n=1}^{t} \psi\left(\sum_{i=1}^{v} a_i g^{r_i n} \right) \right| \leq \frac{1}{s}((Rs-1)q^{1/2} + 1) \leq R q^{1/2}.$$

□

Now we are ready to prove Proposition 2.4.1.

Proof. For any integer $k \geq 2$, we have

$$S_{a,b}{}^k = \sum_{n_1,\dots,n_k \in \mathbb{Z}_t^*} \psi\left(a \sum_{j=1}^{k} g^{1/n_j}\right) e_t\left(b \sum_{j=1}^{k} n_j\right).$$

For $m \in \mathbb{Z}_t$, we collect together the terms with $n_1 + \dots + n_k \equiv m \bmod t$, getting:

$$|S_{a,b}|^k \leq \sum_{m \in \mathbb{Z}_t} \left| \sum_{\substack{n_1,\dots,n_k \in \mathbb{Z}_t^* \\ n_1+\dots+n_k \equiv m \bmod t}} \psi\left(a \sum_{j=1}^{k} g^{1/n_j}\right) \right|.$$

By the Cauchy inequality, we can upper-bound $|S_{a,b}|^{2k}$ by

$$t \sum_{m \in \mathbb{Z}_t} \left| \sum_{\substack{n_1,\dots,n_k \in \mathbb{Z}_t^* \\ n_1+\dots+n_k \equiv m \bmod t}} \psi\left(a \sum_{j=1}^{k} g^{1/n_j}\right) \right|^2 = t \sum_{(n_1,\dots,n_{2k}) \in N_k} \psi\left(a \sum_{j=1}^{2k} (-1)^j g^{1/n_j}\right)$$

where the outside summation is taken over the set of vectors

$$N_k = \{(n_1, \dots, n_{2k}) \in (\mathbb{Z}_t^*)^{2k} : n_1 + \dots + n_{2k-1} \equiv n_2 + n_4 + \dots + n_{2k} \bmod t)\}.$$

One can see that for any $m \in \mathbb{N}$ with $\gcd(m, t) = 1$, we have

$$\sum_{(n_1,\dots,n_{2k}) \in N_k} \psi\left(a \sum_{j=1}^{2k} (-1)^j g^{1/n_j}\right) = \sum_{(n_1,\dots,n_{2k}) \in N_k} \psi\left(a \sum_{j=1}^{2k} (-1)^j g^{m/n_j}\right).$$

Let us fix some parameter Q with $Q \geq 2 \log t$. Let $\mathcal{Q}$ be the set of primes $m \leq Q$ with $\gcd(m, t) = 1$. Averaging over all $m \in \mathcal{Q}$, we obtain

$$|S_{a,b}|^{2k} \leq \frac{t}{\sharp \mathcal{Q}} \sum_{m \in \mathcal{Q}} \sum_{(n_1,\dots,n_{2k}) \in N_k} \psi\left(a \sum_{j=1}^{2k} (-1)^j g^{m/n_j}\right).$$

The number $w(t)$ of prime divisors of t satisfies $w(t) \leq (1 + o(1))(\log t)/(\log \log t)$ (which can be seen from the trivial inequality $w(t)! \leq t$ and the Stirling formula). By the prime number theorem, we have (since $Q \geq 2 \log t$):

$$\sharp \mathcal{Q} \geq (1 + o(1)) \frac{Q}{\log Q} - (1 + o(1)) \frac{\log t}{\log(\log t)} \geq 0.5 \frac{Q}{\log Q},$$

provided that t is large enough. We have $\sharp N_k \leq t^{2k-1}$. Using the Hölder inequality and then

extending the region of summation, we obtain that for any integer $\ell \geq 1$:

$$\begin{aligned}
|S_{a,b}|^{4k\ell} &\leq \frac{t^{2\ell}}{\sharp\mathcal{Q}^{2\ell}}(\sharp N_k)^{2\ell-1} \sum_{n_1,\dots,n_{2k}\in\mathbb{Z}_t^*} \left|\sum_{m\in\mathcal{Q}} \psi\left(a\sum_{j=1}^{2k}(-1)^j g^{m/n_j}\right)\right|^{2\ell} \\
&\ll \frac{t^{4k\ell-2k+1}\log^{2\ell} Q}{Q^{2\ell}} \sum_{n_1,\dots,n_{2k}=1}^{t} \left|\sum_{m\in\mathcal{Q}} \psi\left(a\sum_{j=1}^{2k}(-1)^j g^{mn_j}\right)\right|^{2\ell} \\
&= \frac{t^{4k\ell-2k+1}\log^{2\ell} Q}{Q^{2l}} \sum_{n_1,\dots,n_{2k}=1}^{t} \sum_{m_1,\dots,m_{2\ell}\in\mathcal{Q}} \psi\left(a\sum_{j=1}^{2k}\sum_{h=1}^{2\ell}(-1)^{j+h} g^{m_h n_j}\right) \\
&= \frac{t^{4k\ell-2k+1}\log^{2\ell} Q}{Q^{2\ell}} \sum_{m_1,\dots,m_{2\ell}\in\mathcal{Q}} \left|\sum_{n=1}^{t} \psi\left(a\sum_{h=1}^{2\ell}(-1)^h g^{m_h n}\right)\right|^{2k}.
\end{aligned}$$

For $O(\sharp\mathcal{Q}^\ell)=O(Q^\ell \log^{-\ell} Q)$ tuples $(m_1,\dots,m_{2\ell}) \in \mathcal{Q}^{2\ell}$ such that the tuple of the elements on the odd positions $(m_1,\dots,m_{2\ell-1})$ is a permutation of the elements on the even positions $(m_2,\dots,m_{2\ell})$, we estimate the inner sum trivially as t.
For the remaining $O((\sharp Q)^{2\ell}) = O(Q^{2\ell}(\log Q)^{-2\ell})$ tuples, we use the bound of Lemma 2.4.3. Therefore,

$$\begin{aligned}
|S_{a,b}|^{4k\ell} &\ll \frac{t^{4k\ell-2k+1}\log^{2\ell} Q}{Q^{2l}}(Q^\ell \log^{-\ell} Q t^{2k} + Q^{2\ell}\log^{-2\ell} Q (Q q^{1/2})^{2k}) \\
&= t^{4k\ell-2k+1}(Q^{-\ell}\log^\ell Q t^{2k} + Q^{2k} q^k).
\end{aligned}$$

Taking $Q = 2t^{2k/(2k+\ell)} q^{-k/(2k+\ell)} (\log q)^{\ell/(2k+\ell)}$ and if $t \geq q^{1/2}(\log q)^2$, one can see that $Q \geq 2\log t$ and we obtain

$$|S_{a,b}|^{4k\ell} \ll t^{4k\ell-(2k\ell-2k-\ell)/(2k+\ell)} q^{k\ell/(2k+\ell)} (\log q)^{\ell/(2k+\ell)}$$

and the result follows. □

2.4.2. Elliptic Curves and Exponential Sums

Let E be an elliptic curve and $P \in E(\mathbb{F}_p)$ be a point of order $t \geq 1$. For $a \in \mathbb{F}_p^*$ and $b \in \mathbb{Z}_t$, we define the following exponential sum which is an analogous of the sum $S_{a,b}$ over an elliptic curve:

$$\hat{S}_{a,b} = \sum_{n\in\mathbb{Z}_t^*} e_p\left(aX\left(\left[\frac{1}{n}\right]P\right)\right) e_t(bn),$$

where $X(P)$ denotes the abscissa of the point P.

In [OS11, Theorem 6], Ostafe and Shparlinski obtained an upper-bound on $\hat{S}_{a,b}$ (with $H(X) = X^{-1}$ following the notation from [OS11]):

Proposition 2.4.4 ([OS11]). *For any integers $k \geq 2$, $\ell \geq 1$ we have for $t \geq q^{1/2}(\log q)^2$:*

$$\hat{S}_{a,b} \leq t^{1-\alpha_{k,\ell}} p^{\beta_{k,\ell}+o(1)},$$

where $\alpha_{k,\ell} = \frac{1}{2(4k+\ell)} - \frac{1}{4k\ell}$ *and* $\beta_{k,\ell} = \frac{1}{4(4k+\ell)}$.

In the proof, we need the following lemma which is a special case of the bound of Bombieri see [Bom66]:

Lemma 2.4.5. *Let $s \geq 1$ be an integer. For any integers $1 \leq u_1 < \cdots < u_s \leq U$ and elements $c_1, \ldots, c_s \in \mathbb{F}_p$ with $c_s \neq 0$, the following bound holds:*

$$\sum_{R \in \mathcal{H}, R \neq O} e_p\left(\sum_{i=1}^{s} c_i X(u_i R)\right) \ll U^2 p^{1/2}$$

where $\mathcal{H}$ is a subgroup of $E(\mathbb{F}_p)$ of order t such that $\gcd(t, u_1, \ldots, u_s) = 1$.

Next we then prove Proposition 2.4.4.

Proof. Following the same path as in the finite fields case by replacing the multiplicative law by an additive one and applying the Lemma 2.4.5 rather than Lemma 2.4.3, we obtain:

$$|S_{a,b}|^{4k\ell} \ll t^{4k\ell-2k+1}(Q^{-\ell}\log^{\ell} Q t^{2k} + Q^{4k}p^k).$$

Taking

$$Q = 2t^{2k/(4k+\ell)}p^{-k/(4k\ell+\ell)}(\log p)^{\ell/(4k+\ell)},$$

we obtain

$$\left|\hat{S}_{a,b}\right|^{4k\ell} \ll t^{4k\ell-2k+1+8k^2/(4k+\ell)}p^{k\ell/(4k+\ell)}(\log p)^{4k\ell/(4k+\ell)}$$

and the result follows. □

2.5. Polynomial Approximation of the Discrete Logarithm

In Chapters 3 and 4, we study the polynomial interpolation and we use the same techniques as in the polynomial approximation of the discrete logarithm problem. In this section, we recall some known lower bounds on the degree and weight of polynomial interpolating the discrete logarithm modulo a prime number p due to Coppersmith and Shparlinski (see [CS00] for details). The technique works as follows: from the initial polynomial, one constructs a non zero polynomial having a certain number of roots and whose lower bounds on its degree or weight (which can be obtained from some known lemmas) allow us to obtain lower bounds on the degree or weight of the initial polynomial . Let us fix a primitive root g modulo a prime number $p \geq 3$. For an integer x such that $\gcd(x, p) = 1$, we denote by $ind\, x$ its discrete logarithm, that is, the smallest non-negative integer u with $g^u = x \bmod p$. In public-key cryptography, the discrete logarithm problem is considered as a hard problem and to break this problem modulo p, it would be sufficient to have a univariate polynomial f polynomial over $\mathbb{F}_p$ of low degree which computes the discrete logarithm for almost all elements in $\{1, \ldots, p-1\}$, that is, $f(x) = ind\, x \bmod p$, for $x \in S \subseteq \{1, \ldots, p-1\}$ of the same size as p. It has been shown in [MD86] that the polynomial

$$f(x) = -1 + \sum_{k=1}^{p-2}(g^{-k} - 1)^{-1}x^k \bmod p$$

is the unique interpolation polynomial of the discrete logarithm modulo p for $S = \{1, \ldots, p-1\}$. Noting that any function over $\mathbb{F}_p$ can be approximated at $p-1$ points by a polynomial of

degree at most $p-2$, this polynomial is actually of the largest possible degree and is dense (namely it contains $p-1$ monomials). The two following results (see [CS00]) show that a low degree and low weight univariate polynomial cannot interpolate the discrete logarithm for sufficiently large sets S. These results do not guarantee the hardness of the discrete logarithm problem but they can be considered as a good indication that this problem is indeed a computationally hard problem.

Theorem 2.5.1. *Let $f(X) \in \mathbb{Z}[X]$ be a polynomial of degree $n = \deg(f) \leq p-2$ and of weight $t = w(f)$ such that*

$$ind\, x = f(x) \mod p, \quad x \in S,$$

for a set $S \subseteq \{1, \ldots, p-1\}$ of cardinality $|S| = p-1-s$. Then

$$\deg(f) \geq p-2-2s, \quad w(f) \geq (p-1)/(2s+1) - 1.$$

Proof. Let R be the set of $x \in \{1, \ldots, p-1\}$ for which both

$$ind\, x = f(x) \mod p \quad \text{and} \quad ind\, gx = f(gx) \mod p.$$

We have $|R| \geq p-1-2(p-1-|S|) = p-1-2s$. Indeed for $x \in \{1, \ldots, p-1\}$, $x \notin R$ if and only if ($x \notin S$ or $gx \notin S$). So there are at most $2s$ elements $x \in \{1, \ldots, p-1\}$ such that $x \notin R$ and thus $|R| \geq p-1-2s$. We have $ind\, gx = 1 + ind\, x$ if $x \neq g^{p-2} \mod p$. Hence

$$f(gx) = 1 + f(x) \bmod p,$$

for $x \in R$ with $x \neq g^{p-2} \mod p$. Therefore the polynomial $h(X) = f(gX) - f(X) - 1$ has at least $|R| - 1$ zeros modulo p and is not identical to zero modulo p (since $h(0) = -1$). Thus $n \geq \deg(h) \geq |R| - 1$. Also if f contains t monomials, then h contains at most $t+1$ monomials. Applying Lemma 2.2.1, we see that $p-1-(|R|-1) \geq (p-1)/(t+1)$ and the desired result follows. □

In particular, if $s = o(p)$, then $\deg(f), w(f) \sim p$. Theorem 2.5.1 is non trivial if the set S is dense enough ($|S| > p/2$). The next result is applicable to quite sparse sets S beginning with $|S| > (2p)^{1/2}$.

Theorem 2.5.2. *Let $f(X) \in \mathbb{Z}[X]$ be a polynomial of degree $n = \deg(f) \leq p-2$ such that*

$$ind\, x = f(x) \mod p, \quad x \in S,$$

for a set $S \subseteq \{1, \ldots, p-1\}$. Then

$$\deg(f) \geq \frac{|S|(|S|-1)}{2(p-2)}.$$

Proof. Let us consider the set

$$D = \{a = yx^{-1} \bmod p,\ 2 \leq a \leq p-1,\ x, y \in S\}.$$

Then $|D| \leq p-2$. There is $a \in D$ such that there are at least $\frac{|S|(|S|-1)}{|D|}$ representations $a = yx^{-1} \mod p$, $x, y \in S$; Select this a and let R be the set of $x \in \{1, \ldots, p-1\}$ for which both

$$ind\, x = f(x) \mod p \quad and \quad ind\, ax = f(ax) \mod p.$$

Thus $|R| \geq \frac{|S|(|S|-1)}{2(p-2)}$. Indeed for each representation $a = yx^{-1} \bmod p$, we get a pair x and $y = ax \bmod p$ of elements of S. We also have $ind\, ax = ind\, x + ind\, a$ or $ind\, ax = ind\, x + ind\, a - p + 1$. Hence either

$$f(ax) = ind\, ax = ind\, x + ind\, a = f(x) + ind\, a \bmod p$$

or

$$f(ax) = ind\, ax = ind\, x + ind\, a - p + 1 = f(x) + ind\, a + 1 \bmod p \bmod p$$

for $x \in R$. Therefore at least one of the polynomials $h_1(X) = f(aX) - f(X) - ind\, a$ and $h_2(X) = f(aX) - f(X) - ind\, a - 1$ has at least $|R|/2$ zeros modulo p. Because of our choice of D neither of these polynomials is identical to zero modulo p. Indeed,

$$h_1(0) = -ind\, a - 1 \neq 0 \bmod p$$

since $a \neq 1$, and

$$h_2(0) = -ind\, a \neq 0 \bmod p$$

since $0 \leq ind\, a \leq p - 2$. Thus $n \geq |R|/2$ and the desired result follows. □

The polynomial interpolation is a question which is well studied in general for cryptographic hard functions. For instance, lower bounds on the degree or weight of polynomials interpolating the discrete logarithm problem imply some lower bounds on the "sequential arithmetic complexity of the discrete logarithm in the computational tree model and in the random access machine model over real numbers" and lower bounds on the degree or weight of polynomials interpolating the Computational Diffie-Hellman assumption or the Decision Diffie-Hellman assumption have been obtained as well (see [MS01; Win01; KW04; Shp03] and references therein).

Chapter 3.

Polynomial Interpolation of the Naor-Reingold Pseudo-Random Functions

Many efficient public key cryptographic protocols are constructed using some assumptions which are believed to be hard, for instance the Computational Diffie-Hellman assumption or the Decision Diffie-Hellman assumption. In 1997, based on the Decision Diffie-Hellman assumption, Naor and Reingold [NR97; NR04] proposed an efficient pseudo-random function family. Since proving that the Decision Diffie-Hellman assumption holds seems currently to be out of reach, several number-theoretic properties and complexity measures have been studied for the Naor-Reingold pseudo-random functions over finite fields as well as over elliptic curves: distribution (see [LSW14; Shp00b] and references therein), linear complexity (see [CGS10; GGI11; Shp00a; SS01]) and non-linear complexity (see [BGLS00]). These results are incomparable but they all support the assumption of the pseudo-randomness of the Naor-Reingold function. In order to break the security of the Naor-Reingold function, it would be sufficient to have a k-variate polynomial f over a finite field (of low degree or low weight) with $k \geq 1$ which reveals information on the functions values that is a k-variate polynomial f satisfying: $(f(g^{\boldsymbol{a}^{x^1}}, \ldots, g^{\boldsymbol{a}^{x^k}}) = g^{\boldsymbol{a}^{x^{k+1}}}$, for all $\boldsymbol{a} = (a_1, \ldots, a_n) \in S$ for a large subset $S \subseteq (\mathbb{F}_\ell^*)^n$, and for some known values $x^1, \ldots, x^{k+1} \in \{0, \cdots, 2^n - 1\}$) or $(f(g^{\boldsymbol{a}^x}, g^{\boldsymbol{a}^{x+t_1}}, \ldots, g^{\boldsymbol{a}^{x+t_{k-1}}}) = g^{\boldsymbol{a}^{x+t_k}}$ for many integers $x \in \{0, 1, \ldots, 2^n - 1\}$, and for some known values $t_1, \ldots, t_k$ and for some known secret key $\boldsymbol{a}$), where for $x \in \{0, 1, \ldots, 2^n - 1\}$ $\boldsymbol{a}^x$ is defined in the next section. We refer the first case to *the polynomial interpolation with variable secret key* and the second case to *the polynomial interpolation with fixed secret key*. We prove that a low weight or degree k-variate polynomial cannot reveal information on the functions values. We consider the settings of a finite field and an elliptic curve and in both cases, we obtain lower bounds on the degree of polynomials interpolating the Naor-Reingold function with a fixed secret key and variable secret key. This Chapter is organized as follows: we first recall the Naor-Reingold pseudo-random function and some known results that we need in the rest of the Chapter. Then we study the polynomial interpolation of the Naor-Reingold function defined over a finite field with fixed secret key and variable secret key. We conclude this Chapter by the polynomial interpolation of the Naor-Reingold function defined over an elliptic curve with fixed secret key and variable secret key.

Contents

3.1. Naor-Reingold pseudo-random function 31
3.2. Auxiliary results 32
3.3. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Function over Finite Fields 32
3.3.1. Polynomial Interpolation with variable secret key 32
3.3.2. Polynomial Interpolation with fixed secret key 38
3.4. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Function over Elliptic Curves 42
3.4.1. Polynomial Interpolation with fixed secret key 42
3.4.2. Polynomial Interpolation with variable secret key 48

3.1. Naor-Reingold pseudo-random function

In cryptography, a pseudo-random function family is a collection of functions (that can be evaluated efficiently using a secret-key) with the property that an adversary cannot efficiently observe any significant difference between the input-output behavior of a random instance of the family or that of a random function.

More formally, we consider collections of functions $\{F_n : \mathcal{K}_n \times \mathcal{D}_n \to \mathcal{R}_n\}_{n\in\mathbb{N}}$ that can be evaluated by a (deterministic) polynomial-time Turing Machine. We define an adversary as a (non-uniform) probabilistic polynomial-time oracle Turing machine with either access to:

- an oracle implementing a function $F : \mathcal{D}_n \to \mathcal{R}_n$ defined by picking uniformly at random a secret-key $k \in \mathcal{K}_n$ such that $F(m) = F_n(k, m)$ for any $m \in \mathcal{D}_n$;
- or an oracle simulating a truly random function $F : \mathcal{D}_n \to \mathcal{R}_n$ (*i.e.* whose outputs are sampled uniformly and independently at random).

This adversary can decide which queries to make to the oracle, perhaps based on answers received to previous queries and eventually, it outputs a single bit (which is its decision as to which function the oracle is implementing). The *advantage* of the adversary is the function of n defined as the difference of the probabilities (taken over the random choices made by the adversary and the oracle) that the adversary outputs 1 in the two cases. A collection of functions $\{F_n : \mathcal{K}_n \times \mathcal{D}_n \to \mathcal{R}_n\}_{n\in\mathbb{N}}$ is a pseudo-random function family if and only if no adversary with advantage asymptotically larger than the inverse of a polynomial exists.

In 1997, Naor and Reingold [NR97; NR04] proposed a (candidate) pseudo-random function family which takes inputs in $\{0,1\}^n$ (for some parameter n) and outputs an element in some (multiplicatively written) group $\mathbb{G}$ of prime order ℓ with generator g. The secret key is an n-dimensional vector $\boldsymbol{a} = (a_1, \ldots, a_n) \in (\mathbb{Z}_\ell^*)^n$ and the Naor-Reingold function is defined as:

$$\begin{array}{rccl} f_{\boldsymbol{a}} & : & \{0,1\}^n & \longrightarrow & \mathbb{G} \\ & & (x_1, \ldots, x_n) & \longmapsto & f_{\boldsymbol{a}}(x_1, \ldots, x_n) = g^{\prod_{i=1}^n a_i^{x_i} \bmod \ell} \end{array}$$

The evaluation of $f_{\boldsymbol{a}}$ is thus efficient[1] since it consists only in n modular multiplications in $\mathbb{Z}_\ell$ and one modular exponentiation in $\mathbb{G}$. To lighten the notation, given an n-dimensional vector $\boldsymbol{a} = (a_1, \ldots, a_n) \in (\mathbb{Z}_\ell^*)^n$ and a variable x that will denote indifferently an n-bit string $(x_1, \ldots, x_n) \in \{0,1\}^n$ or an integer $x \in \{0, 1, \ldots, 2^n - 1\}$ (which implicitly defines $(x_1, \ldots, x_n) \in \{0,1\}^n$ the bit representation of x with extra leading zeros if necessary), we denote $\boldsymbol{a}^x$ the element in $\mathbb{Z}_\ell$ defined by $\boldsymbol{a}^x = a_1^{x_1} \cdots a_n^{x_n} \bmod \ell$. With this notation, the Naor-Reingold function is simply defined by $f_{\boldsymbol{a}}(x) = g^{\boldsymbol{a}^x}$. Two interesting candidates for $\mathbb{G}$ are a subgroup of the multiplicative group of a finite field and a subgroup of the points of an elliptic curve defined over a finite field. In the setting of an elliptic curve E defined over $\mathbb{F}_p$ for $p > 3$, we also define the function $\tilde{f}_{\boldsymbol{a}}(x) = [\boldsymbol{a}^x]P \in E \subset \mathbb{F}_p^2$, for a secret key $\boldsymbol{a} = (a_1, \ldots, a_n) \in (\mathbb{F}_\ell^*)^n$ where again x will denote indifferently an n-bit string $(x_1, \ldots, x_n) \in \{0,1\}^n$ or an integer $x \in \{0, 1, \ldots, 2^n - 1\}$. Because of the algebraic structure of E, this function is not pseudo-random and the Naor-Reingold pseudo-random function over $E(\mathbb{F}_p)$ is thus defined as, $f_{\boldsymbol{a}}(x) = X(\tilde{f}_{\boldsymbol{a}}(x))$, where $X(P)$ denotes the abscissa of $P \in E$.

[1] More efficient candidates of pseudo-random function families are known, but the Naor-Reingold function family is among the most efficient ones with strong security guarantees under a standard computational assumption.

3.2. Auxiliary results

In the forthcoming sections, we also need the following lemmas from [WS99] and [GGI11] about the distribution of products $\mathbf{a}^x$ in $\mathbb{F}_\ell^*$ to obtain concrete lower bounds on the weight and degree of polynomials interpolating the Naor-Reingold pseudo-random function with fixed secret key.

Lemma 3.2.1 ([WS99])**.** *Let $m \geq 1$ be an integer. For any $\Delta > 0$ and for all but at most $2^{-m}\Delta^{-1}(\ell-1)^{m+2}$ vectors $\mathbf{a} = (a_1, \ldots, a_m) \in (\mathbb{F}_\ell^*)^m$, the products $\mathbf{a}^x$ for $x \in \{0,1\}^m$ take at least $\ell - 1 - \Delta$ values in $\mathbb{F}_\ell^*$.*

Lemma 3.2.2 ([GGI11])**.** *Let $n \geq j > 0$ be two integers. For all but at most $(3^j - 1)(\ell-1)^{n-1}/2$ vectors $\mathbf{a} = (a_1, \ldots, a_n) \in (\mathbb{F}_\ell^*)^n$ the products $\mathbf{a}^x$ for $x \in \{0,1\}^n$ take at least 2^j values in $\mathbb{F}_\ell^*$.*

3.3. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Function over Finite Fields

In this section, we study the polynomial representation of the Naor-Reingold pseudo-random function over finite fields. From the known lower bounds on the polynomial interpolation on the discrete logarithm and the Diffie-Hellman Problem in the groups we considered (*e.g.* [CS00; KW06; LW02; LW03a; MW08; KW04; Shp03] and references therein) and the known result on the non-linear complexity of the Naor-Reingold pseudo-random function (see [BGLS00]), we prove that a low weight or degree k-variate polynomial cannot reveal information on the functions values.

3.3.1. Polynomial Interpolation with variable secret key

In this section, q is a prime power, n is an integer and $g \in \mathbb{F}_q^*$ is an element of prime order ℓ (with $\ell \mid q-1$). We prove results on the multivariate polynomial approximation of the Generalized Diffie-Hellman and the Naor-Reingold functions over a finite field. We consider polynomials that approximate values of these functions for fixed values in $\{0, \cdots, 2^n - 1\}$ and a large set of keys. First, we consider an approximation by a polynomial with k variables, with $k \leq n$.

Theorem 3.3.1. *Let $1 \leq k \leq n$ be an integer. Let $S \subseteq (\mathbb{F}_\ell^*)^n$, with $|S| = (\ell-1)^n - s$ with $|S| > k(\ell-1)^{n-1}$. Let $x^1, \ldots, x^{k+1} \in \{1, \cdots, 2^n - 1\}$ be pairwise distinct and let $f \in \mathbb{F}_q[X_1, \ldots, X_k]$, be a polynomial satisfying:*

$$f\left(g^{\boldsymbol{a}^{x^1}}, \ldots, g^{\boldsymbol{a}^{x^k}}\right) = g^{\boldsymbol{a}^{x^{k+1}}}, \quad \text{for all } \boldsymbol{a} = (a_1, \ldots, a_n) \in S. \tag{3.1}$$

If the elements x^i, $i \in \{1, \ldots, k\}$ seen as vectors over $\mathbb{F}_2^n$ are linearly independent over $\mathbb{F}_2$, then:

$$\deg(f) \geq \frac{\ell-1}{2} - \frac{s}{(\ell-1)^{n-1}}$$

and if $\deg_{X_i}(f) \leq \frac{\ell-1}{2}$, for all $i \in \{1, \ldots, k\}$, we have

$$w(f) \geq \frac{\ell^{k/2}}{2^{1/2}(\ell^k - (\ell-1)^k + 2s/(\ell-1)^{n-k})^{1/2}}.$$

In particular, for $s = o(\ell^n)$, we have $\deg(f) = \Omega(\ell)$ and if $\deg_{X_i}(f) \leq \frac{\ell-1}{2}$, for all $i \in \{1, \ldots, k\}$, we have $w(f) = \Omega(\ell^{k/2})$.

Proof. For any $i \in \{1, \ldots, k+1\}$, we denote $x^i = x_1^i \ldots x_n^i$ its binary representation and put $\tilde{x^i} = x_1^i \ldots x_k^i 0 \ldots 0$ which is obtained from x^i by considering the k first positions of its binary representation and replacing the $n-k$ last positions by 0. We suppose without loss of generality that the elements $\tilde{x^i}$, $i \in \{1, \ldots, k\}$ seen as vectors over $\mathbb{F}_2^n$ are linearly independent over $\mathbb{F}_2$.

Since $|S| > k(\ell-1)^{n-1}$, we have $w(f) \geq 2$ by the following claim:

Claim 3.3.2. *If $w(f) = 1$ then (3.1) holds for at most $k(\ell-1)^{n-1}$ keys $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.*

Proof. If $w(f) = 1$, then f is a monomial and there exists $(\alpha_1, \ldots, \alpha_k) \in \{0, \ldots, \ell-1\}^k$ such that $\alpha_1 \boldsymbol{a}^{x^1} + \cdots + \alpha_k \boldsymbol{a}^{x^k} = \boldsymbol{a}^{x^{k+1}} \bmod \ell$ (where f is the monomial $f(X_1, \ldots, X_k) = X_1^{\alpha_1} \ldots X_k^{\alpha_k}$). We prove by induction on k that the number of $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ such that $\alpha_1 \boldsymbol{a}^{x^1} + \cdots + \alpha_k \boldsymbol{a}^{x^k} = \boldsymbol{a}^{x^{k+1}}$ does not exceed $k(\ell-1)^{n-1}$.

1. For $k = 0$, the equation $\boldsymbol{a}^{x^{k+1}} = 0$ has no solution and the statement is clearly true.
2. Otherwise, because $x^{k+1} \neq x^k$, there exists j such that the j-th component of x^{k+1} is different from the j-th component of x^k. Then the above equation can be written in the form $A = Ba_j$ where A and B do not depend on a_j. If $B \neq 0$, then for any vector $(a_1, \cdots, a_{j-1}, a_{j+1}, \cdots, a_n) \in (\mathbb{F}_\ell^*)^{n-1}$, the value of a_j is defined uniquely. If $B = 0$, then $A = 0$ and by induction, the number of $(a_1, \cdots, a_{j-1}, a_{j+1}, \cdots, a_n) \in (\mathbb{F}_\ell^*)^{n-1}$ does not exceed $(k-1)(\ell-1)^{n-2}$. Therefore, the number of solutions does not exceed $(k-1)(\ell-1)^{n-1} + (\ell-1)^{n-1} = k(\ell-1)^{n-1}$, and the result follows.

□

There is some $t \in \{1, \ldots, n\}$ such that $x_t^{k+1} = 1$ and let $T = \{i : x_t^i = 1\} \subseteq \{1, \ldots, k\}$ that we denote $T = \{i_1, \ldots, i_v\}$, with $i_j < i_{j+1}$ for $j \in \{1, \ldots, v-1\}$. Let

$$W = \left\{ \boldsymbol{a} \in (\mathbb{F}_\ell^*)^n : \begin{array}{l} \boldsymbol{a} = (a_1, \ldots, a_n) \in S \\ \text{and } (a_1, \ldots, a_{t-1}, 2a_t, a_{t+1}, \ldots, a_n) \in S \end{array} \right\},$$

then by the union bound, $|W| \geq (\ell-1)^n - 2s$. By the pigeonhole principle, there exists $(b_{k+1}, \ldots, b_n) \in (\mathbb{F}_\ell^*)^{n-k}$ such that the set

$$T = \{(a_1, \ldots, a_k) \in (\mathbb{F}_\ell^*)^k : a' = (a_1, \ldots, a_k, b_{k+1}, \ldots, b_n) \in W\}$$

satisfies $|T| \geq (\ell-1)^k - 2s/(\ell-1)^{n-k}$. For all $\boldsymbol{a_0} = (a_1, \ldots, a_k) \in T$, putting $\boldsymbol{a'} = (a_1, \ldots, a_k, b_{k+1}, \ldots, b_n)$, we have:

$$\begin{cases} f\left(g^{\boldsymbol{a'}^{x^1}}, \ldots, g^{\boldsymbol{a'}^{x^k}}\right) = g^{\boldsymbol{a'}^{x^{k+1}}} \\ f\left(g^{\boldsymbol{a'}^{x^1}}, \ldots, g^{\boldsymbol{a'}^{x^{i_1}-1}}, g^{2\boldsymbol{a'}^{x^{i_1}}}, g^{\boldsymbol{a'}^{x^{i_1+1}}}, \ldots, g^{\boldsymbol{a}^{x^{i_v}-1}}, g^{2\boldsymbol{a}^{x^{i_v}}}, g^{\boldsymbol{a}^{x^{i_v+1}}}, \ldots, g^{\boldsymbol{a}^{x^n}}\right) = g^{2\boldsymbol{a'}^{x^{k+1}}} \end{cases}$$

Since the elements $\tilde{x^i}$, $i \in \{1, \ldots, k\}$ seen as vectors over $\mathbb{F}_2^n$ are linearly independent over $\mathbb{F}_2$, one can verify that the set

$$\left\{ \left(g^{\boldsymbol{a'}^{x^1}}, \ldots, g^{\boldsymbol{a'}^{x^k}}\right) \in (\mathbb{F}_q^*)^k : \boldsymbol{a'} = (\boldsymbol{a_0}, b_{k+1}, \ldots, b_n) \in W, \text{ and } \boldsymbol{a_0} \in T \right\}$$

is of the same cardinality as T. Hence the polynomial

$$F(X_1,\ldots,X_k) = f(X_1,\ldots,X_{i_1}^2,\ldots,\ldots,X_{i_v}^2,\ldots,X_k) - f^2(X_1,\ldots,X_k)$$

has at least $|T| \geq (\ell-1)^k - 2s/(\ell-1)^{n-k}$ zeros. Since $w(f) \geq 2$, one can see that F is a nonzero polynomial and $\deg F \leq 2\deg f$. By Lemma 2.2.2, we obtain:

$$\deg(f) \geq \frac{\ell-1}{2} - \frac{s}{(\ell-1)^{n-1}}.$$

Furthermore if $\deg_{X_i}(f) \leq \frac{\ell-1}{2}$, for all $i \in \{1,\ldots,n\}$, and since $w(F) \leq 2w(f)^2$, then by applying Lemma 2.2.3 we have:

$$w(f) \geq \frac{\ell^{k/2}}{2^{1/2}(\ell^k - (\ell-1)^k + 2s/(\ell-1)^{n-k})^{1/2}}.$$

□

Now we consider an approximation by a polynomial with k variables and $k > n$ with some technical conditions on the input values $x^i \in \{0,\cdots,2^n-1\}$ for $i \in \{1,\ldots,k\}$.

Theorem 3.3.3. *Let $k > n$ be some integer. Let $S \subseteq (\mathbb{F}_\ell^*)^n$, with $|S| = (\ell-1)^n - s$. Let $x^1,\ldots,x^{k+1} \in \{0,\cdots,2^n-1\}$ be pairwise distinct such that $x^1 = 2^{n-1} = (1,0,\ldots,0)$, $x_1^{k+1} = 1$ and $x_1^i = 0$ for $i \in \{2,\ldots,k\}$ and let $f \in \mathbb{F}_q[X_1,\ldots,X_k]$, with $k > n$ be a polynomial satisfying:*

$$f\left(g^{\boldsymbol{a}^{x^1}},\ldots,g^{\boldsymbol{a}^{x^k}}\right) = g^{\boldsymbol{a}^{x^{k+1}}}, \quad \text{for all } \boldsymbol{a} = (a_1,\ldots,a_n) \in S.$$

We have

$$\deg(f) \geq \frac{\ell-1}{2} - \frac{s}{(\ell-1)^{n-1}}.$$

Proof. Let W be the set of vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ such that $\boldsymbol{a} = (a_1,\ldots,a_n) \in S$, $(a_1+1,\ldots,a_n) \in S$ and $\boldsymbol{a'} = (1,a_2,\ldots,a_n)$ satisfies $\boldsymbol{a'}^{x^{k+1}} \neq \alpha \mod \ell$ for all $\alpha \in \{1,\ldots,d\}$, where d denotes the degree of f.

Claim 3.3.4. *We have*

$$|W| \geq (\ell-1)^n - 2s - \deg(f)(\ell-1)^{n-1}.$$

Proof. Let $\alpha \in \{1,\ldots,d\}$, the number of $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ such that $\boldsymbol{a'}^{x^{k+1}} = \alpha \mod \ell$ does not exceed $(\ell-1)^{n-1}$.

Indeed, since $x^k \neq x^1$, there exists $j \in \{2,\ldots,n\}$ such that $x_j^{k+1} = 1$, then for any vector $(a_1,\ldots,a_{j-1},a_{j+1},\cdots,a_n) \in (\mathbb{F}_\ell^*)^{n-1}$, the value of a_j is defined uniquely by this equation. Since the number of vectors $\boldsymbol{a} = (a_1,\ldots,a_n) \in S$ such that $(a_1+1,\ldots,a_n) \notin S$ does not exceed s, the result follows. □

By the pigeonhole principle, there exists $\boldsymbol{b} = (1,b_2,\ldots,b_n) \in (\mathbb{F}_\ell^*)^n$ such that the set

$$T = \{a_1 \in \mathbb{F}_\ell : \boldsymbol{a} = (a_1,b_2,\ldots,b_n) \in W\}$$

satisfies $|T| \geq \ell - 1 - \deg(f) - \frac{2s}{(\ell-1)^{n-1}}$. Then for all $a_1 \in T$, putting $\boldsymbol{a} = (a_1,b_2,\ldots,b_n)$, we have:

$$\begin{cases} f\left(g^{a_1}, g^{\boldsymbol{b}^{x^2}} \ldots, g^{\boldsymbol{b}^{x^k}}\right) = g^{\boldsymbol{a}^{x^{k+1}}} \\ f\left(g^{a_1+1}, g^{\boldsymbol{b}^{x^2}}, \ldots, g^{\boldsymbol{b}^{x^k}}\right) = g^{(a_1+1,b_2,\ldots,b_n)^{x^{k+1}}} = g^{\boldsymbol{b}^{x^{k+1}}}.g^{\boldsymbol{a}^{x^{k+1}}} \end{cases}$$

We have for all $a_1 \in T$

$$f\left(g.g^{a_1}, g^{\boldsymbol{b}^{x^2}}, \ldots, g^{\boldsymbol{b}^{x^k}}\right) - g^{\boldsymbol{b}^{x^{k+1}}} f\left(g^{a_1}, g^{\boldsymbol{b}^{x^2}} \ldots, g^{\boldsymbol{b}^{x^k}}\right) = 0 \tag{3.2}$$

and the polynomial

$$F(X) = f\left(gX, g^{\boldsymbol{b}^{x^2}}, \ldots, g^{\boldsymbol{b}^{x^k}}\right) - g^{\boldsymbol{b}^{x^{k+1}}} f\left(X, g^{\boldsymbol{b}^{x^2}} \ldots, g^{\boldsymbol{b}^{x^k}}\right)$$

has at least $\ell - 1 - \deg(f) - \frac{2s}{(\ell-1)^{n-1}}$ zeros.

The polynomial $f(X, g^{\boldsymbol{b}^{x^2}}, \ldots, g^{\boldsymbol{b}^{x^k}})$ is a nonzero polynomial by the first equation of the previous system and has degree smaller than $\deg(f)$. Let d_0 its degree, then $\boldsymbol{b}^{x^{k+1}} \neq d_0 \mod \ell$ by construction of W and it follows that the leading monomial of F is nonzero which implies that the polynomial F is nonzero. We also have $\deg(F) \leq \deg(f)$ and hence, by Lemma 2.2.2, we obtain:

$$\deg(f) \geq \ell - 1 - \deg(f) - \frac{2s}{(\ell-1)^{n-1}},$$

and the result follows. □

Theorem 3.3.3 can be applied to give lower bounds on the degree of interpolating polynomials for several generalized Diffie-Hellman problems (with $k > n$ variables) from [BCP07].

Since the weight of a polynomial is a more discerning complexity estimate, we now prove a lower bound on the weight of an approximation by a polynomial with k variables and $k > n$ (and without any condition on the input values $x^i \in \{0, \cdots, 2^n - 1\}$ for $i \in \{1, \ldots, k\}$).

Theorem 3.3.5. *Let $k > n$ be some integer. Let $S \subseteq (\mathbb{F}_\ell^*)^n$, with $|S| = (\ell-1)^n - s$. Let $x^1, \ldots, x^{k+1} \in \{1, \cdots, 2^n - 1\}$ be pairwise distinct and let $f \in \mathbb{F}_q[X_1, \ldots, X_k]$ be a polynomial satisfying:*

$$f\left(g^{\boldsymbol{a}^{x^1}}, \ldots, g^{\boldsymbol{a}^{x^k}}\right) = g^{\boldsymbol{a}^{x^{k+1}}}, \quad \text{for all } \boldsymbol{a} = (a_1, \ldots, a_n) \in S,$$

for some different values $x^1, \ldots, x^{k+1} \in \{1, \cdots, 2^n - 1\}$. Then

$$w(f) \geq \left(\frac{\ell - 3 - \frac{s}{(\ell-1)^{n-1}}}{2 + 2k + \frac{s}{(\ell-1)^{n-1}}}\right)^{1/2}.$$

Proof. Let $I = \{i \in \{1, \ldots, k\} : x_n^i = 1\}$ that we denote $I = \{i_1, \ldots, i_v\}$ with $i_1 < i_2 < \cdots < i_v$. Let $A = \{\alpha_i = (\alpha_i^1, \ldots, \alpha_i^v) \in \{0, \ldots, \deg(f)\}^v\}$ be a set of cardinality at most $w(f)$ which will be given explicitly later in the proof and W_A be the set of vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ such that:

1. $\boldsymbol{a} = (a_1, \ldots, a_n) \in S$

2. $\boldsymbol{a}$ satisfies $\alpha_i^1 \boldsymbol{a}^{x^{i_1}-1} + \cdots + \alpha_i^v \boldsymbol{a}^{x^{i_v}-1} \neq \boldsymbol{a}^{x^{k+1}-1}$ for all $\alpha_i \in A$

Claim 3.3.6. *We have* $|W_A| \geq (\ell-1)^n - T_0$, *where* $T_0 = s + w(f)k(\ell-1)^{n-1}$.

Proof. For a fixed tuple $\alpha_i \in A$, by proceeding exactly as in the proof of Claim 1 one can prove by induction in v that the number of $a \in (\mathbb{F}_\ell^*)^n$ such that $\alpha_i^1 \boldsymbol{a}^{x^{i_1}-1} + \cdots + \alpha_i^v \boldsymbol{a}^{x^{i_v}-1} \neq \boldsymbol{a}^{x^{k+1}-1}$ does not exceed $v(\ell-1)^{n-1}$. Since the cardinality of A is at most $w(f)$ and $v \leq k$, we thus have $|W_A| \geq |S| - kw(f)(\ell-1)^{n-1}$. □

There exists by the pigeonhole principle $\boldsymbol{b} = (b_1, \ldots, b_{n-1}, 1) \in (\mathbb{F}_\ell^*)^n$ such that

$$T = \{a_n \in \mathbb{F}_\ell \,:\, \boldsymbol{a} = (b_1, \ldots, b_{n-1}, a_n) \in W_A\}$$

satisfies $|T| \geq \ell - 1 - \frac{T_0}{(\ell-1)^{n-1}}$. Then for all $a_n \in T$, we have:

$$f\left(g^{\boldsymbol{b}^{x^1}}, \ldots, g^{\boldsymbol{b}^{x^{i_1}-1}}, g^{\boldsymbol{b}^{x^{i_1}} a_n}, g^{\boldsymbol{b}^{x^{i_1+1}}}, \ldots \right.$$
$$\left. \ldots, g^{\boldsymbol{b}^{x^{i_v}-1}}, g^{\boldsymbol{b}^{x^{i_v}-1} a_n}, g^{\boldsymbol{b}^{x^{i_v+1}}}, \ldots, g^{\boldsymbol{b}^{x^k}}\right) = g^{\boldsymbol{b}^{x^{k+1}-1} a_n}.$$

Let

$$H(X) = f\left(g^{\boldsymbol{b}^{x^1}}, \ldots, g^{\boldsymbol{b}^{x^{i_1}-1}}, X^{\boldsymbol{b}^{x^{i_1}-1}}, g^{\boldsymbol{b}^{x^{i_1+1}}}, \ldots \right.$$
$$\left. \ldots, g^{\boldsymbol{b}^{x^{i_v}-1}}, X^{\boldsymbol{b}^{x^{i_v}-1}}, g^{\boldsymbol{b}^{x^{i_v+1}}}, \ldots, g^{\boldsymbol{b}^{x^k}}\right) - X^{\boldsymbol{b}^{x^{k+1}-1}}$$

and $K(X)$ the polynomial obtained from $H(X)$ by reducing the exponents of every monomial modulo ℓ. If we choose A to be the set of vectors obtained from the multivariate polynomial f by considering the monomials with variables $X_{i_1}, \ldots, X_{i_v}$ from each monomial of f, then A is of cardinality at most $w(f)$ and does not depend on $\boldsymbol{b}$. One can see that $K(X)$ is a nonzero polynomial by the choice of $\boldsymbol{b}$ and has degree less than ℓ with at least $|T|$ zeros. Hence by Lemma 2.2.3, we obtain:

$$w(f) + 1 \geq w(K) \geq \frac{\ell}{1 + \frac{T_0}{(\ell-1)^{n-1}}},$$

and $(w(f)+1)\left(2(\ell-1)^{n-1} + s + w(f)k(\ell-1)^{n-1}\right) \geq (\ell-1)^n$. We thus have:

$$w(f)^2\left(2(\ell-1)^{n-1} + s + 2k(\ell-1)^{n-1}\right) \geq (\ell-1)^n - 2(\ell-1)^{n-1} - s,$$

and the result follows.

□

Theorem 3.3.5 gives a lower bound on the weight of explicit polynomials approximating the Naor-Reingold pseudo-random function and it immediately gives a lower bound on the weight of explicit polynomials approximating the n-partite Diffie-Hellman problem by some well chosen inputs:

Corollary 3.3.7. *Let* $S \subseteq (\mathbb{F}_\ell^*)^n$, *with* $|S| = (\ell-1)^n - s$.
Let $f \in \mathbb{F}_q[X_1, \ldots, X_n]$ *be a polynomial satisfying* $f(g^{a_1}, \ldots, g^{a_n}) = g^{a_1 \cdots a_n}$ *for all* $\boldsymbol{a} = (a_1, \ldots, a_n) \in S$. *We have*

$$w(f) \geq \left(\frac{\ell - 3 - \frac{s}{(\ell-1)^{n-1}}}{2 + 2n + \frac{s}{(\ell-1)^{n-1}}}\right)^{1/2}.$$

The next theorem extends the previous approach and gives a lower bound on the weight of implicit polynomials approximating the generalized Diffie-Hellman problem.

Theorem 3.3.8. *Let $S \subseteq (\mathbb{F}_\ell^*)^n$, with $|S| = (\ell-1)^n - s$.*
Let $f \in \mathbb{F}_q[X_1, \ldots, X_{n+1}]$ be a polynomial satisfying:

$$f(g^{a_1}, \ldots, g^{a_n}, g^{a_1 \ldots a_n}) = 0, \quad \textit{for all } \boldsymbol{a} = (a_1, \ldots, a_n) \in S,$$

then

$$w(f) \geq \left(\frac{\ell(\ell-1)^{n-1}}{2(\ell-1)^{n-1} + s} \right)^{1/2}.$$

Proof. Let $(\alpha, \beta) \in \{0, \ldots, \deg(f)\}^2\}$ with $(\alpha, \beta) \neq (0, 0)$.
Let $A = \{(\alpha', \beta') \in \{0, \ldots, \deg(f)\}^2\}$ be a set of cardinality at most $w(f)$ with $(\alpha, \beta) \notin A$ and let W_A be the set of vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ such that:

1. $\boldsymbol{a} = (a_1, \ldots, a_n) \in S$
2. $\boldsymbol{a}$ satisfies $\alpha + \beta(a_2 \ldots a_n) \neq \alpha' + \beta'(a_2 \ldots a_n) \mod \ell$ for all $(\alpha', \beta') \in A$

Claim 3.3.9. *We have $|W_A| \geq (\ell-1)^n - s - w(f)(\ell-1)^{n-1}$.*

Proof. Given $(\alpha', \beta') \in A$, the number of $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ such that

$$\alpha + \beta(a_2 \ldots a_n) = \alpha' + \beta'(a_2 \ldots a_n) \bmod \ell$$

does not exceed $(\ell-1)^{n-1}$. Indeed, we have

$$\alpha - \alpha' + (\beta - \beta')(a_2 \ldots a_n) = 0 \mod \ell,$$

and we can easily see that $\beta - \beta' \neq 0 \mod \ell$ (since otherwise, we have $\alpha - \alpha' = 0 \mod \ell$). Therefore, for any vector $(a_1, a_3, \cdots, a_n) \in (\mathbb{F}_\ell^*)^{n-1}$, the value of a_2 is defined uniquely.

Since the total number of couples (α', β') does not exceed $w(f)$, the number of $\boldsymbol{a} \in S$ such that $\boldsymbol{a} \notin W_A$ does not exceed $w(f)(\ell-1)^{n-1}$. □

There exists by the pigeonhole principle $\boldsymbol{b} = (b_2, \ldots, b_n) \in (\mathbb{F}_\ell^*)^{n-1}$ such that $T = \{a_1 \in \mathbb{F}_\ell : \boldsymbol{a} = (a_1, \boldsymbol{b}) \in W_A\}$ satisfies $|T| \geq \ell - 1 - w(f) - \frac{s}{(\ell-1)^{n-1}}$. Then for all $a_1 \in T$, we have:

$$f(g^{a_1}, g^{b_2}, \ldots g^{b_n}, g^{a_1 b_2 \ldots b_n}) = 0.$$

Let $H(X) = f(X, g^{b_2}, \ldots, g^{b_n}, X^{b_2 \ldots b_n})$ and $K(X)$ the polynomial obtained from $H(X)$ by reducing the exponents of every monomial modulo ℓ. If we choose A independent of $\boldsymbol{b}$ and of cardinality at most $w(f)$, as in the proof of Theorem 3.3.5 (but this time with variables X_1 and X_{n+1}) , then $K(X)$ is not a zero polynomial by the choice of $\boldsymbol{b}$ and has degree less than ℓ with at least $|T|$ zeros. Hence by Lemma 2.2.3, we obtain:

$$w(f) \geq w(K) \geq \frac{\ell}{1 + w(f) + \frac{s}{(\ell-1)^{n-1}}},$$

and the result follows. □

3.3.2. Polynomial Interpolation with fixed secret key

In this section, p is an odd prime number, n is an integer and $g \in \mathbb{F}_p^*$ is an element of prime order ℓ (with $\ell \mid p-1$). We prove results on the univariate and multivariate polynomial interpolation of the Naor-Reingold pseudo-random function over finite fields. We consider polynomials that interpolates values of the Naor-Reingold pseudo-random function for a fixed secret key $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$. The values considered are evaluation of the function at integers $x \in \{0, \cdots, 2^n - 1\}$ and translates of these values by some fixed constants $t_1, t_2, \cdots, t_k \in \mathbb{N}$. This setting is interesting for applications in cryptography. Note that if one value $x + t_i$ is larger than 2^n for some $i \in \{1, \ldots, k\}$ then, the Naor-Reingold function is not defined at $x+t_i$. In the following, we consider simple sets where all translates belong to the Naor-Reingold function domain but our method can be adapted to other settings.

First, we consider multivariate polynomial interpolation over large sets of values.

Theorem 3.3.10. *Let $t \geq 1$ be an integer. Let $t_1, t_2, \cdots, t_k$ be fixed distinct integers such that $t_1, t_2, \cdots, t_k < 2^t$ and let $A \subseteq \{0, \cdots, 2^n - 1\}$. For some $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, let $F_{\boldsymbol{a}}(X_1, \ldots, X_k) \in \mathbb{F}_p[X_1, \cdots, X_k]$ such that*

$$F_{\boldsymbol{a}}(f_{\boldsymbol{a}}(x), f_{\boldsymbol{a}}(x + t_1), \cdots, f_{\boldsymbol{a}}(x + t_{k-1})) = f_{\boldsymbol{a}}(x + t_k) \tag{3.3}$$

for all $x \in A$. For all but at most $2k(\ell-1)^{n-1}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, we have

$$\begin{cases} \binom{\deg(F_{\boldsymbol{a}}) + k}{k} \geq \dfrac{\ell}{2\Delta + 1} - 1 \\ w(F_{\boldsymbol{a}}) \geq \dfrac{\ell}{2\Delta + 1} - 1 \end{cases}$$

where $\Delta = \ell - 1 - \sharp S$ for the set $S = \{\boldsymbol{a}^{2^t x} \in \mathbb{F}_\ell^* : 2^t x \in A\}$,

It is worth noting that the conclusion of Theorem 3.3.10 cannot hold for all vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$. For instance, if we consider a secret key $\boldsymbol{a} = (a_1, \ldots, a_n) \in (\mathbb{F}_\ell^*)^n$ such that $a_{n-1} = a_n$ and the simple case $k = 1$ and $t_1 = 1$, we have $f_{\boldsymbol{a}}(x + t_1) = f_{\boldsymbol{a}}(x)$ for all integer x in the set $A = \{x \in \{0, \cdots, 2^n - 1\}, x \equiv 1 \bmod 4\}$, (since $x = (x_1, x_2, \ldots, x_{n-2}, 0, 1)$ and $x + t_1 = (x_1, x_2, \ldots, x_{n-2}, 1, 0)$). The polynomial $F_{\boldsymbol{a}}(X_1) = X_1$ of degree 1 and weight 1 therefore satisfies (3.3) for all $x \in A$ where the set A is very large since $\sharp A = 2^n/4$. However, Theorem 3.3.10 ensures that the lower bounds on the degree and the weight of F hold with probability $1 - 2k/(\ell - 1)$ when the secret key $\boldsymbol{a}$ is picked uniformly at random (and hence with overwhelming probability for k polynomial in the security parameter).

In Theorem 3.3.10 statement, it is also necessary to consider the cardinality of a subset of $\{\boldsymbol{a}^x \in \mathbb{F}_\ell^*, x \in A\}$ and not the cardinality of A itself since it is possible that for some secret key $\boldsymbol{a}$, the latter is "large" while the former is "small". For instance, for a secret key $\boldsymbol{a} = (a, \ldots, a) \in (\mathbb{F}_\ell^*)^n$ (where all components are equal to some constant value $a \in \mathbb{F}_\ell^*$), we have $f_{\boldsymbol{a}}(x) = g^{\boldsymbol{a}^x} = g^{\boldsymbol{a}^{\mathrm{hw}(x)}}$ where $\mathrm{hw}(x)$ denotes x's Hamming weight (i.e., its number of non-zero coordinates). In this case, even if the set A is very large, $\{\boldsymbol{a}^x \in \mathbb{F}_\ell^*, x \in A\}$ is of cardinality at most n and one can construct a small degree multivariate polynomial that interpolates the values of the Naor-Reingold pseudo-random function.

Proof. Since $t_1, t_2, \cdots, t_k < 2^t$, we have

$$\boldsymbol{a}^{2^t x + t_i} = \boldsymbol{a}^{2^t x} \boldsymbol{a}^{t_i}$$

for all $x \in A$ such that $2^t x \leq 2^n - 1$ and $i \in \{1, \cdots, k\}$. The relation (3.3) thus becomes

$$F_{\boldsymbol{a}}(g^u, g^{u\boldsymbol{a}^{t_1}}, \cdots, g^{u\boldsymbol{a}^{t_{k-1}}}) = g^{u\boldsymbol{a}^{t_k}},$$

for all $u \in S$. Let $R = \{u \in S | u(\boldsymbol{a}^{t_k})^{-1} \in S\}$. We put $\Delta = \ell - 1 - \sharp S$ and, by the union bound, we have $\sharp R \geq \ell - 1 - 2\Delta$ and

Claim 3.3.11.

$$F_{\boldsymbol{a}}(g^{u(\boldsymbol{a}^{t_k})^{-1}}, \cdots, g^{u\boldsymbol{a}^{t_{k-1}}(\boldsymbol{a}^{t_k})^{-1}}) = g^u,$$

for all $u \in R$.

Let $H_{\boldsymbol{a}}(X) = F_{\boldsymbol{a}}(X^{(\boldsymbol{a}^{t_k})^{-1}}, \cdots, X^{\boldsymbol{a}^{t_{k-1}}(\boldsymbol{a}^{t_k})^{-1}}) - X \in \mathbb{F}_p[X]$ and $K_{\boldsymbol{a}}(X)$ the polynomial obtained from $H_{\boldsymbol{a}}(X)$ by considering the degree of monomials of $H_{\boldsymbol{a}}(X)$ modulo ℓ.

Claim. The polynomial $K_{\boldsymbol{a}}(X)$ is not a zero polynomial for all but at most $2k(\ell-1)^{n-1}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

Proof. Indeed if $K_{\boldsymbol{a}}(X)$ is a zero polynomial, then

- either $F_{\boldsymbol{a}}$ is a monomial of the form $X_1^{\alpha_1} \cdots X_k^{\alpha_k}$, with $(\alpha_1, \cdots, \alpha_k) \neq (0, \cdots, 0)$
- or $F_{\boldsymbol{a}}$ would be a sum of at least two monomials $X_1^{\alpha_1} \cdots X_k^{\alpha_k}$ and $X_1^{\beta_1} \cdots X_k^{\beta_k}$ and there would exist $(\alpha_1, \cdots, \alpha_k) \neq (\beta_1, \cdots, \beta_k)$ such that

  $$\alpha_1(\boldsymbol{a}^{t_k})^{-1} + \cdots + \alpha_k \boldsymbol{a}^{t_{k-1}}(\boldsymbol{a}^{t_k})^{-1} = \beta_1(\boldsymbol{a}^{t_k})^{-1} + \cdots + \beta_k \boldsymbol{a}^{t_{k-1}}(\boldsymbol{a}^{t_k})^{-1}$$

  in $\mathbb{F}_\ell$.

If $F_{\boldsymbol{a}}$ is of the form $X_1^{\alpha_1} \cdots X_k^{\alpha_k}$, then from (3.3), it will follows that

$$\alpha_1 \boldsymbol{a}^x + \cdots + \alpha_k \boldsymbol{a}^{x+t_{k-1}} = \boldsymbol{a}^{x+t_k} \quad \text{in } \mathbb{F}_\ell, \text{ for all } x \in A. \tag{3.4}$$

Let x such that (3.4) is satisfied. Then we can easily prove for all $n \geq 1$ by induction in k that the number of $a \in (\mathbb{F}_\ell^*)^n$ solutions of (3.4) does not exceed $k(\ell-1)^{n-1}$.

1. For $k = 0$, the equation $\boldsymbol{a}^{x+t_k} = 0$ has no solution and the statement is clearly true.
2. Otherwise, let $j = \max(\{i \in \{1 \ldots, k\} | \alpha_i \neq 0\})$. Because $x + t_k \neq x + t_j$, there exists i such that i-th component of $x + t_k$ is different from the i-th component of $x + t_j$. Then the above equation can be written in the form $T_1 = T_2 a_i$ where T_1 and T_2 do not depend on a_i. If $T_2 \neq 0$, then for any vector $(a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n) \in (\mathbb{F}_\ell^*)^{n-1}$, the value of a_i is defined uniquely. If $T_2 = 0$, then by induction, the number of $(a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n) \in (\mathbb{F}_\ell^*)^{n-1}$ does not exceed $(k-1)(\ell-1)^{n-1}$. Therefore, the number of solutions does not exceed $(k-1)(\ell-1)^{n-1} + (\ell-1)^{n-1} = k(\ell-1)^{n-1}$, and the result follows.

In the second case, if there exists $(\alpha_1, \cdots, \alpha_k) \neq (\beta_1, \cdots, \beta_k)$ such that

$$\alpha_1(\boldsymbol{a}^{t_k})^{-1} + \cdots + \alpha_k \boldsymbol{a}^{t_{k-1}}(\boldsymbol{a}^{t_k})^{-1} = \beta_1(\boldsymbol{a}^{t_k})^{-1} + \cdots + \beta_k \boldsymbol{a}^{t_{k-1}(\boldsymbol{a}^{t_k})^{-1}}$$

in $\mathbb{F}_\ell$ then we have

$$(\alpha_1 - \beta_1)\boldsymbol{a}^0 + \cdots + (\alpha_k - \beta_k)\boldsymbol{a}^{t_{k-1}} = 0$$

in $\mathbb{F}_\ell$. Then by proceeding as previously by induction on k, for all n, one can see that the number of solutions $a \in (\mathbb{F}_\ell^*)^n$ does not exceed $(k-1)(\ell-1)^{n-1}$. □

For $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ such that $K_{\boldsymbol{a}}(X)$ is not a zero polynomial, we have by Lemma 2.2.1, that $w(K_{\boldsymbol{a}}(X)) \geq \frac{\ell}{\ell-(\ell-1-2\Delta)}$, since $\deg(K_{\boldsymbol{a}}(X)) \leq \ell-1$ and $K_{\boldsymbol{a}}(X)$ has at least $\ell-1-2\Delta$ roots of the g^u. Therefore, by Lemma 2.2.1, we have

$$\begin{cases} \binom{\deg(F_{\boldsymbol{a}})+k}{k} \geq \dfrac{\ell}{2\Delta+1} \\ w(F_{\boldsymbol{a}}) \geq \dfrac{\ell}{2\Delta+1} - 1 \end{cases}$$

for all but at most $2k(\ell-1)^{n-1}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ and the result follows. □

Remark 3.3.12. *Theorem 3.3.10 is non-trivial only when $\sharp S \geq (3\ell-2)/4$. Since $\sharp S \leq 2^{n-t}$, Theorem 3.3.10 only applies to settings where the message length n is greater than the sum of the bit-length of the underlying group order and t.*

The cardinality of the set S depends on A and on the secret key $\boldsymbol{a} = (a_1, \ldots, a_n) \in (\mathbb{F}_\ell^*)^n$. In the following lemma for certain condition on A and on n, we show that $\sharp S$ is close to ℓ for almost all secret key $\boldsymbol{a}$. This allows us to obtain Corollary 3.3.14 and for the forthcoming theorems in this Chapter to obtain non trivial lower bounds.

Lemma 3.3.13. *Let $\gamma > \delta > 0$ such that $n \geq (1+\gamma)\log(\ell-1)$.*
Let $t = \lfloor \min(1, (\gamma-\delta)/2)\log(\ell-1) \rfloor - 1$ and let $A \subseteq \{0, \cdots, 2^n-1\}$ such that $\{2^t x : x \in \{0, \ldots, 2^{n-t}-1\}\} \subseteq A$. Putting $\Gamma = \lfloor (\ell-1)2^{-t} \rfloor$, we obtain:

$$\sharp S \geq \ell - 1 - \Gamma,$$

for all but at most $(\ell-1)^{n-\delta}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

Proof. We denote again $S = \{\boldsymbol{a}^{2^t x} \in \mathbb{F}_\ell^* : 2^t x \in A\}$.
Putting $\Gamma = \lfloor (\ell-1)2^{-t} \rfloor$ and applying Lemma 3.2.1, we have $\sharp S \geq \ell - 1 - \Gamma$ for all but at most $2^{t-n}\Gamma^{-1}(\ell-1)^{n+2} \leq (\ell-1)^{n-\delta}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$. □

We apply Lemma 3.3.13 to Theorem 3.3.10 to obtain the following corollary:

Corollary 3.3.14. *Let $\gamma > \delta > 0$ such that $n \geq (1+\gamma)\log(\ell-1)$. Let $t = \lfloor \min(1, (\gamma-\delta)/2)\log(\ell-1) \rfloor - 1$ and $t_1, t_2, \cdots, t_k$ be fixed distinct integers such that $t_1, t_2, \cdots, t_k < 2^t$ and let $A \subseteq \{0, \cdots, 2^n-1\}$. For some $a \in (\mathbb{F}_\ell^*)^n$, let $F_{\boldsymbol{a}}(X_1, \ldots, X_k) \in \mathbb{F}_p[X_1, \cdots, X_k]$ such that Relation (3.3) holds for all $x \in A$. If $\{2^t x : x = 0, \ldots, 2^{n-t}-1\} \subseteq A$, we have*

$$\begin{cases} \binom{\deg(F_{\boldsymbol{a}})+k}{k} \geq \dfrac{1}{8}(\ell-1)^{\min(1,(\gamma-\delta)/2)} \\ w(F_{\boldsymbol{a}}) \geq \dfrac{1}{8}(\ell-1)^{\min(1,(\gamma-\delta)/2)} - 1 \end{cases}$$

for all but at most $2k(\ell-1)^{n-1} + (\ell-1)^{n-\delta}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

The proof is straightforward since, with the previous notation, we have in this case $\Delta < \Gamma$. Likewise Lemma 3.3.13 can be applied to the next theorems of this paper to obtain non-trivial lower bounds for almost all vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

For the cases where the cardinality of the set S is smaller than $(3\ell-2)/4$, Theorem 3.3.10 does not give a non-trivial lower bound on F's degree. In the next theorem, we obtain such a lower bound for much smaller sets S with $\#S \in [\sqrt{\ell}+1, (3\ell-2)/4]$. Theorem 3.3.15 only applies for univariate interpolation (i.e. $k=1$).

Theorem 3.3.15. *Let $t \geq 1$ be a fixed integer and let $A \subseteq \{0, \cdots, 2^n - 1\}$. For some $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, let $F_{\boldsymbol{a}}(X) \in \mathbb{F}_p[X]$ such that*

$$F_{\boldsymbol{a}}(f_{\boldsymbol{a}}(x)) = f_{\boldsymbol{a}}(x + t) \tag{3.5}$$

for all $x \in A$. For all but at most $2(\ell-1)^{n-1}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, we have

$$\deg(F_{\boldsymbol{a}}) \geq \frac{\sharp S(\sharp S - 1)}{\ell - 1}.$$

where $S = \{\boldsymbol{a}^{2^t x} \in \mathbb{F}_\ell^* : 2^t x \in A\}$.

Proof. As in the previous proof, we have

$$F_{\boldsymbol{a}}(g^u) = g^{u\boldsymbol{a}^t} \quad \text{for all } u \in S.$$

Consider

$$D = \{1 \leq b \leq \ell - 1 : b \equiv y - x \mod \ell, x, y \in S\}.$$

There exists $b \in D$ such that there are at least

$$\frac{\sharp S(\sharp S - 1)}{\sharp D} \geq \frac{\sharp S(\sharp S - 1)}{\ell - 1}$$

representations $b \equiv y - x \bmod \ell$, with $x, y \in S$. We choose this b and put

$$R = \{x \in S : b + x \equiv y \bmod \ell, y \in S\}.$$

Then we have

$$\sharp R \geq \frac{\sharp S(\sharp S - 1)}{\ell - 1}.$$

For $u \in R$, (since $g^x = g^{x+\ell}$, for all x), we have

$$\begin{aligned} F_{\boldsymbol{a}}(g^{u+b}) &= g^{(u+b)\boldsymbol{a}^t} \\ &= g^{u\boldsymbol{a}^t} \times g^{b\boldsymbol{a}^t} \\ &= F_{\boldsymbol{a}}(g^u) \times g^{b\boldsymbol{a}^t} \end{aligned}$$

Let $H_{\boldsymbol{a}}(X) = F_{\boldsymbol{a}}(g^b X) - g^{b\boldsymbol{a}^t} F_{\boldsymbol{a}}(X)$. Then $H_{\boldsymbol{a}}(X)$ has at least $\sharp R$ zeros. As in the previous proof, $H_{\boldsymbol{a}}(X) \neq 0$ for all but at most $2(\ell-1)^{n-1}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ and $\deg(H_{\boldsymbol{a}}) \leq \deg(F_{\boldsymbol{a}})$, we have

$$\deg(F_{\boldsymbol{a}}) \geq \frac{\sharp S(\sharp S - 1)}{\ell - 1}.$$

for all but at most $2(\ell-1)^{n-1}$ vectors $a \in (\mathbb{F}_\ell^*)^n$. □

In the following lemma, we show that there exists numerous sets A and corresponding S such that $\#S \in [\sqrt{\ell} + 1, (3\ell - 2)/4]$. For such sets Theorem 3.3.10 does not give a non-trivial lower bound on F's degree.

Lemma 3.3.16. *Let* $\frac{1}{\log(3)} - \frac{1}{2} > \delta > 0$ *(with* $\frac{1}{\log(3)} - \frac{1}{2} \simeq 0.1309\ldots$*).*
Let $t \geq 1$ *and* n *be integers such that* $n = t + \lceil (1/2+\delta)\log(\ell-1)\rceil + s$ *for some integer* s *such that* $0 \leq s \leq \log(3\ell-2) - 2 - \lceil (1/2+\delta)\log(\ell-1)\rceil$*. Let* $A \subseteq \{0, \cdots, 2^n - 1\}$ *such that* $\{2^t x : x \in \{0, \ldots, 2^{n-t}-1\}\} \subseteq A$*. Putting* $\gamma = 1 - \log(3)(1/2+\delta)$ *we obtain:*

$$(3\ell-2)/4 \geq \sharp S \geq (\ell-1)^{(1/2+\delta)}$$

for all but at most $3/2(\ell-1)^{n-\gamma}$ *vectors* $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$*.*

Proof. We denote again $S = \{\boldsymbol{a}^{2^t x} \in \mathbb{F}_\ell^* : 2^t x \in A\}$.
Putting $j = \lceil (1/2+\delta)\log(\ell-1)\rceil$ and applying Lemma 3.2.2, we obtain readily $\sharp S \geq (\ell-1)^{(1/2+\delta)}$ for all but at most $(3^j-1)(\ell-1)^{n-1} \leq 3/2(\ell-1)^{n-\gamma}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$. Since $\sharp S \leq 2^{j+s} \leq (3\ell-2)/4$, we obtain the desired result. □

For such sets A and S and parameters n, t, s given in Lemma 3.3.16, we have (using the notation of Theorem 3.3.15), that the degree of polynomial $F_{\boldsymbol{a}}$ satisfying (3.5) verifies $\deg(F_{\boldsymbol{a}}) \geq c \cdot \ell^{2\delta}$ for all but at most $2(\ell-1)^{n-1} + 3/2(\ell-1)^{n-\gamma}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ (where c is an absolute constant close to 1).

Remark 3.3.17. *This proof technique cannot be used to obtain a lower bound on the weight of a univariate polynomial* F *or on the degree of a multivariate polynomial* F *for* $k \geq 2$ *and it remains an open problem to improve Theorem 3.3.10 for smaller sets* S *with* $\#S \leq (3\ell-2)/4$ *in these settings.*

3.4. Polynomial Interpolation of the Naor-Reingold Pseudo-Random Function over Elliptic Curves

3.4.1. Polynomial Interpolation with fixed secret key

3.4.1.1. Univariate Interpolation of the Naor-Reingold Pseudo-Random Function over Elliptic Curves

In this section, p is an odd prime number, n is an integer, E is an elliptic curve over $\mathbb{F}_p$ and P is a point of the curve E with prime order ℓ (with $\ell \mid \#E(\mathbb{F}_p)$). We prove results on the univariate polynomial interpolation of the Naor-Reingold pseudo-random function from elliptic curves defined by $f_{\boldsymbol{a}}(x) = X([\boldsymbol{a}^x]P)$ for a secret key $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$ and an integer $x \in \{0, 1, \ldots, 2^n - 1\}$ (where $X(Q)$ denotes the abscissa of a point $Q \in E(\mathbb{F}_p)$). First, we consider interpolation over large sets of values.

Theorem 3.4.1. *Let* $E : y^2 = x^3 + \gamma x + \delta$ *be an elliptic curve over* $\mathbb{F}_p$ *with* $\gamma\delta \neq 0$*. Let* $t \geq 1$ *be a fixed integer and let* $A \subseteq \{0, \cdots, 2^n - 1\}$*. For some* $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$*, let* $F_{\boldsymbol{a}}(X) \in \mathbb{F}_p[X]$ *such that*

$$F_{\boldsymbol{a}}(f_{\boldsymbol{a}}(x)) = f_{\boldsymbol{a}}(x+t) \tag{3.6}$$

for all $x \in A$*. For all but at most* $2(\ell-1)^{n-1}$ *vectors* $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$*, we have*

$$\deg(F_{\boldsymbol{a}}) \geq \frac{2\sharp S - (\ell-1)}{14}.$$

where $S = \{\boldsymbol{a}^{2^t x} \in \mathbb{F}_\ell^* : 2^t x \in A\}$.

Proof. We have $F_{\boldsymbol{a}}(x_u) = x_{u\boldsymbol{a}^t}$ for all $u \in S$, where $x_t = X([t]P)$, for all $t \in \mathbb{F}_\ell$. We consider the $R = \{u \in S : 2u \in S\}$ with $\sharp R \geq \ell - 1 - 2\Delta$. For all $u \in R$, $2u \in S$ and $[2u]P \neq O$ and $F_{\boldsymbol{a}}(x_{2u}) = x_{2u\boldsymbol{a}^t}$ is well-defined in $\mathbb{F}_p$ and $x_{u\boldsymbol{a}^t}$ is thus not a root of ψ_2. Therefore, we have:

$$\begin{aligned} F_{\boldsymbol{a}}(x_{2u}) &= x_{2u\boldsymbol{a}^t} \\ &= \theta_2(x_{u\boldsymbol{a}^t})/\psi_2^2(x_{u\boldsymbol{a}^t}) \\ &= \theta_2(F_{\boldsymbol{a}}(x_u))/\psi_2^2(F_{\boldsymbol{a}}(x_u)), \quad \text{for all } u \in R. \end{aligned}$$

We thus get

$$F_{\boldsymbol{a}}\left(\frac{\theta_2(x_u)}{\psi_2^2(x_u)}\right) = \frac{\theta_2(F_{\boldsymbol{a}}(x_u))}{\psi_2^2(F_{\boldsymbol{a}}(x_u))}$$

for all $u \in R$. Finally, we consider the polynomial:

$$H_{\boldsymbol{a}}(X) = \psi_2^{2d}(X)\psi_2^2(F_{\boldsymbol{a}}(X))\left(F_{\boldsymbol{a}}\left(\frac{\theta_2(X)}{\psi_2^2(X)}\right) - \frac{\theta_2(F_{\boldsymbol{a}}(X))}{\psi_2^2(F_{\boldsymbol{a}}(X))}\right),$$

where $d = \deg(F_{\boldsymbol{a}})$. The polynomial $H_{\boldsymbol{a}}(X)$ has at least $\sharp R/2$ zeros. If $\boldsymbol{a}^t \neq \pm 1$, we will have $F_{\boldsymbol{a}}(X) \neq X$ and by Lemma 2.3.1, it will imply that there exists $\alpha \in \overline{\mathbb{F}_p}$ such that $\psi_2^2(F_{\boldsymbol{a}}(\alpha)) = 0$ and $\psi_2^2(\alpha) \neq 0$. Hence, we have $H_{\boldsymbol{a}}(\alpha) = -\theta_2(F_{\boldsymbol{a}}(\alpha))\psi_2^{2d}(\alpha) \neq 0$, since $\theta_2(X)$ and $\psi_2^2(X)$ have no common zeros.

Therefore, $H_{\boldsymbol{a}}(X)$ is a non-zero polynomial and $\deg(H_{\boldsymbol{a}}) \leq 7d$. Then we get that $7d \geq \sharp R/2$ and then $d \geq \frac{\ell-1-2\Delta}{14}$. Since $\boldsymbol{a}^t \neq \pm 1$ for all but at most $2(\ell-1)^{n-1}$ vectors $a \in (\mathbb{F}_\ell^*)^n$, the result follows. □

Theorem 3.4.1 is only non-trivial if $\sharp S \geq (\ell + 13)/2$. Again, the cardinality of the set S depends on A and on the secret key $\boldsymbol{a} = (a_1, \ldots, a_n) \in (\mathbb{F}_\ell^*)^n$, but using again Lemma 3.3.13, we can easily obtain (as in Corollary 3.3.14) non-trivial lower bounds for specific sets A and parameter n.

In the following theorem, we obtain a lower bound for smaller sets S.

Theorem 3.4.2. *Let $t \geq 1$ be a fixed integer, $A \subseteq \{0, \cdots, 2^n - 1\}$, $0 < \epsilon < 1$ and $S = \{\boldsymbol{a}^{2^t x} \in \mathbb{F}_\ell^* : 2^t x \in A\}$ with $\sharp S \geq \frac{2(\ell-1)}{\epsilon \log(\ell)}$. For some $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, let $F_{\boldsymbol{a}}(X) \in \mathbb{F}_p[X]$ such that*

$$F_{\boldsymbol{a}}(f_{\boldsymbol{a}}(x)) = f_{\boldsymbol{a}}(x + t) \tag{3.7}$$

for all $x \in A$. For all but at most $2(\ell-1)^{n-1}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, we have

$$\deg(F_{\boldsymbol{a}}) \geq \frac{\sharp S}{4\epsilon \log(\ell) \times \ell^{2\epsilon}}.$$

Proof. We have $F_{\boldsymbol{a}}(x_u) = x_{u\boldsymbol{a}^t}$ for all $u \in S$ where, as above, we denote $x_t = X([t]P)$, for all $t \in \mathbb{F}_\ell$. Let K be an integer and let us consider the sets

$$S_i = \{1 \leq b \leq \ell - 1 : 2^i m \equiv b \bmod \ell,\ m \in S\},$$

for $0 \leq i \leq K$, and $R_{i,j} = S_i \cap S_j$ for $0 \leq i < j \leq K$. We have

$$(K+1)\sharp S - \sum_{0 \leq i < j \leq K} \sharp R_{i,j} \leq \sharp\left(\bigcup_{i=0}^{K} S_i\right) \leq \ell - 1.$$

Therefore, there is a pair $0 \leq i < j \leq K$ such that

$$\sharp R_{0,j-i} = \sharp R_{i,j} \geq \frac{2((K+1)\sharp S - (\ell-1))}{K(K+1)}.$$

For $u \in R_{0,j-i}$, there exists a unique $m \in S$ such that $2^{j-i}m \equiv u \bmod \ell$, with $u \in S$ and the corresponding m's are distinct for two different u's. Since $x_k = x_{k+l}$ for all k, then we have $F_{\boldsymbol{a}}(x_{2^{j-i}m}) = x_{2^{j-i}m\boldsymbol{a}^t}$ for at least $\sharp R_{0,j-i}$ different $m \in S$. For each such m, we have

$$\begin{aligned} F_{\boldsymbol{a}}\left(\frac{\theta_{2^{j-i}}(x_m)}{\psi^2_{2^{j-i}}(x_m)}\right) &= x_{2^{j-i}m\boldsymbol{a}^t} \\ &= \theta_{2^{j-i}}(x_{m\boldsymbol{a}^t})/\psi^2_{2^{j-i}}(x_{m\boldsymbol{a}^t}) \\ &= \theta_{2^{j-i}}(F_{\boldsymbol{a}}(x_m))/\psi^2_{2^{j-i}}(F_{\boldsymbol{a}}(x_m)), \end{aligned}$$

since $m \in S$. Finally, we consider the polynomial

$$H_{\boldsymbol{a}}(X) = \psi^{2d}_{2^{j-i}}(X)\psi^2_{2^{j-i}}(F_{\boldsymbol{a}}(X))\left(F_{\boldsymbol{a}}\left(\frac{\theta_{2^{j-i}}(X)}{\psi^2_{2^{j-i}}(X)}\right) - \frac{\theta_{2^{j-i}}(F_{\boldsymbol{a}}(X))}{\psi^2_{2^{j-i}}(F_{\boldsymbol{a}}(X))}\right),$$

where $d = \deg(F_{\boldsymbol{a}})$.

The polynomial $H_{\boldsymbol{a}}(X)$ has at least $\sharp R_{0,j-i}$ zeros. Since $d \geq 2$ and 2^{j-i} and p are coprime, then by Lemma 2.3.2, there exists $\alpha \in \overline{\mathbb{F}_p}$ such that $\psi^2_{2^{j-i}}(F_{\boldsymbol{a}}(\alpha)) = 0$ and $\psi^2_{2^{j-i}}(\alpha) \neq 0$. Hence, we have $H_{\boldsymbol{a}}(\alpha) = -\theta_{2^{j-i}}(F_{\boldsymbol{a}}(\alpha))\psi^{2d}_{2^{j-i}}(\alpha) \neq 0$, since $\theta_{2^{j-i}}(X)$ and $\psi^2_{2^{j-i}}(X)$ have no common zeros.

Therefore $H_{\boldsymbol{a}}(X)$ is a non-zero polynomial and $\deg(H_{\boldsymbol{a}}) \leq d(2(2^{j-i})^2 - 1)$ and we get

$$d \geq \frac{\sharp R_{0,j-i}}{2(2^{2(j-i)+1}-1)} \geq \frac{(K+1)\sharp S - (\ell-1)}{K(K+1)(2^{2(j-i)+1}-1)}.$$

Since $j - i \leq K$, then we have

$$\begin{aligned} d \geq \frac{(K+1)\sharp S - (\ell-1)}{K(K+1)(2^{2K+1}-1)} &\geq \frac{1}{2^{2K+1}-1}\left(\frac{\sharp S}{K} - \frac{\ell-1}{K(K+1)}\right) \\ &\geq \frac{\sharp S}{K(2^{2K+1}-1)}\left(1 - \frac{\ell-1}{\sharp S(K+1)}\right). \end{aligned}$$

Letting $K = \lfloor \epsilon \log(\ell) \rfloor$, for any $0 < \epsilon < 1$, we have

$$d \geq \frac{\sharp S}{2\epsilon \log(\ell) \times \ell^{2\epsilon}}\left(1 - \frac{\ell-1}{\sharp S \epsilon \log(\ell)}\right) \geq \frac{\sharp S}{4\epsilon \log(\ell) \times \ell^{2\epsilon}}.$$

□

Theorem 3.4.2 also applies to numerous sets A and S for which Theorem 3.4.1 does not apply. For instance, we can consider parameters n, t, s given in Lemma 3.3.16 such that $2^{n-t-s} \geq \frac{2(\ell-1)}{\epsilon \log(\ell)}$.

3.4.1.2. Bivariate Interpolation of the Naor-Reingold Pseudo-Random Function over Elliptic Curves

It seems rather difficult to obtain an analogue of Theorem 3.3.10 in the case of elliptic curves. In this section, we use the methods from [LW03b] and we prove results on bivariate interpolation of the Naor-Reingold pseudo-random function from elliptic curves (but in a slightly different setting). We use the notation from the previous section and, as before, we consider first interpolation over large sets of values.

Theorem 3.4.3. *Let $A_1, A_2 \subseteq \{0, \cdots, 2^n - 1\}$ and $t \geq 1$ be an integer. For some $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, let $F_{\boldsymbol{a}}(X, Y) \in \mathbb{F}_p[X, Y]$ such that*

$$F_{\boldsymbol{a}}(f_{\boldsymbol{a}}(x), f_{\boldsymbol{a}}(x')) = f_{\boldsymbol{a}}(x + x') \tag{3.8}$$

for all $(x, x') \in A_1 \times A_2$. We have

$$\deg(F_{\boldsymbol{a}}) \geq \min\left(\lfloor(\ell - 1)/\Delta\rfloor - 2;\ \lceil(\sharp S_2 - 1)^{1/3}\rceil - 2\right).$$

where $\Delta = \ell - 1 - \sharp S_1$ for the set $S_1 = \{\boldsymbol{a}^x : x \in A_1\,,\ x < 2^t\}$ and where $S_2 = \{\boldsymbol{a}^{2^t x'} \in \mathbb{F}_\ell^* : 2^t x' \in A_2\}$.

Proof. We may suppose $\sharp S_2 \geq 10$ since otherwise the result is trivial. We denote

$$d = \min\left(\lfloor(\ell - 1)/\Delta\rfloor - 2;\ \lceil(\sharp S_2 - 1)^{1/3}\rceil - 2\right).$$

We have

$$F_{\boldsymbol{a}}(x_u, x_{u'}) = x_{uu'}, \quad \text{for all } u \in S_1 \text{ and } u' \in S_2.$$

Let R be the set of $u \in S_1$ such that

$$ia \bmod \ell \in S_1, \quad \forall i \in \{1, \cdots, d + 1\}.$$

The cardinality of R is at least $\ell - 1 - \Delta(d + 1)$. Let $u \in R$, then we have

$$F_{\boldsymbol{a}}(x_{(i+1)u}, x_{v_j}) = x_{(i+1)uv_j}, \quad \text{for all } 0 \leq i, j \leq d,$$

where $v_0, \cdots, v_d$ are any distinct elements of S_2.

Let us suppose that $\deg_X(F_{\boldsymbol{a}}), \deg_Y(F_{\boldsymbol{a}}) \leq d$ namely

$$F_{\boldsymbol{a}}(X, Y) = \sum_{i,j=0}^{d} c_{i,j} X^i Y^j,$$

then we have for $0 \leq k, \ell \leq d$,

$$x_{(k+1)uv_\ell} = \sum_{i,j=0}^{d} c_{i,j} x_{(k+1)u}^i x_{v_\ell}^j.$$

Then $F_{\boldsymbol{a}}$'s coefficients are determined by the following matrix equation:

$$\begin{aligned} C &= \begin{pmatrix} c_{0,0} & \cdots & c_{0,d} \\ \vdots & & \vdots \\ c_{d,0} & \cdots & c_{d,d} \end{pmatrix} \\ &= \begin{pmatrix} x_u^0 & \cdots & x_u^d \\ \vdots & & \vdots \\ x_{(d+1)u}^0 & \cdots & x_{(d+1)u}^d \end{pmatrix}^{-1} \begin{pmatrix} x_{uv_0} & \cdots & x_{uv_d} \\ \vdots & & \vdots \\ x_{(d+1)uv_0} & \cdots & x_{(d+1)uv_d} \end{pmatrix} \begin{pmatrix} x_{v_0}^0 & \cdots & x_{v_d}^0 \\ \vdots & & \vdots \\ x_{v_0}^d & \cdots & x_{v_d}^d \end{pmatrix} \end{aligned}$$

The matrix C is non-singular if and only if the middle matrix on the right hand is non-singular. A subset $\{v_0, \cdots, v_d\}$ of S_2 with this property exists if and only if the vectors $T_k = (x_{kub})_{b \in S_2}$ for $k \in \{1, \cdots, d+1\}$ are linearly independent. If these vectors were linearly dependent, then there would exist an integer ω with $1 \leq \omega \leq d+1$ and coefficients $d_1, \cdots, d_\omega \in \mathbb{F}_p$, $d_\omega \neq 0$, such that

$$\sum_{k=1}^{\omega} d_k x_{kub} = 0, \qquad b \in S_2.$$

As at most two points with first coordinate equal to 0 exist on the elliptic curve and $\sharp S_2 \geq 3$, we get $\omega \geq 2$. Since $x_{kub} = \theta_k(x_{ub})/\psi_k^2(x_{ub})$, the polynomial

$$H(X) = \sum_{k=1}^{\omega} d_k \theta_k(X) \prod_{j=1, j\neq k}^{\omega} \psi_j^2(X)$$

has at least $\lfloor \sharp S_2/2 \rfloor$ zeros and degree at most

$$1 + \sum_{k=1}^{\omega} (k^2 - 1) = (2\omega^3 + 3\omega^2 - 5\omega + 6)/6 \leq \omega^3/2 \leq (d+1)^3/2.$$

Since $p \nmid \omega$, then points of order ω on E exist over $\overline{\mathbb{F}_p}$. Let $\alpha \in \overline{\mathbb{F}_p}$ be the first coordinate of a point of order ω. Then we have $\psi_\omega^2(\alpha) = 0$ and $H(\alpha) = d_\omega \theta_\omega(\alpha) \prod_{j=1}^{\omega-1} \psi_j^2(\alpha) \neq 0$.

The polynomial $H(X)$ is a non-zero polynomial and we have $(d+1)^3/2 \geq \lfloor \sharp S_2/2 \rfloor$ in contradiction with the definition of d. This shows that C is not singular and in particular each row of C has at least one non-zero entry and we have $\deg(F_{\boldsymbol{a}}) \geq \deg_X(F_{\boldsymbol{a}}) \geq d$. □

Theorem 3.4.3 is non-trivial only for $\sharp S_1 \geq (\ell - 1)/2$ and we can obtain (as in Corollary 3.3.14) non-trivial lower bounds on the degree of the interpolating polynomial for specific sets A_1 and A_2 and parameter t.

Lemma 3.4.4. *Let $m \geq 1$ be an integer, $\delta > 0$ and t be an integer such that $t \geq (1+\delta)\log(\ell-1) + m + 1$ and $n - t \geq (1+\delta)\log(\ell-1) + 2$.*
Let $A_1, A_2 \subseteq \{0, \cdots, 2^n - 1\}$ be two sets such that $\{0, \ldots, 2^t - 1\} \subseteq A_1$ and $\{2^t x' : x' \in \{0, \ldots, 2^{n-t} - 1\}\} \subseteq A_2$. Let $S_1 = \{\boldsymbol{a}^x : x \in A_1, x < 2^t\}$ and $S_2 = \{\boldsymbol{a}^{2^t x'} \in \mathbb{F}_\ell^* : 2^t x' \in A_2\}$, we have

$$\sharp S_1 \geq \ell - 1 - \lfloor (\ell-1)2^{-m} \rfloor \quad \text{and} \quad \sharp S_2 \geq (\ell-1)/2 + 1,$$

for all but at most $2(\ell-1)^{n-\delta}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

Proof. Let $\Delta = \lfloor (\ell-1)2^{-m} \rfloor$. Applying Lemma 3.2.1, we have $\sharp S_1 \geq \ell - 1 - \Delta$ for all but at most $2^{-t}\Delta^{-1}(\ell-1)^{t+2}(\ell-1)^{n-t} \leq (\ell-1)^{n-\delta}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

Likewise, applying Lemma 3.2.1 with $\Delta' = (\ell-1)/2 - 1$, we have $\sharp S_2 \geq (\ell-1)/2 + 1$ for all but at most $2^{t-n}\Delta'^{-1}(\ell-1)^{n-t+2}(\ell-1)^t \leq (\ell-1)^{n-\delta}$ vectors $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$. □

We apply Lemma 3.4.4 to Theorem 3.4.3 to obtain the following corollary:

Corollary 3.4.5. *Let $m \geq 1$ be an integer and $\delta > 0$ such that $t \geq (1+\delta)\log(\ell-1) + m + 1$ and $n - t \geq (1+\delta)\log(\ell-1) + 2$.*
Let $A_1 \subseteq \{0, \cdots, 2^n - 1\}$ such that $\{0, \ldots, 2^t - 1\} \subseteq A_1$ and $A_2 \subseteq \{0, \cdots, 2^n - 1\}$ such that $\{2^t x' : x' \in \{0, \ldots, 2^{n-t} - 1\}\} \subseteq A_2$. For some $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, let $F_{\boldsymbol{a}}(X, Y) \in \mathbb{F}_p[X, Y]$ such that

$$F_{\boldsymbol{a}}(f_{\boldsymbol{a}}(x), f_{\boldsymbol{a}}(x')) = f_{\boldsymbol{a}}(x + x') \tag{3.9}$$

for all $(x, x') \in A_1 \times A_2$. *We have*

$$\deg(F_{\boldsymbol{a}}) \geq \min\left(2^m - 2;\, ((\ell-1)/2)^{1/3} - 2\right),$$

for all but at most $2(\ell-1)^{n-\delta}$ *vectors* $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

The proof is straightforward since, with the of notation of Theorem 3.4.3, we have in this case $\Delta \leq (\ell-1)2^{-m}$ and $\sharp S_2 - 1 \geq (\ell-1)/2$.

To conclude this section, we obtain a simple result for smaller sets S_1.

Theorem 3.4.6. *Let* $A_1, A_2 \subseteq \{0, \cdots, 2^n - 1\}$ *and* $t \geq 1$ *be an integer. For some* $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, *let* $F_{\boldsymbol{a}}(X, Y) \in \mathbb{F}_p[X, Y]$ *such that*

$$F_{\boldsymbol{a}}(f_{\boldsymbol{a}}(x), f_{\boldsymbol{a}}(x')) = f_{\boldsymbol{a}}(x + x') \tag{3.10}$$

for all $(x, x') \in A_1 \times A_2$. *We have*

$$\deg(F_{\boldsymbol{a}}) \geq \frac{\sharp S_1}{8}.$$

where $S_1 = \{\boldsymbol{a}^x : x \in A_1\,,\, x < 2^t\}$ *and* $S_2 = \{\boldsymbol{a}^{2^t x'} \in \mathbb{F}_\ell^* : 2^t x' \in A_2\}$ *if there exists* $v \in S_2$ *such that* $2v \in S_2$.

Proof. We have

$$F_{\boldsymbol{a}}(x_u, x_v) = x_{uv} \quad \textit{and} \quad F_{\boldsymbol{a}}(x_u, x_{2v}) = x_{2uv} \quad \text{for all } u \in S_1.$$

Hence

$$F_{\boldsymbol{a}}\left(x_u, \frac{\theta_2(x_v)}{\psi_2^2(x_v)}\right) = \frac{\theta_2(x_{uv})}{\psi_2^2(x_{uv})} = \frac{\theta_2(F_{\boldsymbol{a}}(x_u, x_v))}{\psi_2^2(F_{\boldsymbol{a}}(x_u, x_v))} \quad \text{for all } u \in S_1.$$

Finally, we consider the polynomial

$$U(X) = \psi_2^2(F_{\boldsymbol{a}}(X, x_v))\left(F_{\boldsymbol{a}}(X, \frac{\theta_2(x_v)}{\psi_2^2(x_v)}) - \frac{\theta_2(F_{\boldsymbol{a}}(X, x_v))}{\psi_2^2(F_{\boldsymbol{a}}(X, x_v))}\right).$$

We have $\deg(U) \leq 4\deg(F_{\boldsymbol{a}})$. Let γ be a root of $\psi_2^2(X)$ and β such that $F_{\boldsymbol{a}}(\beta, x_v) = \gamma$. Then

$$U(\beta) = -\theta_2(F_{\boldsymbol{a}}(\beta, x_v)) \neq 0,$$

and U is non-zero polynomial. Since U has at least $\sharp S_1/2$ zeros, it follows that $4\deg(F_{\boldsymbol{a}}) \geq \sharp S_1/2$ i.e. $\deg(F_{\boldsymbol{a}}) \geq \frac{\sharp S_1}{8}$. □

The condition on S_2 in the statement of Theorem 3.4.6 is achieved trivially when $\sharp S_2 > \frac{\ell-1}{2}$. It is worth mentioning that Theorem 3.4.6 also applies to many other sets. In the following lemma, we show that there exists numerous sets A_1, A_2 and corresponding S_1, S_2 such that $\sharp S_1 \in [\sqrt{\ell}+1, (\ell-1)/2]$ and $\sharp S_2 > (\ell-1)/2$. For such sets Theorem 3.4.6 gives a non-trivial lower bound on the degree of the interpolating polynomial while Theorem 3.4.3 does not give a non-trivial lower bound on it. We apply Lemmas 3.2.1 and 3.2.2 like in the proof of Lemmas 3.3.13, 3.3.16 and 3.4.4 to obtain the following lemma:

Lemma 3.4.7. *Let* $\frac{1}{\log(3)} - \frac{1}{2} > \delta_1 > 0$ *(with* $\frac{1}{\log(3)} - \frac{1}{2} \simeq 0.1309\ldots$*) and* $\delta_2 > 0$.
Let t *and* n *be integers such that* $t = \lceil (1/2+\delta_1)\log(\ell-1)\rceil + s$ *for some integer* s *such that* $0 \leq s \leq \log(\ell-1) - 1 - \lceil (1/2+\delta_1)\log(\ell-1)\rceil$ *and* $n - t \geq (1+\delta_2)\log(\ell-1) + 2$. *Let* $A_1 \subseteq \{0,\cdots,2^n-1\}$ *such that* $\{0,\ldots,2^t-1\}\} \subseteq A_1$ *and* $A_2 \subseteq \{0,\cdots,2^n-1\}$ *such that* $\{2^t x : x \in \{0,\ldots,2^{n-t}-1\}\} \subseteq A_2$. *Putting* $\gamma = 1 - \log(3)(1/2+\delta_1)$ *we obtain:*

$$(\ell-1)/2 \geq \sharp S_1 \geq (\ell-1)^{(1/2+\delta_1)} \quad \textit{and} \quad \sharp S_2 \geq (\ell-1)/2 + 1,$$

for all but at most $3/2(\ell-1)^{n-\gamma} + (\ell-1)^{n-\delta_2}$ *vectors* $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

We then apply Lemma 3.4.7 to Theorem 3.4.6 to obtain the following corollary:

Corollary 3.4.8. *Let* $\frac{1}{\log(3)} - \frac{1}{2} > \delta_1 > 0$, $\delta_2 > 0$ *and* $\gamma = 1 - \log(3)(1/2+\delta_1)$.
Let t *and* n *be integers such that* $t = \lceil (1/2+\delta_1)\log(\ell-1)\rceil + s$ *for some integer* s *such that* $0 \leq s \leq \log(\ell-1) - 1 - \lceil (1/2+\delta_1)\log(\ell-1)\rceil$ *and* $n - t \geq (1+\delta_2)\log(\ell-1) + 2$. *Let* $A_1 \subseteq \{0,\cdots,2^n-1\}$ *such that* $\{0,\ldots,2^t-1\}\} \subseteq A_1$ *and* $A_2 \subseteq \{0,\cdots,2^n-1\}$ *such that* $\{2^t x : x \in \{0,\ldots,2^{n-t}-1\}\} \subseteq A_2$. *For some* $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$, *let* $F_{\boldsymbol{a}}(X,Y) \in \mathbb{F}_p[X,Y]$ *such that*

$$F_{\boldsymbol{a}}(f_{\boldsymbol{a}}(x), f_{\boldsymbol{a}}(x')) = f_{\boldsymbol{a}}(x+x') \tag{3.11}$$

for all $(x,x') \in A_1 \times A_2$. *We have*

$$\deg(F_{\boldsymbol{a}}) \geq (\ell-1)^{(1/2+\delta_1)}/8,$$

for all but at most $3/2(\ell-1)^{n-\gamma} + (\ell-1)^{n-\delta_2}$ *vectors* $\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n$.

The proof is straightforward since, with the notation of Theorem 3.4.6, we have in this case $\sharp S_1 \geq (\ell-1)^{(1/2+\delta_1)}$ and there exists $v \in S_2$ such that $2v \in S_2$ since $\sharp S_2 > (\ell-1)/2$

3.4.2. Polynomial Interpolation with variable secret key

In this section, p is an odd prime number and we prove results on the approximation by a polynomial with $k \geq 1$ variables of the Naor-Reingold pseudo-random function from elliptic curves. For a positive integer t, we denote $X(tP)$ by x_t.

Theorem 3.4.9. *Let* n *be an integer,* $S \subseteq (\mathbb{F}_\ell^*)^n$, *with* $|S| = (\ell-1)^n - s$. *Let* $f \in \mathbb{F}_p[X_1,\ldots,X_k]$, *be a polynomial satisfying:*

$$f(x_{\boldsymbol{a}^{x^1}},\ldots,x_{\boldsymbol{a}^{x^k}}) = x_{\boldsymbol{a}^{x^{k+1}}}, \quad \textit{for all } \boldsymbol{a} = (a_1,\ldots,a_n) \in S,$$

for some values $x^1,\ldots,x^n \in \{0,\cdots,2^n-1\}$ *such that* $x_1^1 = 1$, $x_1^{k+1} = 1$ *and* $x_1^i = 0$ *for* $i \in \{2,\ldots,k\}$. *Then*

$$\deg(f) \geq \frac{\ell-1}{14} - \frac{s}{7(\ell-1)^{n-1}}.$$

Proof. Let $W = \{\boldsymbol{a} \in (\mathbb{F}_\ell^*)^n : \boldsymbol{a} = (a_1,\ldots,a_n) \in S \text{ and } (2a_1,\ldots,a_n) \in S\}$, then $|W| \geq (\ell-1)^n - 2s$. Then there exists $\boldsymbol{b} = (b_2,\ldots,b_n) \in (\mathbb{F}_l^*)^{n-1}$ such that the set $T = \{a_1 \in (\mathbb{F}_l^*) : a' = (a_1,b_2,\ldots,b_n) \in W\}$ satisfies $|T| \geq (\ell-1) - 2s/(\ell-1)^{n-1}$. Thus for all $a_1 \in T$, putting $\boldsymbol{a'} = (a_1,b_2,\ldots,b_n)$, we have:

$$\begin{cases} f(x_{\boldsymbol{a'}^{x^1}},\ldots,x_{\boldsymbol{b}^{x^k}}) = x_{\boldsymbol{a'}^{x^{k+1}}} \\ f(x_{2\boldsymbol{a'}^{x^1}},\ldots,x_{\boldsymbol{b}^{x^k}}) = x_{2\boldsymbol{a'}^{x^{k+1}}} \end{cases}$$

Hence for all $a_1 \in T$, we have:

$$f\left(\frac{\theta_2(x_{a_1\boldsymbol{b}^{x^1}})}{\psi_2^2(x_{a_1\boldsymbol{b}^{x^1}})},\ldots,x_{\boldsymbol{b}^{x^k}}\right)=\frac{\theta_2(f((x_{a_1\boldsymbol{b}^{x^1}},\ldots,x_{\boldsymbol{b}^{x^k}})))}{\psi_2^2(f((x_{a_1\boldsymbol{b}^{x^1}},\ldots,x_{\boldsymbol{b}^{x^k}})))}$$

We consider the polynomial:

$$H(X)=\psi_2^{2d}(X)f\left(\frac{\theta_2(X)}{\psi_2^2(X)},\ldots,x_{\boldsymbol{b}^{x^k}}\right)\psi_2^2(f(X,\ldots,x_{\boldsymbol{b}^{x^k}}))-\psi_2^{2d}(X)\theta_2(f(X,\ldots,x_{\boldsymbol{b}^{x^k}}))$$

where $d=\deg_{X_1} f$. Let γ be a root of $\psi_2^2(X)$,then $H(\gamma)=c\theta_2^d(\gamma)\neq 0$, for some coefficient $c\in\mathbb{F}_p^*$. Thus H is a nonzero polynomial and we have $\deg H\leq 7d\leq 7\deg f$. Since H has at least $|T|/2\geq(\ell-1)/2-s/(\ell-1)^{n-1})$ zeros, we obtain:

$$\deg(f)\geq\frac{\ell-1}{14}-\frac{s}{7(\ell-1)^{n-1}}.$$

□

Chapter 4.

Distribution and Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function

In 2005, Dodis and Yampolskiy [DY05] proposed an efficient pseudo-random function family which takes inputs in $\{1, \ldots, d\}$ (for some parameter $d \in \mathbb{N}$) and outputs an element in a group $\mathbb{G}$ (multiplicatively written) of prime order t with generator g. The secret key is a scalar $x \in \mathbb{Z}_t^*$ and the pseudo-random function is defined by:

$$\begin{array}{rccl} V_x & : & \{1, \ldots, d\} & \longrightarrow & \mathbb{G} \\ & & m & \longmapsto & V_x(m) = g^{\frac{1}{x+m}} \quad \text{if } x + m \neq 0 \bmod t \text{ and } 1_{\mathbb{G}} \text{ otherwise.} \end{array}$$

The Dodis-Yampolskiy pseudo-random function family has found numerous applications in cryptography (e.g., for compact e-cash [CHL05] or anonymous authentication [CHK+06]). Dodis and Yampolskiy showed that their construction has some very attractive security properties, provided that some assumption about the hardness of breaking the so-called *Decision Diffie-Hellman Inversion* problem holds in $\mathbb{G}$ [DY05]. This assumption is non-standard and Cheon [Che10] proved that it is stronger than the classical discrete logarithm assumption in $\mathbb{G}$. In this Chapter, we study the distribution of the Dodis-Yampolskiy pseudo-random function over finite fields and over elliptic curves and prove lower bounds on the degree of polynomials which interpolate these functions. The first section deals with the distribution of the Dodis-Yampolskiy pseudo-random function over finite fields and over elliptic curves. In the second section, we prove lower bounds on the degree of polynomials interpolating this pseudo-random function over finite fields and we conclude the chapter by proving lower bounds on the degree of polynomials interpolating this pseudo-random function over an elliptic curve.

Contents

4.1. Distribution of the Dodis-Yampolskiy Pseudo-Random Functions . 53
4.1.1. Distribution of the Dodis-Yampolskiy Pseudo-Random Function over Finite Fields . . . 53
4.1.2. Distribution of the Dodis-Yampolskiy Pseudo-Random Function over Elliptic Curves . . . 55
4.2. Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function over Finite Fields . . . 56
4.3. Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function over Elliptic Curves . . . 57

4.1. Distribution of the Dodis-Yampolskiy Pseudo-Random Functions

For a sequence of N points $\Gamma = (\gamma_{0,n}, \ldots, \gamma_{s-1,n})_{n\in\{1,\ldots,N\}}$ in the s-dimensional unit cube, we define its *discrepancy* by D_Γ:

$$D_\Gamma = \sup_{B\subseteq[0,1)^s} \left| \frac{T_\Gamma(B)}{N} - |B| \right|,$$

where $T_\Gamma(B)$ denotes the number of points of the sequence Γ in a box B (i.e. a polyhedron $[\alpha_0, \beta_0) \times \cdots \times [\alpha_{s-1}, \beta_{s-1}) \subseteq [0,1)^s$) of volume $|B|$ and the supremum is taken over all such boxes. For an integer vector $a = (a_0, \ldots, a_{s-1}) \in \mathbb{Z}^s$, we define $|a| = max_{\nu\in\{0,\ldots,s-1\}}|a_\nu|$ and $r(a) = \prod_{\nu=0}^{s-1} \max\{|a_\nu|, 1\}$. A critical issue with a pseudo-random sequence Γ in the s-dimensional unit cube is that it may not be perfectly equidistributed namely spread in a given volume of the s-dimensional unit cube (as it is the case in quasi-Monte Carlo methods). Sequences in $[0,1)^s$ with low discrepancy will spread over $[0,1)^s$ as uniformly as possible, reducing gaps and clustering of points. In order to show that a sequence Γ is uniformly distributed, we need to show that its discrepancy D_Γ is very small (i.e. tends to 0). The following lemma is our main tool for finding non-trivial upper bound for the discrepancy. It is a slightly weaker form of the Koksma-Szüsz inequality [DT97, Theorem 1.21]. The implied constant in the symbol "$\ll$" depends on the integer s.

Lemma 4.1.1. *For any integer $L > 1$ and any sequence Γ of N points, we have*

$$D_\Gamma \ll \frac{1}{L} + \frac{1}{N} \sum_{0<|a|<L} \frac{1}{r(a)} \left| \sum_{n=1}^{N} e\left(\sum_{\nu=0}^{s-1} a_\nu \gamma_{\nu,n} \right) \right|,$$

where the sum is taken over all integer vectors $a \in \mathbb{Z}^s$ with $0 < |a| < L$.

We also need the well-known orthogonality relation:

$$\sum_{\eta=0}^{m-1} e_m(\eta\lambda) = \begin{cases} 0 & \text{if } \lambda \neq 0 \bmod m \\ m & \text{otherwise} \end{cases} \tag{4.1}$$

and the inequality [[IK04], Bound (8.6)] (which holds for any integers m and M with $1 \leq M \leq m$):

$$\sum_{\eta=0}^{m-1} \left| \sum_{\lambda=1}^{M} e_m(\eta\lambda) \right| \ll m \log m. \tag{4.2}$$

4.1.1. Distribution of the Dodis-Yampolskiy Pseudo-Random Function over Finite Fields

Let $q = p^r$ be a prime power for some integer $r > 1$, let $g \in \mathbb{F}_q^*$ be an element of prime order t. For $x \in \mathbb{Z}_t$ and $d \leq t$, we denote by $D_x(d)$ the discrepancy of the points $(V_{x,1}(n)/p, \ldots, V_{x,r}(n)/p)$ for $1 \leq n \leq d$, where $V_x(n) = g^{\frac{1}{x+n}} \in \mathbb{F}_{p^r}$ and $V_x(n) = V_{x,1}(n)\beta_1 + \cdots + V_{x,r}(n)\beta_r$, where $\{\beta_1, \ldots, \beta_r\}$ is an ordered basis of $\mathbb{F}_{p^r}$ over $\mathbb{F}_p$.

Theorem 4.1.2. *For any $x \in \mathbb{Z}_t$, any integers $k \geq 2$, $\ell \geq 1$ and $1 \leq d \leq t$, we have:*

$$D_x(d) \leq \frac{t^{1-\alpha_{k,l}} q^{\beta_{k,l}+o(1)}}{d},$$

where $\alpha_{k,l} = \frac{1}{2(2k+l)} - \frac{1}{4kl}$ *and* $\beta_{k,l} = \frac{1}{4(2k+l)}$.

Proof. From Lemma 4.1.1, we derive

$$D_x(d) \ll \frac{1}{p} + \frac{1}{d} \sum_{0<|a|<p} \frac{1}{r(a)} \left| \sum_{n=1}^{d} e_p \left(\sum_{j=1}^{r} a_j V_{x,j}(n) \right) \right|,$$

where $a = (a_1, \ldots, a_r)$. Set

$$S_d(a) = \sum_{n=1}^{d} e_p(\sum_{j=1}^{r} a_j V_{x,j}(n)).$$

Let $\{\delta_1, \ldots, \delta_r\}$ be the dual basis of the given ordered basis $\{\beta_1, \ldots, \beta_r\}$. For $j \in \{1, \ldots, r\}$ and $n \in \{1, \ldots, d\}$, we have $V_{x,j}(n) = \mathrm{Tr}(\delta_j V_x(n))$, where Tr denotes the trace of $\mathbb{F}_{p^r}$ over $\mathbb{F}_p$ (namely $\mathrm{Tr}(x) = x + x^p + \cdots + x^{p^{r-1}}$). Therefore,

$$S_d(a) = \sum_{n=1}^{d} e_p \left(\mathrm{Tr} \left(\sum_{j=1}^{r} a_j \delta_j V_x(n) \right) \right) \quad = \quad \sum_{n=1}^{d} e_p(\mathrm{Tr}(\alpha_a V_x(n)))$$

where $\alpha_a = \sum_{j=1}^{r} a_j \delta_j \in \mathbb{F}_{p^r}$.
Let χ be defined by $\chi(z) = e_p(\mathrm{Tr}(z))$. Then χ is a non trivial additive character on $\mathbb{F}_{p^r}$. Since there exists $j \in \{1, \ldots, r\}$ such that $a_j \neq 0$, then $\alpha_a \neq 0$. We have:

$$S_d(a) = \sum_{n=1}^{d} \chi(\alpha_a V_x(n)) \text{ with } \alpha_a \neq 0.$$

We have

$$\begin{aligned} S_d(a) = \sum_{\substack{n=x+1 \\ n \in \mathbb{Z}_t^*}}^{x+d} \chi(\alpha_a g^{1/n}) &= \frac{1}{t} \sum_{n \in \mathbb{Z}_t^*} \chi(\alpha_a g^{1/n}) \times \sum_{c=0}^{t-1} \sum_{\substack{v=x+1 \\ v \in \mathbb{Z}_t^*}}^{x+d} e_t(c(n-v)) \\ &= \frac{1}{t} \sum_{c=0}^{t-1} \left(\sum_{n \in \mathbb{Z}_t^*} \chi\left(\alpha_a g^{1/n}\right) e_t(cn) \right) \times \sum_{\substack{v=x+1 \\ v \in \mathbb{Z}_t^*}}^{x+d} e_t(-cv). \end{aligned}$$

By applying Proposition 2.4.1 and (4.2), we obtain

$$S_d(a) \leq \frac{1}{t} \sum_{c=0}^{t-1} \left| \sum_{\substack{v=x+1 \\ v \in \mathbb{Z}_t^*}}^{x+d} e_t(-cv) \right| \times t^{1-\alpha_{k,\ell}} q^{\beta_{k,\ell}+o(1)} \leq t^{1-\alpha_{k,\ell}} q^{\beta_{k,\ell}+o(1)}.$$

By applying this bound to $D_x(d)$, we have

$$D_x(d) \ll \frac{1}{p} + \frac{t^{1-\alpha_{k,l}} q^{\beta_{k,\ell}+o(1)}}{d} \sum_{0<|a|<p} \frac{1}{r(a)} \ll \frac{1}{p} + \frac{t^{1-\alpha_{k,\ell}} q^{\beta_{k,\ell}+o(1)}}{d} \log^r p \leq \frac{t^{1-\alpha_{k,\ell}} q^{\beta_{k,\ell}+o(1)}}{d}$$

□

With the choice $k = 4$, $l = 8$, $t = q^{1+o(1)}$ and $d = t^{\frac{127}{128}+\epsilon}$, we obtain

$$D_x(d) \leq p^{r(-\epsilon+o(1))} = q^{-\epsilon+o(1)}.$$

4.1.2. Distribution of the Dodis-Yampolskiy Pseudo-Random Function over Elliptic Curves

Let $E : y^2 = x^3 + Ax + B$, be an elliptic curve over $\mathbb{F}_p$. For $P \in E(\mathbb{F}_p)$ of prime order t, for $x \in \mathbb{Z}_t$, and for $1 \leq d \leq t$ we denote by $D_x(d)$ the discrepancy of the points $(X(V_x(n))/p)$ for $n \in \{1, \ldots, d\}$ where $V_x(n) = \left[\frac{1}{x+n}\right] P \in E(\mathbb{F}_p)$. We obtain the following theorem.

Theorem 4.1.3. *For any $x \in \mathbb{Z}_t$, any integers $k \geq 2$, $l \geq 1$ and $1 \leq d \leq t$, we have:*

$$D_x(d) \leq \frac{t^{1-\alpha_{k,\ell}} p^{\beta_{k,\ell}+o(1)}}{d},$$

where $\alpha_{k,\ell} = \frac{1}{2(4k+\ell)} - \frac{1}{4k\ell}$ and $\beta_{k,\ell} = \frac{1}{4(4k+\ell)}$.

Proof. From Lemma 4.1.1, we derive

$$D_x(d) \ll \frac{1}{p} + \frac{1}{d} \sum_{0<|a|<p} \frac{1}{|a|} \left| \sum_{n=1}^{d} e_p\left(aX(W_x(n))\right) \right|,$$

where a is an integer. Set $S_d(a) = \sum_{n=1}^{d} e_p(aX(W_x(n)))$, we have

$$\begin{aligned} S_d(a) &= \sum_{\substack{n=x+1 \\ n \in \mathbb{Z}_t^*}}^{x+d} e_p\left(aX\left(\left[\frac{1}{n}\right] P\right)\right) \\ &= \frac{1}{t} \sum_{n \in \mathbb{Z}_t^*} e_p\left(aX\left(\left[\frac{1}{n}\right] P\right)\right) \times \sum_{c=0}^{t-1} \sum_{\substack{v=x+1 \\ v \in \mathbb{Z}_t^*}}^{x+d} e_t\left(c(n-v)\right) \\ &= \frac{1}{t} \sum_{c=0}^{t-1} \left(\sum_{n \in \mathbb{Z}_t^*} e_p\left(aX\left(\left[\frac{1}{n}\right] P\right)\right) e_t(cn) \right) \times \sum_{\substack{v=x+1 \\ v \in \mathbb{Z}_t^*}}^{x+d} e_t(-cv) \end{aligned}$$

By applying Proposition 2.4.4 and (4.2), we obtain

$$\begin{aligned} S_d(a) &\leq \frac{1}{t} \sum_{c=0}^{t-1} \left| \sum_{\substack{v=x+1 \\ v \in \mathbb{Z}_t^*}}^{x+d} e_t(-cv) \right| \times t^{1-\alpha_{k,\ell}} p^{\beta_{k,\ell}+o(1)} \\ &\leq t^{1-\alpha_{k,\ell}} p^{\beta_{k,\ell}+o(1)} \end{aligned}$$

By applying this bound to $D_x(d)$, we have

$$\begin{aligned} D_x(d) &\ll \frac{1}{p} + t^{1-\alpha_{k,\ell}} p^{\beta_{k,\ell}+o(1)} \times \frac{1}{d} \sum_{0<|a|<p} \frac{1}{|a|} \\ &\ll \frac{1}{p} + t^{1-\alpha_{k,\ell}} p^{\beta_{k,l}+o(1)} \times \frac{1}{d} \log p \\ &\leq t^{1-\alpha_{k,\ell}} p^{\beta_{k,\ell}+o(1)} \times \frac{1}{d} \end{aligned}$$

□

With the choice $k = 4$, $\ell = 16$, $t = p^{1+o(1)}$ and $d = t^{\frac{255}{256}+\epsilon}$, we obtain $D_x(d) \ll p^{-\epsilon+o(1)}$.

Remark 4.1.4. *If $d = t - 1$, with $t = p^{\frac{1}{2}+\epsilon}$, $0 < \epsilon \leq \frac{1}{2}$, then $S_a(d)$ can be estimated easily, and we have $D_x(d) \ll p^{-\epsilon+o(1)}$.*

4.2. Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function over Finite Fields

Let $g \in \mathbb{F}_{p^r}^*$ for some integer $r > 1$, be an element of prime order $t \mid p^r - 1$. In this section, we prove a lower bound on the degree of univariate polynomial interpolation of the Dodis-Yampolskiy pseudo-random function over finite fields. We consider polynomials that interpolate values of the Dodis-Yampolskiy pseudo-random function for a fixed secret key $x \in \mathbb{F}_t^*$. The values considered are evaluation of the function at integers $n \in \{1, \ldots, d\}$ for some integer $1 \leq d \leq t$ and translates of these values by some fixed constants $\lambda \in \mathbb{N}$. This setting is interesting for applications in cryptography [CHL05; CHK+06]. Note that if one value n is larger than d then, the Dodis-Yampolskiy function is not necessarily defined at $n + \lambda$. In the following, we consider simple sets where all translates belong to the function domain but our method can be adapted to other settings.

Theorem 4.2.1. *Let λ be a fixed integer and let $A \subseteq \{1, \ldots, d\}$. For some $x \in \mathbb{F}_t^*$, let $F(X) \in \mathbb{F}_p[X]$ be such that $F(g^{\frac{1}{x+n}}) = g^{\frac{1}{x+n+\lambda}}$ for all $n \in A$. We have*

$$\deg(F) \geq \frac{t-2s}{4} \quad \textit{and} \quad w(F) \geq \left(\frac{t}{4s}\right)^{1/2} \quad \textit{where } \sharp A = t - s.$$

Proof. Let $R = \{(n+x) \bmod t : n \in A\}$. Then $R \subseteq \mathbb{F}_t$ and $\sharp R = t - s$. We have $F(g^{\frac{1}{n}}) = g^{\frac{1}{n+\lambda}}$ for all $n \in R$. Noticing that $\frac{1}{n+\lambda} = \frac{1}{\lambda}(1 - \frac{1}{\frac{\lambda}{n}+1})$, we obtain $F(g^{\frac{u}{\lambda}}) = g^{\frac{1}{\lambda}(1-\frac{1}{u+1})}$ for all $u = \frac{\lambda}{n}$, $n \in R$.

Let $R_0 = \{u = \frac{\lambda}{n} : n \in R \setminus \{0\}\}$ and $T = \{u \in R_0 : 2u + 1 \in R_0\}$. Since $\sharp R_0 = t - s$, we have $\sharp T \geq t - 2s$. Then

$$F\left(g^{\frac{2u+1}{\lambda}}\right) = g^{\frac{1}{\lambda}(1-\frac{1}{2u+2})} = g^{\frac{1}{\lambda}(\frac{1}{2}+\frac{1}{2}(1-\frac{1}{u+1}))} \quad = \quad g^{\frac{1}{2\lambda}} \times g^{\frac{1}{2\lambda}(1-\frac{1}{u+1})}$$

for all $u \in T$. We thus have

$$F^2\left(g^{\frac{2u+1}{\lambda}}\right) = g^{\frac{1}{\lambda}} \times g^{\frac{1}{\lambda}(1-\frac{1}{u+1})} \quad = \quad g^{\frac{1}{\lambda}} \times F(g^{\frac{u}{\lambda}}), \quad \text{for all } u \in T.$$

Let $H(X) = F^2(g^{\frac{1}{\lambda}}X^2) - g^{\frac{1}{\lambda}}F(X)$. The polynomial $H(X)$ is a non-zero polynomial and $\deg(H) \leq 4\deg(F)$. Since $H(X)$ has at least $\sharp T = t - 2s$ zeros, we have $4\deg(F) \geq t - 2s$ and then $\deg(F) \geq \frac{t-2s}{4}$. Moreover, if $\deg(H) \leq t-1$, since the zeros of H are the powers of $g^{\frac{1}{\lambda}}$, then we have by Lemma 2.2.1, $w(H) \geq t/(t-(t-2s))$, and since $w(H) \leq 2(w(F))^2$, it follows that $w(F) \geq (t/4s)^{1/2}$. □

Remark 4.2.2. *Theorem 4.2.1 is non-trivial only when $\sharp A > t/2$. It remains an open question to obtain non-trivial lowers bounds for smaller sets A.*

4.3. Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function over Elliptic Curves

In this section, p is an odd prime number, E is an elliptic curve defined over $\mathbb{F}_p$ and P is a point of the curve $E(\mathbb{F}_p)$ with prime order t. We prove lower bounds on the degree of polynomial interpolation of the Dodis-Yampolskiy pseudo-random function over elliptic curves defined by $V_x(n) = X\left(\left[\frac{1}{x+n}\right]P\right)$ for a secret key $x \in \mathbb{F}_t^*$ and an integer $n \in \{1,\ldots,d\}$, with $1 \leq d \leq t$.

Theorem 4.3.1. *Let $S \subseteq \{1,\ldots,d\}$, $\sharp S = t - s$. We suppose $X(P) \neq 0$. For some $x \in \mathbb{F}_t^*$, let $F(X) \in \mathbb{F}_p[X]$ be such that $\psi_2^2(F(X(P))) \neq 0$ and $F(V_x(n)) = V_x(n+1)$ for all $n \in S$. We have*

$$\deg(F) \geq \frac{t-2s}{176}.$$

Proof. Let $R = \{(n+x) \bmod t \ : \ n \in S\} \subseteq \mathbb{F}_t$. We have $\sharp R = t - s$. Let us denote $x_k = X([k]P)$ and $R_0 = \{\frac{1}{n} : n \in R\}$, then we have $F(x_u) = x_{1-\frac{1}{1+u}}$ for all $u \in R_0$. We consider the set $T = \{u \in R_0 : 2u+1 \in R_0\}$, then $\sharp T \geq t - 2s$. For all $u \in T$, we have:

$$F(x_{2u+1}) = x_{1-\frac{1}{2(u+1)}} = x_{1/2+1/2(1-1/(u+1))} \quad \text{and} \quad F(x_u) = x_{1-1/(u+1)} \tag{4.3}$$

Using division polynomials (see Section 2.3.2), we can write:

$$x_{1+1-\frac{1}{(u+1)}} = \frac{\theta_2(F(x_{2u+1}))}{\psi_2^2(F(x_{2u+1}))} \tag{4.4}$$

Using the elliptic curve addition law, we have

$$x_{1+\alpha} = \frac{a(x_\alpha) - 2y_1y_\alpha}{(x_\alpha - x_1)^2} \text{ where } a(X) = x_1X^2 + (x_1^2 + A)X + Ax_1 + 2B,$$

and for any polynomial G of degree $m \geq 1$, we have

$$G(x_{1+\alpha}) = \frac{u(x_\alpha) - y_\alpha v(x_\alpha)}{(x_\alpha - x_1)^{2m}} \text{ and } lc(u) = G(x_1)$$

with uniquely determined polynomials $u(X)$ and $v(X)$ with $\deg(u) \leq 2m$ ($\deg(u) = 2m$ if $G(x_1) \neq 0$) and $\deg(v) \leq 2m-2$ and where $lc(u)$ is the leading coefficient of the polynomial $u(X)$. Since $F(x_u) = x_{1-\frac{1}{u+1}}$, we can rewrite (4.4) as:

$$\frac{a(F(x_u)) - y_1y_{1-\frac{1}{u+1}}}{(F(x_u) - x_1)^2} = \frac{\theta_2(F(x_{2u+1}))}{\psi_2^2(F(x_{2u+1}))}.$$

Since the point $(x_{1-\frac{1}{u+1}}, y_{1-\frac{1}{u+1}}) \in E(\mathbb{F}_p)$ and $F(x_u) = x_{1-\frac{1}{u+1}}$, the polynomial $y_1^2(F(x_u)^3 + A \cdot F(x_u) + B)\psi_2^4(F(x_{2u+1}))$ is equal to the polynomial $[(F(x_u) - x_1)^2\theta_2(F(x_{2u+1})) - a(F(x_u))\psi_2^2(F(x_{2u+1}))]^2$. We thus obtain

$$y_1^2(F(x_u)^3 + A \cdot F(x_u) + B) \times \frac{p_1(x_{2u}) - y_{2u}p_2(x_{2u})}{(x_{2u} - x_1)^{12d_0}} = Q(x_u, x_{2u}, y_{2u}),$$

where $d_0 = \deg(F)$ and $Q(x_u, x_{2u}, y_{2u})$ denotes a polynomial of the form

$$\left[(F(x_u) - x_1)^2\frac{p_3(x_{2u}) - y_{2u}p_4(x_{2u})}{(x_{2u} - x_1)^{8d_0}} - a(F(x_u))\frac{p_5(x_{2u}) - y_{2u}p_6(x_{2u})}{(x_{2u} - x_1)^{6d_0}}\right]^2$$

such that $\deg(p_1) \leq 6d_0$, $\deg(p_2) \leq 6d_0 - 2$, $\deg(p_3) \leq 4d_0$, $\deg(p_4) \leq 4d_0 - 2$, $\deg(p_5) \leq 3d_0$ and $\deg(p_6) \leq 3d_0 - 2$. We obtain:

$$y_1^2(F(x_u)^3 + AF(x_u) + B)(x_{2u} - x_1)^{4d_0}(p_1(x_{2u}) - y_{2u}p_2(x_{2u})) = P(x_u, x_{2u}, y_{2u}),$$

where $P(x_u, x_{2u}, y_{2u}) = [(F(x_u) - x_1)^2p_3(x_{2u}) - a(F(x_u))(x_{2u} - x_1)^{2d_0}p_5(x_{2u})$
$- y_{2u}((F(x_u) - x_1)^2p_4(x_{2u}) - a(F(x_u))(x_{2u} - x_1)^{2d_0}p_6(x_{2u}))]^2$.

We then proceed as previously by trying to eliminate y_{2u}. We obtain an expression in function of x_u and x_{2u} and we replace x_{2u} by $\frac{\theta_2(x_u)}{\psi_2^2(x_u)}$. We finally obtain a rational function in x_u of the form:

$$\frac{Q(x_u)}{\psi_2^{40d_0}(x_u)} = 0, \text{ where } Q(X) \in \mathbb{F}_p[X] \text{ and } \deg(Q) \leq 88d_0.$$

Claim 4.3.2. *We have*

$$Q(X) \neq 0 \quad \text{if } \psi_2^2(F(x_1)) \neq 0 \text{ and } x_1 \neq 0$$

Proof. We have $\deg(P_5) = 3d_0$ iff $\psi_2^2(F(x_1)) \neq 0$. If $\deg(P_5) = 3d_0$, One can then verify that the leading coefficient of Q is the leading coefficient of the numerator of the rational function obtained from $[(F(x_u) - x_1)^2p_3(x_{2u}) - a(F(x_u))(x_{2u} - x_1)^{2d_0}p_5(x_{2u})]^4$ after replacing x_{2u} by $\frac{\theta_2(x_u)}{\psi_2^2(x_u)}$.

Therefore, if $\deg(P_5) = 3d_0$, then the leading coefficient of Q is $(f^2 \times x_1 \times \psi_2^2(F(x_1)))^4$ which is non zero if $x_1 \neq 0$ since $\deg(P_5) = 3d_0$ iff $\psi_2^2(F(x_1)) \neq 0$, where f is the leading coefficient of F. Then if $\psi_2^2(F(x_1)) \neq 0$ and $x_1 \neq 0$, $Q(X)$ is a non-zero polynomial.

□

If $\psi_2^2(F(x_1)) \neq 0$ and $x_1 \neq 0$, $Q(X)$ is a non-zero polynomial with at least $\sharp T/2$ different zeros. We thus have $88d_0 \geq (t - 2s)/2$ and the claimed result. □

The condition $X(P) \neq 0$ in the statement of Theorem 4.3.1 holds obviously for almost all point P. The lower bound then holds if the group order $\sharp E(\mathbb{F}_p)$ is odd since in this case, the technical condition $\psi_2^2(F(X(P))) \neq 0$ is always satisfied. However, we obtain a weaker lower bound for the polynomial degree which holds for every curve E.

Theorem 4.3.3. *Let $1 \leq d \leq t$ be a fixed integer and let $A \subseteq \{1, \ldots, d\}$, $\sharp A = t - s$. For some $x \in \mathbb{F}_t^*$, let $F(X) \in \mathbb{F}_p[X]$ such that $F(V_x(n)) = V_x(n+1)$ for all $x \in A$. We have* $\deg(F) \geq (t - 3s)^{1/2}/6$.

Proof. Let $R = \{(n + x) \bmod t : n \in A\}$. Then $R \subseteq \mathbb{F}_t$ and $\sharp R = t - s$. The equation $F(V_x(n)) = V_x(n+1)$ then becomes:

$$F\left(X\left(\left[\frac{1}{n}\right]P\right)\right) = X\left(\left[\frac{1}{n+1}\right]P\right),$$

for all $n \in R$. Denoting $x_k = X([k]P) = X([k \bmod t]P)$ and considering the set $T = \{n \in R : n/2, n+1 \in R\}$, we have

$$\begin{aligned} F\left(x_{\frac{2}{n}}\right) = F\left(x_{\frac{1}{n/2}}\right) = x_{\frac{1}{n/2+1}} = x_{\frac{2}{n+2}} &= \frac{\theta_2(x_{\frac{1}{n+2}})}{\psi_2^2(x_{\frac{1}{n+2}})} \\ &= \frac{\theta_2(F(x_{\frac{1}{n+1}}))}{\psi_2^2(F(x_{\frac{1}{n+1}}))} \\ &= \frac{\theta_2(F(F(x_{\frac{1}{n}})))}{\psi_2^2(F(F(x_{\frac{1}{n}})))}, \end{aligned}$$

hence we have

$$F\left(\frac{\theta_2(x_{\frac{1}{n}})}{\psi_2^2(x_{\frac{1}{n}})}\right) = \frac{\theta_2(F(F(x_{\frac{1}{n}})))}{\psi_2^2(F(F(x_{\frac{1}{n}})))}, \text{ for all } n \in T.$$

Finally, we consider the polynomial

$$H(X) = \psi_2^{2d_0}(X)\psi_2^2(F(F(X)))\left(F\left(\frac{\theta_2(X)}{\psi_2^2(X)}\right) - \frac{\theta_2(F(F(X)))}{\psi_2^2(F(F(X)))}\right).$$

The polynomial $H(X)$ has at least $\sharp T/2$ zeros. We have $F(F(X)) \neq X$ and by Lemma 2.3.2, it will imply that there exists $\alpha \in \overline{\mathbb{F}_p}$ such that $\psi_2^2(F(F(\alpha))) = 0$ and $\psi_2^2(\alpha) \neq 0$. Hence, we have $H(\alpha) = -\theta_2(F(F(\alpha)))\psi_2^{2d_0}(\alpha) \neq 0$, since $\theta_2(X)$ and $\psi_2^2(X)$ have no common zeros. Therefore, $H(X)$ is a non-zero polynomial and $\deg(H) \leq 9d_0^2$. Then we get that $9d_0^2 \geq \sharp R/2$ and the result follows.

□

Part II.

Lattice-Based Cryptanalysis of Pseudo-Random Generators and Signatures

Chapter 5.

Preliminaries

In this Chapter, we recall two short descriptions of Coppersmith's methods that we use in the next two Chapters. These methods have been introduced in 1996 by Don Coppersmith for polynomial of one or two variables see [Cop96b; Cop96a]. Because of its importance in cryptanalysis, these methods have been reformulated [How97] and extended for multivariate polynomials [JM06]. They have been used in cryptography to attack many schemes (see [BD00; BM03; HM10] for RSA and its variants, [HM09; BVZ12] for pseudorandom generators). We also recall the analytic combinatorics proposed by [FS09] and used by [BCTV16] to ease the Coppersmith's methods and we conclude the Chapter by presenting some useful examples that enable to understand the analytic combinatorics.

Contents

5.1. Coppersmith's methods 64
5.1.1. **First method** 64
5.1.2. **Second method** 66
5.2. Analytic Combinatorics 67
5.2.1. Introduction 68
5.2.2. Combinatorial Classes, Sizes, and Parameters 68
5.2.3. Counting the Elements: Generating Functions 69
5.2.4. Counting the Parameters of the Elements: Bivariate Generating Functions 70
5.2.5. Counting the Parameters of the Elements up to a Certain Size 71
5.2.6. Asymptotic Values: Transfer Theorem 72
5.3. Some useful applications of the technique 72
5.3.1. Counting the Bounds for the Monomials (Useful Examples) 72
5.3.2. Counting the Bounds for the Polynomials 74

5.1. Coppersmith's methods

In this section, we give two short descriptions of Coppersmith's methods for solving a multivariate modular polynomial system of equations modulo an integer N. We refer the reader to [JM06] for details and proofs.

5.1.1. First method

5.1.1.1. Problem definition.

Let $f_1(y_1, \ldots, y_n), \ldots, f_s(y_1, \ldots, y_n)$ be irreducible multivariate polynomials defined over $\mathbb{Z}$, having a root $(x_1, \ldots, x_n)$ modulo a known integer N, namely $f_i(x_1, \ldots, x_n) \equiv 0 \mod N$. We want this root to be *small* in the sense that each of its components is bounded by a known value X_i. We also need to bound the sizes of X_i allowing to recover the desired root in polynomial time.

5.1.1.2. Polynomials collection.

In a first step, one generates a collection $\mathscr{P}$ of polynomials $\{\tilde{f}_1, \ldots, \tilde{f}_r\}$ linearly independent having $(x_1, \ldots, x_n)$ as a root modulo N . Usually, multiples and powers of products of f_i, $i = 1, \ldots, s$ are chosen , namely $\tilde{f}_\ell = y_1^{\alpha_{1,\ell}} \cdots y_n^{\alpha_{n,\ell}} f_1^{k_{1,\ell}} \cdots f_s^{k_{s,\ell}}$ for some integers $\alpha_{1,\ell}, \ldots, \alpha_{n,\ell}, k_{1,\ell}, k_{s,\ell}$. Such polynomials satisfy the relation $\tilde{f}_\ell(x_1, \ldots, x_n) \equiv 0 \mod N^{\sum_{i=1}^{s} k_{i,\ell}}$, i.e., there exists an integer c_i such that $\tilde{f}_i(x_1, \ldots, x_n) = c_i N^{\sum_{j=1}^{s} k_{j,\ell}}$.

5.1.1.3. Matrix construction.

We denote as $\mathscr{M}$ the set of monomials appearing in collection of polynomials $\mathscr{P}$. Then each polynomial $\tilde{f}_i$ can be expressed as a vector with respect to a chosen order on $\mathscr{M}$. We hence construct a matrix $\mathcal{M}$ as follows and we define as $\mathcal{L}$ the lattice generated by its rows:

$$\mathcal{M} = \left(\begin{array}{c|c|c} \begin{array}{c} 1 \\ X_1^{-1} \\ \ddots \\ X_1^{-a_1} \ldots X_n^{-a_n} \end{array} & \begin{array}{ccc} \tilde{f}_1 & \ldots & \tilde{f}_r \\ \downarrow & \downarrow & \downarrow \end{array} & \begin{array}{c} 1 \\ y_1 \\ \vdots \\ y_1^{a_1} \ldots y_n^{a_n} \end{array} \\ \hline 0 & \begin{array}{c} N^{\sum_{i=1}^{s} k_{i,1}} \\ \ddots \\ N^{\sum_{i=1}^{s} k_{i,r}} \end{array} & \end{array}\right)$$

On that figure, every row of the upper part is related to one monomial of $\mathscr{M}$ (we assume in the figure that $\mathscr{M}$ contains 1, y_1, and $y_1^{a_1} \ldots y_n^{a_n}$ among other monomials). The left-hand side contains the bounds on these monomials (e.g., the coefficient $X_1^{-1} X_2^{-2}$ is put in the row related to the monomial $y_1 y_2^2$). The right-hand side is formed by all vectors coming from $\mathscr{P}$.

5.1.1.4. A short vector in a sublattice.

Let us now consider the row vector

$$r_0 = (1, x_1, \ldots, x_1^{a_1} \ldots x_n^{a_n}, -c_1, \ldots, -c_r) \ .$$

By multiplying this vector by the matrix $\mathcal{M}$, one obtains:

$$s_0 = \left(1, \left(\frac{x_1}{X_1}\right), \ldots, \left(\frac{x_1}{X_1}\right)^{a_1} \cdots \left(\frac{x_n}{X_n}\right)^{a_n}, 0, \ldots, 0\right) \quad .$$

$s_0 \in \mathcal{L}$ is sufficient to recover the root we are searching for and its norm is very small since $\|s_0\|_2 \leqslant \sqrt{\sharp\mathscr{M}}$. Thus, the recovery of a small vector in $\mathcal{L}$, will likely lead to the recovery of the desired root $(x_1, \ldots, x_n)$. To this end, we first restrict ourselves in a more appropriated subspace. Indeed, noticing that the last coefficients of s_0 are all null, we know that this vector belongs to a sublattice $\mathcal{L}'$ whose last coordinates are composed by zero coefficients. By doing elementary operations on the rows of $\mathcal{M}$, one can construct that sublattice and its determinant is the same as the one of $\mathcal{L}$.

5.1.1.5. Method conclusion.

To finally compute all the small solutions $(x_1, \ldots, x_n)$ of the original modular polynomial system, we require n independent polynomials having as root $(x_1, \ldots, x_n)$ over the integers. To this end, we combine the two following lemmas, where Lemma 5.1.1 is due to Coppersmith see [Cop96a] and Lemma 5.1.2 is due to Jutla see [Jut98].

Lemma 5.1.1. *Let $(b_1, \ldots, b_\omega)$ be an LLL -reduced basis of a lattice $\mathcal{L}$. If a lattice element s satisfies $\|s\| \leqslant \|b_i^\star\|$ for all $i \in \{k+1, \ldots, \omega\}$, then s lies in the space spanned by $b_1, \ldots, b_k$.*

Lemma 5.1.2. *Let $A = (v_1, \ldots, v_\omega)$ be a basis of a lattice $\mathcal{L}$. If b_{max} denotes the maximal length of the Gram-Schmidt orthogonalized basis $(b_1^\star, \ldots, b_\omega^\star)$ of A, namely $b_{max} = \max_i \|b_i^\star\|$. For an LLL -reduced basis $B = (b_1, \ldots, b_\omega)$ of the lattice $\mathcal{L}$, it holds that:*

$$\|b_i^\star\| \geqslant 2^{-\frac{i-1}{4}} \left(\frac{\det(\mathcal{L})}{b_{max}^{\omega-i}}\right)^{\frac{1}{i}} .$$

In order to obtain n polynomials over the integers having as root $(x_1, \ldots, x_n)$, one computes an LLL-reduced basis of the lattice $\mathcal{L}'$ and computes the Gram-Schmidt's orthogonalized basis $(b_1^\star, \ldots, b_t^\star)$ of the LLL output basis $(b_1, \ldots, b_t)$, with $t = \sharp\mathscr{M}$. s_0 belongs to $\mathcal{L}'$ and if its norm is smaller than those of $b_i^\star$ for $i \in \{t-n+1, \ldots, t\}$, then s_0 is orthogonal to $b_i^\star$ for $i \in \{t-n+1, \ldots, t\}$ by Lemma 5.1.1. Extracting the coefficients appearing in $b_i^\star$ for $i \in \{t-n+1, \ldots, t\}$, one can construct n polynomial $p_1, \ldots, p_n$ defined over $\mathbb{Z}$ such that $\{p_1(x_1, \ldots, x_n) = 0, \ldots, p_n(x_1, \ldots, x_n) = 0\}$. Under the (heuristic) assumption that all created polynomials define an algebraic variety of dimension 0, the previous system can be solved (e.g., using elimination techniques such as resultant computation or Gröebner basis) and the desired root recovered in polynomial time.

By Lemma 5.1.2, putting $t = \sharp\mathscr{M}$ the condition on the bounds X_i that make this method work is:

$$\|s_0\| \leqslant \sqrt{t} \leqslant 2^{-\frac{t-n}{4}} \left(\frac{\det(\mathcal{L})}{b_{\max}^{n-1}}\right)^{\frac{1}{t-n+1}} ,$$

which can be rewritten as:

$$1 \leqslant t^{-\frac{t-n+1}{2}} 2^{-\frac{(t-n)(t-n+1)}{4}} b_{\max}^{1-n} \det(\mathcal{L}),$$

in the next Chapter, for large dimensions lattices we let $b_{\max}^{n-1}$ and $t^{-\frac{t-n+1}{2}}2^{-\frac{(t-n)(t-n+1)}{4}}$ contribute to an error term (see [M R10] for details) to obtain the simplified condition:

$$1 < \det(\mathcal{L})$$

The condition on the bounds X_i that make this method work is then given by the following (simplified) inequation :

$$\prod_{y_1^{k_1}\dots y_n^{k_n}\in\mathfrak{M}} X_1^{k_1}\cdots X_n^{k_n} < N^{\sum_{\ell=1}^{r}\sum_{i=1}^{s}k_{i,\ell}} \ . \tag{5.1}$$

For such techniques, the most complicated part is the choice of the collection of polynomials, what could be a really intricate task when working with multiple polynomials.

Remark 5.1.3. *In practice, for small dimensions lattices, the experimental bounds on X_i could be worse than the expected one due to the fact that we cannot give useful estimates for the value b_{max} in general.*

5.1.2. Second method

5.1.2.1. Problem definition.

Let $f_1(y_1,\dots,y_n),\dots,f_s(y_1,\dots,y_n)$ be irreducible multivariate polynomials defined over $\mathbb{Z}$, having a root $(x_1,\dots,x_n)$ modulo a known integer N namely $f_i(x_1,\dots,x_n)\equiv 0 \mod N$. Our goal is to recover the desired root $(x_1,\dots,x_n)$. This problem is generally intractable but becomes solvable (under some conditions) in polynomial time $\log(p)^{O(1)}$ (for constant n and constant total degree of the input polynomials) if the root $(x_1,\dots,x_n)$ is upper-bounded by some values $(X_1,\dots,X_n)$ that depends on N and the degree of the polynomials $f_1,\dots,f_s$.

5.1.2.2. Polynomials collection.

In a first step, one generates a larger collection $\mathscr{P}$ of polynomials $\{\tilde{f}_1,\dots,\tilde{f}_r\}$ linearly independent having $(x_1,\dots,x_n)$ as a root modulo N^m, for some positive integer m. Usually, the technique consists in taking product of powers of the modulus N, the polynomials f_i for $i\in\{1,\dots,s\}$ and some well-chosen monomials, such as

$$\tilde{f}_\ell = p^{m-\sum_{j=1}^{s}k_{j,\ell}} y_1^{\alpha_{1,\ell}}\cdots y_n^{\alpha_{n,\ell}} f_1^{k_{1,\ell}}\cdots f_s^{k_{s,\ell}}$$

for some positive integers $\alpha_{1,\ell},\dots,\alpha_{n,\ell},k_{1,\ell},k_{s,\ell}$. Such polynomials satisfy the relation $\tilde{f}_\ell(x_1,\dots,x_n)\equiv 0 \mod p^m$.

5.1.2.3. Lattice construction.

In a second step, one denotes as $\mathscr{M}$ the set of monomials appearing in collection of polynomials $\mathscr{P}$, and one writes the polynomials $\tilde{f}_i(y_1X_1,\dots,y_nX_n)$ for $i\in\{1,\dots,r\}$ as a vector $b_i\in(\mathbb{Z})^\omega$, where $\omega=\sharp\mathscr{M}$. One then constructs a lattice $\mathcal{L}$ generated by the vectors $b_1,\dots,b_r$ and computes its reduced basis using the LLL algorithm [LLL82].

Lemma 5.1.4. *Let $\mathcal{L}$ be a lattice of dimension ω. In polynomial time, the LLL algorithm given as input of basis of $\mathcal{L}$ outputs a* reduced *basis of $\mathcal{L}$ formed by vectors v_i, $1\leqslant i\leqslant\omega$ that satisfy:*

$$\|v_1\|\leqslant\|v_2\|\leqslant\cdots\leqslant\|v_2\|\leqslant 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}}\det(L)^{\frac{1}{\omega+1-i}}.$$

5.1.2.4. Generating new polynomials.

In a third step of the method, one combines Lemma 5.1.5 below (from [How97]) and Lemma 5.1.4 to obtain n multivariate polynomials $g_1(y_1, \ldots, y_n), \ldots, g_n(y_1, \ldots, y_n)$ having $(x_1, \ldots, x_n)$ as a root over the integers.

Lemma 5.1.5. *(**Howgrave-Graham** Let $h(y_1, \ldots, y_n)$ be a polynomial over $\mathbb{Z}$ having at most ω monomials. Suppose that:*

1. *$h(x_1, \ldots, x_n) = 0 \bmod W$ for some $|x_1| < X_1, \ldots, |x_n| < X_n$ and,*
2. *$\|h(X_1 y_1, \ldots, X_n y_n)\| \leqslant \frac{W}{\sqrt{\omega}}$. Then $h(x_1, \ldots, x_n) = 0$ holds over the integers.*

The LLL algorithm run on the lattice $\mathcal{L}$ to obtain n reduced vectors v_i, $i \in \{1, \ldots, n\}$ that we see as some polynomials $\tilde{h}_i(y_1 X_1, \ldots, y_n X_n)$, $i \in \{1, \ldots, n\}$. One can see that for $i \in \{1, \ldots, n\}$, $\tilde{h}_i(x_1, \ldots, x_n) = 0 \mod p^m$, since $\tilde{h}_i$ is a linear combination of $\tilde{f}_1, \ldots, \tilde{f}_r$. Then if the following condition holds:

$$2^{\frac{r(r-1)}{4(r+1-n)}} \det(L)^{\frac{1}{r+1-n}} < \frac{p^m}{\sqrt{\omega}},$$

by Lemmas 5.1.4 and 5.1.5, $\tilde{h}_i(x_1, \ldots, x_n) = 0$, $i \in \{1, \ldots, n\}$ holds over the integers and we then obtain n polynomials having $(x_1, \ldots, x_n)$ as a root over the integers.

5.1.2.5. Condition.

In our attacks, the number of polynomials in the first step is equal to the number of monomials that appears in the collection, so $r = \omega = \sharp\mathscr{M}$. In the analysis, we let (as usual in this setting) terms that do not depend on N contribute to an error term ε, and the simplified condition becomes:

$$\det(L) < N^{m(\omega+1-n)}.$$

Under the (heuristic) assumption that all created polynomials in the third step define an algebraic variety of dimension 0, the previous system can be solved (e.g., using elimination techniques such as resultant computation or Gröbner basis) and the desired root recovered in polynomial time [1] $\log(p)^{O(1)}$ (for constant n and constant total degree of the input polynomials). We assume that these polynomials define an algebraic variety of dimension 0 and we justify the validity of our attacks by computer experiments.

5.2. Analytic Combinatorics

We now recall the analytic combinatorics results (see [FS09; BCTV16] for more details) that we need in the next two Chapters.

[1] It is well known that the computational complexity of Gröbner basis algorithm may be exponential or even doubly exponential. In our setting, the number of variables and the total degree of the input polynomials are fixed and the theoretical complexity is polynomial in the field size (and thus in the security parameter).

5.2.1. Introduction

To make things clear, we will explain the method with one multivariate modular polynomial. As explained in the former section, Coppersmith's method requires polynomials which are usually constructed as $f_{\mathbf{k}} = y_1^{k_1} \dots y_n^{k_n} f^{k_\ell}$ (with f being a polynomial of degree e in the variables $y_1, \dots, y_n$). In the following, we thus consider a set of polynomials looking like

$$\mathscr{P} = \{ f_{\mathbf{k}} = y_1^{k_1} \dots y_n^{k_n} f^{k_\ell} \bmod N^{k_\ell} \mid 1 \leqslant k_\ell < t \\ \text{and } \deg(f_{\mathbf{k}}) = k_1 + \dots + k_n + k_\ell e < te \},$$

where the notation modN^{k_ℓ} is only here to recall that the considered solution verifies $f_{\mathbf{k}} \equiv 0 \bmod N^{k_\ell}$ (to make things clearer). We suppose that f is not just a monomial (i.e., is the sum of at least two distinct monomials) and therefore each $\mathbf{k}$ corresponds to a distinct polynomial $f_{\mathbf{k}}$.

The set of monomials appearing in the collection $\mathscr{P}$ will usually look like

$$\mathscr{M} = \{ y_{\mathbf{k}} = y_1^{k_1} \dots y_n^{k_n} \mid 0 \leqslant \deg(y_{\mathbf{k}}) = k_1 + \dots + k_n < te \} \ .$$

By construction, since $(x_1, \dots, x_n)$ is a modular root of the polynomials $f_{\mathbf{k}}$, there exists an integer $c_{\mathbf{k}}$ such that $f_{\mathbf{k}}(x_1, \dots, x_n) = c_{\mathbf{k}} N^{k_\ell}$ (see Section 5.1). Furthermore, this root is *small* in the sense that each of its components is bounded by a known value, namely $|x_1| < X_1, \dots, |x_n| < X_n$. These considerations imply that for the final condition in Coppersmith's method (see Equation (5.1)), one needs to compute the values

$$\psi = \sum_{f_{\mathbf{k}} \in \mathscr{P}} k_\ell \quad \text{and} \quad \forall i \in \{1, \dots, n\}, \quad \alpha_i = \sum_{y_{\mathbf{k}} \in \mathscr{M}} k_i \ .$$

These values correspond to the exponent of N and X_i (for $i \in \{1, \dots, n\}$) in Equation (5.1) respectively.

For the sake of readability for the reader unfamiliar with analytic combinatorics, we first show how to compute the number of polynomials in $\mathscr{P}$ or $\mathscr{M}$ of a certain degree and then how to compute these sums ψ and α_i but only for polynomials in $\mathscr{P}$ or $\mathscr{M}$ of a certain degree. These computations are of no direct use for Coppersmith's method but are a warm-up for the really interesting computation, namely these sums ψ and a_i for polynomials in $\mathscr{P}$ or $\mathscr{M}$ up to a certain degree.

5.2.2. Combinatorial Classes, Sizes, and Parameters

A combinatorial class is a finite or countable set on which a size function is defined, satisfying the following conditions: (i) the size of an element is a non-negative integer and (ii) the number of elements of any given size is finite. Polynomials of a "certain" form and up to a "certain" degree can be considered as a combinatorial class, using a size function usually related to the degree of the polynomial.

In the following, we can consider the set $\mathscr{P}$ as a combinatorial class, with the size function $S_{\mathscr{P}}$ defined as $S_{\mathscr{P}}(f_{\mathbf{k}}) = \deg(f_{\mathbf{k}}) = k_1 + \dots + k_n + k_\ell e$. In order to compute the sum of the k_ℓ as explained in Section 5.2.1, we define another function $\chi_{\mathscr{P}}$, called a *parameter* function, such that $\chi_{\mathscr{P}}(f_{\mathbf{k}}) = k_\ell$. This function will enable us, instead of counting "1" for each polynomial, to count "k_ℓ" for each polynomial, which is exactly what we need (see Section 5.2.4 for the details).

As for the monomials, we will also consider the set $\mathscr{M}$ as a combinatorial class, with the size function $S_{\mathscr{M}}$ defined as $S_{\mathscr{M}}(y_{\mathbf{k}}) = k_1 + \cdots + k_n$. In the case the bounds on the variables are equal ($X_1 = \cdots = X_n = X$), the parameter function corresponding to the exponent α_1 of X_1 in the final condition in Coppersmith's method will be set as $\chi_{\mathscr{M}}(y_{\mathbf{k}}) = k_1 + \cdots + k_n$. Otherwise, one will be able to define other parameter functions in case the bounds are not equal .

5.2.3. Counting the Elements: Generating Functions

The counting sequence of a combinatorial class $\mathcal{A}$ with size function S is the sequence of integers $(A_p)_{p\geqslant 0}$ where $A_p = |\{a \in \mathcal{A}_p \mid S(a) = p\}|$ is the number of objects in class $\mathcal{A}$ that have size p. For instance, if we consider the set $\mathscr{M}$ defined in 5.2.1, we have the equality $M_1 = n$ since there are n monomials in n variables of degree 1.

Definition 5.2.1. *The ordinary generating function (OGF) of a combinatorial class $\mathcal{A}$ is the generating function of the numbers A_p, for $p \geqslant 0$, i.e., the formal[2] power series $A(z) = \sum_{p=0}^{+\infty} A_p z^p$.*

For instance, if we consider the set $\mathscr{M}^{(1)} = \{y_1^{k_1} \mid 1 \leqslant k_1 < t\}$ of the monomials with one variable, then one gets $M_p^{(1)} = 1$ for all $p \in \mathbb{N}$, implying that $M^{(1)}(z) = \sum_{p=0}^{+\infty} z^p = \frac{1}{1-z}$.

In the former example, we constructed the OGF $A(z)$ from the sequence of numbers A_p of objects that have size p. Of course, what we are really interested in is to do it the other way around. We now describe an easy way to construct the OGF, and we will deduce from this function and classical analytic tools the value of A_p for every integer p. We assume the existence of an "atomic" class, comprising a single element of size 1, here a variable, usually denoted as $\mathcal{Z}$. We also need a "neutral" class, comprising a single element of size 0, here 1, usually denoted as ε. Their OGF are $Z(z) = z$ and $E(z) = 1$. We show in Table 5.1 the possible admissible constructions that we will need here, as well as the corresponding generating functions.

Table 5.1.: Combinatorics constructions and their OGF

	Construction	OGF
Atomic class	$\mathcal{Z}$	$Z(z) = z$
Neutral class	ε	$E(z) = 1$
Disjoint union	$\mathcal{A} = \mathcal{B} + \mathcal{C}$ (when $\mathcal{B} \cap \mathcal{C} = \emptyset$)	$A(z) = B(z) + C(z)$
Complement	$\mathcal{A} = \mathcal{B} \setminus \mathcal{C}$ (when $\mathcal{C} \subseteq \mathcal{B}$)	$A(z) = B(z) - C(z)$
Cartesian product	$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	$A(z) = B(z) \cdot C(z)$
Cartesian exponentiation	$\mathcal{A} = \mathcal{B}^k = \mathcal{B} \times \cdots \times \mathcal{B}$	$A(z) = B(z)^k$
Sequence	$\mathcal{A} = \textsc{Seq}(\mathcal{B}) = \varepsilon + \mathcal{B} + \mathcal{B}^2 + \ldots$	$A(z) = \frac{1}{1-B(z)}$

One then recovers the formula $M^{(1)}(z) = \frac{1}{1-z}$ from $Z(z) = z$ and the construction $\textsc{Seq}(\mathcal{Z})$ to describe $\mathscr{M}^{(1)}$. Similarly, if we now consider the set $\mathscr{M}^{(2)} = \{y_{\mathbf{k}} = y_1^{k_1} y_2^{k_2} \mid 0 \leqslant k_1 + k_2 < t\}$ of the monomials with two variables, with the size function $S(y_{\mathbf{k}}) = k_1 + k_2$, then one gets $M^{(2)}(z) = M^{(1)}(z) \cdot M^{(1)}(z) = \frac{1}{(1-z)^2}$ from $\mathscr{M}^{(2)} = \mathscr{M}^{(1)} \times \mathscr{M}^{(1)}$. Finally, since

[2] We stress that it is a "formal" series, i.e., with no need to worry about the convergence.

$\frac{1}{(1-z)^2} = \sum_{p=1}^{+\infty} pz^{p-1}$, one gets, for all $p \geqslant 1$, $(M_2)_p = p+1$, which is exactly the number of monomials with two variables of size p.

When the class contains elements of different sizes (such as variables of degree 1 and polynomials of degree e), the variables are represented by the atomic element $\mathcal{Z}$ and the polynomials by the element $\mathcal{Z}^e$, in order to take into account the degree of the polynomial f. If we consider for instance the set $\mathscr{P}^{(1,2)} = \{f_{\mathbf{k}} = y_1^{k_1} f^{k_\ell} \mid 1 \leqslant k_\ell < t \text{ and } \deg(f_{\mathbf{k}}) = k_1 + 2k_\ell < 2t\}$, with f a polynomial of degree 2, this set is isomorphic to $\textsc{Seq}(\mathcal{Z}) \times \mathcal{Z}^2\textsc{Seq}(\mathcal{Z}^2)$, since $\deg(f) = 2$. This leads to an OGF equals to

$$\frac{1}{1-z}\frac{z^2}{1-z^2} = \sum_{q=0}^{+\infty} qz^q \sum_{r=1}^{+\infty} rz^{2r} = \sum_{p=0}^{+\infty} \sum_{r=1}^{\lfloor p/2 \rfloor} (p-2r)rz^p\,,$$

which gives $P_p^{(1,2)} = \sum_{r=1}^{\lfloor p/2 \rfloor} (p-2r)r$, which is exactly the number of polynomials of degree p contained in the class.

5.2.4. Counting the Parameters of the Elements: Bivariate Generating Functions

As seen in the former section, when one considers a combinatorial class $\mathcal{A}$ of polynomials and computes the corresponding OGF $A(z)$, classical analytic tools enable to recover A_p as the coefficient of z^p in the OGF. As explained in the introduction of this section, however, Coppersmith's method requires a computation a bit more tricky, which involves an additional parameter. For the sake of simplicity, we describe this technique on an example.

For instance, consider our monomial set example $\mathscr{M}^{(2)}$, but now assume that $X_1 \neq X_2$. Our goal is to compute $\sum k_1$, where the sum is taken over all the monomials in $\mathscr{M}^{(2)}$ of size p. We set a parameter function[3] $\chi(y_{\mathbf{k}}) = k_1$ and we do not compute $M_p^{(2)}$ (for $p \geqslant 1$) anymore, but rather

$$\chi_p(\mathscr{M}^{(2)}) = \sum_{y_{\mathbf{k}} \in \mathscr{M}^{(2)} | S(y_{\mathbf{k}}) = p} \chi(y_{\mathbf{k}}) = \sum_{y_{\mathbf{k}} \in \mathscr{M}^{(2)} | S(y_{\mathbf{k}}) = p} k_1$$

where, informally speaking, instead of counting for 1, every monomial counts for the value of its parameter (here the degree k_1 in y_1).

The value $\chi_p(\mathscr{M}^{(2)})$ cannot be obtained by the construction of $\mathscr{M}^{(2)}$ as $\textsc{Seq}(\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z})$ that we used in the former section, since the two atomic elements $\mathcal{Z}$ do not play the same role (the first one is linked with the parameter, whereas the second one is not). The classical solution is simply to "mark" the atomic element useful for the parameter, with a new variable u: With this new parameter function, $\mathscr{M}^{(2)}$ is seen as $\textsc{Seq}(u\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z})$, defining the bivariate ordinary generating function (BGF)[4] $M_2(z,u) = \frac{1}{1-uz}\frac{1}{1-z}$. We remark that when we set $u = 1$, we get the original non-parameterized OGF. Informally speaking, the BGF of a combinatorial class $\mathcal{A}$ with respect to a size function S and a parameter function χ is obtained from the corresponding OGF by replacing each z by $u^k z$ where k is the value

[3] Note that it is possible to count the exponents of both X_1 and X_2 at once using two parameters, but it is usually easier to count them separately, which often boils down to the same computation. See concrete examples in Section 5.3.

[4] In complex cases, the marker u can be put to some exponent k, for instance if the parameter considered has a value equal to k for the atomic element.

of the parameter taken on the atomic element $\mathcal{Z}$. We then obtain $\chi_p(\mathcal{A})$ via the following result:

Theorem 5.2.2. *Assume $\mathcal{A}$ is a combinatorial class with size function S and parameter function χ, and assume $A(z,u)$ is the bivariate ordinary generating function for $\mathcal{A}$ corresponding to this parameter (constructed as explained above). Then, if we define*

$$\chi_p(\mathcal{A}) = \sum_{a\in\mathcal{A}|S(a)=p} \chi(a)$$

the ordinary generating function of the sequence $(\chi_p(\mathcal{A}))_{p\geqslant 0}$ is equal to the value $(\partial A(z,u)/\partial u)_{u=1}$, meaning that we have the equality

$$\left.\frac{\partial A(z,u)}{\partial u}\right|_{u=1} = \sum_{p=0}^{+\infty} \chi_p(\mathcal{A}) z^p \stackrel{def}{=} \chi(A)(z) \ .$$

Coming back to our example, one then gets

$$\chi(M^{(2)})(z) = \sum_{p=0}^{+\infty} \chi_p(\mathscr{M}^{(2)}) z^p = \left.\frac{\partial M^{(2)}(z,u)}{\partial u}\right|_{u=1} = \frac{z}{(1-z)^3} = \sum_{p=1}^{+\infty} \frac{p(p-1)}{2} z^{p-1}.$$

meaning that $\chi_p(\mathscr{M}^{(2)}) = p(p+1)/2$ (remind that it is an equality on formal series). Finally, the sum of the degrees k_1 of the elements of size p is $p(p+1)/2$, which can be checked by enumerating them: $y_2^p, y_1y_2^{p-1}, y_1^2y_2^{p-2}, \ldots, y_1^{p-1}y_2, y_1^p$. It is easy to see that the result is exactly the same for X_2, without any additional computation, by symmetry.

5.2.5. Counting the Parameters of the Elements up to a Certain Size

We described in the former section a technique to compute the sum of the (partial) degrees of elements of size p, but how about computing the same sum for elements of size *up to* p? Using the notations of the former section, we want to compute

$$\chi_{\leqslant p}(\mathcal{A}) = \sum_{a\in\mathcal{A}|S(a)\leqslant p} \chi_p(a) \ .$$

The naive way is to sum up the values $\chi_i(\mathcal{A})$ for all i between 0 and p:

$$\chi_{\leqslant p}(\mathcal{A}) = \sum_{i=0}^{p} \sum_{a\in\mathcal{A}|S(a)=i} \chi_i(a)\,,$$

but an easier way to do so is to artificially force all elements a of size less than or equal to p to be of size exactly p by adding enough times a dummy element y_0 such that $\chi(y_0) = 0$.

In our context of polynomials, the aim of the dummy variable y_0 is to homogenize the polynomial. If we consider again the set $\mathscr{M}^{(2)}$ of monomials of two variables y_1 and y_2, with size function equal to $S(y_\mathbf{k}) = k_1 + k_2$ and parameter function equal to $\chi(y_\mathbf{k}) = k_1$, and if we are interested in the sum of the degrees k_1 of the elements in this set of size *up to* p, we now describe this set as $\textsc{Seq}(u\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z})$, the last part being the class of monomials

in the unique variable y_0. This variable is not marked, since its degree is not counted. One obtains the new bivariate generating function $M^{(2)}(z,u) = \frac{1}{1-uz}\frac{1}{(1-z)^2}$ and

$$\chi_{\leqslant}(M^{(2)})(z) = \sum_{p=0}^{+\infty} \chi_{\leqslant p}(\mathscr{M}^{(2)}) z^p = \left.\frac{\partial M^{(2)}(z,u)}{\partial u}\right|_{u=1} = \frac{z}{(1-z)^4}$$
$$= \sum_{p=2}^{+\infty} \frac{p(p-1)(p-2)}{6} z^{p-2},$$

meaning that $\chi_{\leqslant p}(\mathscr{M}^{(2)}) = p(p+1)(p+2)/6$ (remind that it is an equality on formal series). Finally, the sum of the degrees k_1 of the elements of size up to p (i.e., the exponent of X_1 in Coppersmith's method) is $p(p+1)(p+2)/2$, which can be checked by the computation

$$\sum_{i=0}^{p} \frac{i(i+1)}{2} = \frac{p(p+1)(p+2)}{6} .$$

Again, it is easy to see that the result is exactly the same for X_2, without any additional computation.

5.2.6. Asymptotic Values: Transfer Theorem

Finding the OGF or BGF of the combinatorial classes is usually an easy task, but finding the exact value of the coefficients can be quite painful. Coppersmith's method is usually used in an asymptotic way. Singularity analysis enables us to find the asymptotic value of the coefficients in an simple way, using the technique described in [FS09], Corollary $VI.1$ (sim-transfer), page 392. Adapted to our context, their transfer theorem can be stated as follows:

Theorem 5.2.3 (Transfer Theorem)**.** *Assume $\mathcal{A}$ is a combinatorial class with an ordinary generating function F regular enough such that there exists a value c verifying*

$$F(z) \underset{z\to 1}{\sim} \frac{c}{(1-z)^\alpha}$$

for a non-negative integer α. Then the asymptotic value of the coefficient F_n is

$$F_n \underset{n\to\infty}{\sim} \frac{cn^{\alpha-1}}{(\alpha-1)!} .$$

5.3. Some useful applications of the technique

We now describe how to use the generic tools recalled in the former section to count the exponents of the bounds $X_1, \ldots, X_n$ and of the modulo N.

5.3.1. Counting the Bounds for the Monomials (Useful Examples)

5.3.1.1. First Example.

In this example, we consider

$$\mathscr{M} = \{{y_1}^{i_1} \cdots {y_n}^{i_n} \mid 1 \leqslant i_1 + \cdots + i_n < t\}$$

with the bounds $|y_j| < X_j$ for $1 \leqslant j \leqslant n$. In order to obtain the exponent for the bound X_j, we consider $\mathscr{M}$ as a combinatorial class, with the size function $S({y_1}^{i_1} \dots {y_n}^{i_n}) = i_1 + \cdots + i_n$ and the parameter function $\chi_{X_j}({y_1}^{i_1} \dots {y_n}^{i_n}) = i_j$.

We then describe $\mathscr{M}$ as

$$\prod_{\substack{k=1\\k\neq j}}^{n} \underbrace{\textsc{Seq}(\mathcal{Z})}_{y_k} \times \underbrace{\textsc{Seq}(u\mathcal{Z})}_{y_j} \times \underbrace{\textsc{Seq}(\mathcal{Z})}_{y_0} \setminus \varepsilon$$

(the last $\textsc{Seq}(\mathcal{Z})$ being for the dummy value y_0), which leads to the OGF

$$F(z,u) = \left(\frac{1}{1-z}\right)^n \left(\frac{1}{1-uz}\right) - 1 \ .$$

The next step is to compute the partial derivative in u at $u = 1$:

$$\left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} = \left(\frac{1}{1-z}\right)^n \left(\frac{z}{1-uz}\right)^2 \Bigg|_{u=1} = \frac{z}{(1-z)^{n+2}}$$

and take the equivalent value when $z \to 1$:

$$\left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} \underset{z\to 1}{\sim} \frac{1}{(1-z)^{n+2}} \, ,$$

which finally leads, using Theorem 5.2.3, to $\chi_{X_j,<t}(\mathscr{M}) \sim \frac{(t-1)^{n+1}}{(n+1)!} \sim \frac{t^{n+1}}{(n+1)!}$.

This set of monomials used in Coppersmith's method (first method) thus leads to the bound $\prod_{i=1}^{n} X_i^{\frac{t^{n+1}}{(n+1)!}}$.

Concrete bounds. For $n = 2$, we give in the table below the exponent for the bound X_j for smaller t by computing the Taylor series at 0 of the function $\left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1}$. One can verify that each value in the table equals $\sum_{j=1}^{t-1} \frac{j(j+1)}{2}$ which is the exponent for the bound X_j by direct computations.

t	2	3	4	5	6	7	8	9	10
exponent	1	4	10	20	35	56	84	120	165

In the case where $X_1 = \cdots = X_n = X$, the bound becomes $X^{\frac{nt^{n+1}}{(n+1)!}}$. In this same case, one could consider $\mathscr{M}$ as a combinatorial class, with the size function $S({y_1}^{i_1} \dots {y_n}^{i_n}) = i_1 + \cdots + i_n$ and the parameter function $\chi_{X_j}({y_1}^{i_1} \dots {y_n}^{i_n}) = i_1 + \cdots + i_n$ (since the bound X is the same for all variables) and then obtain the same bound.

5.3.1.2. Second Example.

In this example, we consider

$$\mathscr{M} = \{{y_1}^{i_1} \dots {y_n}^{i_n} \mid (i_1 = 0 \text{ or } 0 \leqslant i_2 \leqslant e) \\ \text{and } 1 \leqslant i_1 + \cdots + i_n < t\}$$

with the bounds $|y_i| < X$ for $1 \leqslant i \leqslant n$. We use the size function $S({y_1}^{i_1} \dots {y_n}^{i_n}) = i_1 + \cdots + i_n$ and the parameter function $\chi({y_1}^{i_1} \dots {y_n}^{i_n}) = i_1 + \cdots + i_n$.

The first step is to split $\mathscr{M}$ into disjoint subsets. In our case, the three disjoint subsets correspond to $(i_1 = 0 \text{ and } 0 \leqslant i_2 \leqslant e)$, $(i_1 \neq 0 \text{ and } 0 \leqslant i_2 \leqslant e)$ and $(i_1 = 0 \text{ and } e+1 \leqslant i_2)$. Taking into account the dummy value y_0, we describe them as

$$(\varepsilon + \mathcal{Z} + \cdots + \mathcal{Z}^e) \times \prod_{i=1}^{n-2} \textsc{Seq}(u\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z})$$

for the first one and

$$(u\mathcal{Z}) \times \textsc{Seq}(u\mathcal{Z}) \times (\varepsilon + \mathcal{Z} + \cdots + \mathcal{Z}^e) \times \prod_{i=1}^{n-2} \textsc{Seq}(u\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z})$$

for the second one and

$$(u\mathcal{Z})^{e+1} \times \textsc{Seq}(u\mathcal{Z}) \times \prod_{i=1}^{n-2} \textsc{Seq}(u\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z})$$

for the third one. This leads to the OGF

$$\begin{aligned} F(z,u) &= \left(\left(1 + \frac{uz}{1-uz}\right)(1 + z + \cdots + z^e) + \frac{(uz)^{e+1}}{1-uz}\right)\left(\frac{1}{1-uz}\right)^{n-2}\frac{1}{1-z} \\ &= \frac{1 + z + \cdots + z^e + (uz)^{e+1}}{(1-uz)^{n-1}}\frac{1}{1-z}, \end{aligned}$$

which gives, after computations,

$$\left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} = \left.\frac{(e+1)(1-uz)u^e z^{e+1} + (n-1)z(1 + z + \cdots + z^e + (uz)^{e+1})}{(1-uz)^n}\frac{1}{1-z}\right|_{u=1},$$

and take the equivalent value when $z \to 1$:

$$\left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} \underset{z\to 1}{\sim} \frac{(e+2)(n-1)}{(1-z)^{n+1}},$$

which finally leads, using Theorem 5.2.3, to $\chi_{<t}(\mathscr{M}) \sim \frac{(e+2)(n-1)t^n}{n!}$. This set of monomials used in Coppersmith's method (first method) thus leads to the bound $X^{\frac{(e+2)(n-1)t^n}{n!}}$

5.3.2. Counting the Bounds for the Polynomials

5.3.2.1. First Example.

We now consider the set

$$\begin{aligned} \mathscr{P} = \{f_{\mathbf{k}} = y_1^{k_1} \dots y_n^{k_n} f^{k_\ell} \bmod N^{k_\ell} \mid\ & 1 \leqslant k_\ell < t \\ & \text{and } \deg(f_{\mathbf{k}}) = k_1 + \cdots + k_n + k_\ell e < te\} \end{aligned}$$

with the bounds $X_1 = \cdots = X_n = X$ for the variables. In order to obtain the exponent for the modulus N, we consider the size function $S({y_1}^{k_1} \dots {y_n}^{k_n} f^{k_\ell}) = k_1 + \cdots + k_n + k_\ell$ and the parameter function $\chi_N({y_1}^{k_1} \dots {y_n}^{k_n} f^{k_\ell}) = k_\ell$.

For the sake of simplicity, we can consider $0 \leqslant k_\ell < t$ since the parameter function is equal to 0 on the elements $f_{\mathbf{k}}$ such that $k_\ell = 0$. We describe $\mathscr{P}$ as $\prod_{i=1}^{n} \textsc{Seq}(\mathcal{Z}) \times \textsc{Seq}(u\mathcal{Z}^e) \times \textsc{Seq}(\mathcal{Z})$ (the last one being for the dummy value y_0), since only f needs a marker and its degree is e. This leads to the OGF

$$F(z,u) = \left(\frac{1}{1-z}\right)^{n+1} \frac{1}{1-uz^e} \ .$$

The next step is to compute the partial derivative in u at $u = 1$:

$$\left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} = \left.\frac{z^e}{(1-uz^e)^2}\left(\frac{1}{1-z}\right)^{n+1}\right|_{u=1} = \frac{z^e}{(1-z^e)^2}\left(\frac{1}{1-z}\right)^{n+1}$$

and take the equivalent value when $z \to 1$, using the formula $1 - z^e \sim e(1-z)$:

$$\left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} \underset{z\to 1}{\sim} \frac{1}{e^2(1-z)^{n+3}} \, ,$$

which finally leads, using Theorem 5.2.3, to $\chi_{N,<te}(\mathscr{P}) \sim \frac{(te)^{n+2}}{e^2(n+2)!}$.

5.3.2.2. Second Example.

Let $f_1(y_1, \ldots, y_n), \ldots, f_s(y_1, \ldots, y_n)$ be multivariate polynomials defined over $\mathbb{Z}$ of degree $e_1, \ldots, e_s$ respectively, we now consider the set

$$\mathscr{P} = \{\tilde{f}_\ell = y_1^{\alpha_{1,\ell}} \ldots y_n^{\alpha_{n,\ell}} f_1^{k_{1,\ell}} \cdots f_s^{k_{s,\ell}} \bmod N^{\sum_{j=1}^{s} k_{j,\ell}} \mid 1 \leqslant \sum_{j=1}^{s} k_{j,\ell} \text{ and } \deg(\tilde{f}_\ell) = \sum_{j=1}^{n} \alpha_{j,\ell} + \sum_{j=1}^{s} e_j k_{j,\ell} < te\}$$

where $e = \max(e_1, \ldots, e_s)$. In order to obtain the exponent for the modulus N at the end of Coppersmith's method, we consider the size function $S(\tilde{f}_\ell) = \sum_{j=1}^{n} \alpha_{j,\ell} + \sum_{j=1}^{s} e_j k_{j,\ell}$ and the parameter function $\chi_{f_j}(\tilde{f}_\ell) = k_{j,\ell}$, for $j \in \{1, \ldots, s\}$.

For the sake of simplicity, we can consider $0 \leqslant \sum_{j=1}^{s} k_{j,\ell}$. Since the degree of each f_j is e_j, we describe $\mathscr{P}$ as

$$\prod_{k=1}^{n} \underbrace{\textsc{Seq}(\mathcal{Z})}_{y_k} \times \prod_{\substack{k=1 \\ k\neq j}}^{s} \underbrace{\textsc{Seq}(\mathcal{Z}^{e_k})}_{f_k} \times \underbrace{\textsc{Seq}(u\mathcal{Z}^{e_j})}_{f_j} \times \underbrace{\textsc{Seq}(\mathcal{Z})}_{\text{dummy var.}}$$

which leads to the following generating function

$$F_j(u,z) = \left(\frac{1}{1-z}\right)^{n} \left(\prod_{\substack{k=0 \\ k\neq j}}^{s} \frac{1}{1-z^{e_k}}\right) \frac{1}{1-uz^{e_j}} \frac{1}{1-z} \ .$$

The next step is to compute the partial derivative in u at $u = 1$:

$$\left.\frac{\partial F_j}{\partial u}(u,z)\right|_{u=1} = \left(\frac{1}{1-z}\right)^{n+1} \times \left(\prod_{\substack{k=0 \\ k\neq j}}^{s} \frac{1}{1-z^{e_k}}\right) \times \frac{z^{e_j}}{(1-z^{e_j})^2} \ .$$

We take the equivalent when $z \to 1$, using the formula $1 - z^m \sim m(1-z)$:

$$\left.\frac{\partial F_j}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{1}{e_j(\prod_{k=1}^{s} e_k)(1-z)^{n+s+2}},$$

which finally leads, using Theorem 5.2.3, to

$$\chi_{f_j,<te}(\mathscr{P}) \sim \frac{(te)^{n+s+1}}{e_j(\prod_{k=1}^{s} e_k)(n+s+1)!}.$$

This set of polynomials used in Coppersmith's method thus leads to the bound

$$N^{\frac{(te)^{n+s+1}}{(n+s+1)!\prod_{k=1}^{s} e_k}\sum_{j=1}^{s}\frac{1}{e_j}}.$$

Chapter 6.

Inferring a Linear Congruential Generator and a Power Generator on Elliptic Curves

In this Chapter, we analyze the security of the Elliptic Curve Linear Congruential Generator (EC-LCG) and of the Elliptic Curve Power Generator (EC-PG). We use the Coppersmith's methods to show that these generators are insecure if sufficiently many bits are output at each iteration. Gutierrez and Ibeas showed that the EC-LCG is insecure given a certain amount of most significant bits of some consecutive values of the sequence. Using the Coppersmith's methods, we are able to improve the security bounds of this generator in their setting and in other settings. We also show that the EC-PG is insecure if sufficiently many bits are output at each iteration using the same techniques. The Chapter is organized as follows, in the first section we present the Linear Congruential Generator and Power Generator on Elliptic Curves. In the second and third sections, we infer the EC-LCG when the composer is known or unknown and we conclude the Chapter by inferring the EC-PG.

Contents

6.1. Linear Congruential Generator and Power Generator on Elliptic Curves 78
6.2. Predicting EC-LCG Sequences for Known Composer 78
6.3. Predicting EC-LCG Sequences for Unknown Composer 90
6.3.1. Complexity of the attack 92
6.4. Predicting the Elliptic curve power generator 94

6.1. Linear Congruential Generator and Power Generator on Elliptic Curves

In cryptography, a pseudo-random number generator is a deterministic algorithm which takes as input a short random seed and output and output a long pseudo-random sequence. Random numbers have found a number of applications in the literature. For instance they are useful for privacy, randomized algorithms and key generation. In 1994, S. Hallgren see [S H94] proposed a pseudo-random numbers generator based on a subgroup of points of an elliptic curve defined over a prime finite field. This generator is known as the Linear Congruential Generator on Elliptic Curves (EC-LCG). If E is an elliptic curve defined over a prime finite field $\mathbb{F}_p$, for a given point $G \in E(\mathbb{F}_p)$, the Linear Congruential Generator on Elliptic Curves, EC-LCG is a sequence (U_n) of pseudo-random numbers defined by the relation:

$$U_n = U_{n-1} \oplus G = nG \oplus U_0, \quad n \in \mathbb{N}$$

where $U_0 \in E(\mathbb{F}_p)$ is the initial value or seed. We refer to G as the composer of the generator. For a positive integer $e > 1$ and a point $G \in E(\mathbb{F}_p)$ of order ℓ with $gcd(e; \ell) = 1$, the Elliptic curve power generator, EC-PG is a sequence (V_n) of pseudo-random numbers defined by the relation:

$$V_n = eV_{n-1} = e^n G \quad n \in \mathbb{N}$$

where $V_0 \in E(\mathbb{F}_p)$ is the initial value or seed. The EC-LCG and the EC-PG provide a very attractive alternative to linear and non-linear congruential generators and they have been extensively studied in the literature, see [Shp08; HS05; GL01; GBS99; MS02; PJ02; Shp09b] and the references therein. In Cryptography, we want to use the output of the generator as a stream cipher. One can notice that if two consecutive values U_n, U_{n+1} of the generator are revealed, it is easy to find U_0 and G. So, we output only the most significant bits of each coordinate of U_n, $n = 0, 1, \ldots$ in the hope that this makes the resulting output sequence difficult to predict. Likewise we output only the most significant bits of each coordinate of V_n, $n = 0, 1, \ldots$. We show that the EC-LCG and the EC-PG are insecure if sufficiently many bits are output at each stage. Therefore a secure use of these generators requires to output much fewer bits at each iteration and the efficiency of the schemes is thus degraded. Our attacks used the well-known Coppersmith's methods for finding small roots on polynomial equations. These methods have been used to infer many pseudorandom generators and to cryptanalyze many schemes in Cryptography (see [BCTV16; BVZ12] and the references therein). Throughout this chapter, $\Delta < p^\delta$, with $0 < \delta < 1$, corresponds to the situation where a proportion of at most δ of the least significant bits of the output sequence remain hidden.

6.2. Predicting EC-LCG Sequences for Known Composer

In the cryptographic setting, the initial value $U_0 = (x_0, y_0)$ and the constants G, a and b are supposed to be the secret key. In the following we infer the EC-LCG in the case where the composer G is known and the curve is kept secret. We consider two cases: the case where the most significant bits of consecutive values U_n of the sequence is output and the case where the most significant bits of the abscissa of consecutive multiple values U_{kn} (for some fixed integer k) of the sequence is output. In the first case, we show that the generator is insecure

if at least a proportion of 4/5 of the most significant bits of two consecutive values U_0 and U_1 of the sequence is output. In the second case, We show that the generator is insecure if at least a proportion of 7/8 of the most significant bits of two values $X(U_0)$ and $X(U_k)$ is output, $X(P)$ denoting the abscissa of the point P.

Theorem 6.2.1. ***(two consecutive outputs)*** *Given Δ-approximations W_0, W_1 to two consecutive affine value U_0, U_1 produced by the EC-LCG, and given the value of the composer $G = (x_G, y_G)$. Under the heuristic assumption that all created polynomials we get by applying Coppersmith's method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 in heuristic polynomial time in $\log p$ as soon as $\Delta < p^{1/5}$.*

Proof. We suppose $U_0 \notin \{-G, G\}$. Then, clearing denominators in 2.1, we can translate

$$U_1 = U_0 \oplus G$$

into the following identities in the field $\mathbb{F}_p$:

$$L_1 = L_1(x_0, y_0, x_1) = 0 \, mod \, p, \quad L_2 = L_2(x_0, y_0, x_1, y_1) = 0 \, mod \, p$$

where $U_0 = (x_0, y_0)$, $U_1 = (x_1, y_1)$ and

$$L_1 = x_G^3 + x_1 x_G^2 - x_0 x_G^2 - 2x_1 x_G x_0 - x_G x_0^2 + x_0^3 + 2y_G y_0 + x_1 x_0^2 - y_G^2 - y_0^2,$$

$$L_2 = y_1 x_G - y_1 x_0 - y_G x_0 + y_G x_1 - y_0 x_1 + y_0 x_G.$$

Set $W_0 = (\alpha_0, \beta_0)$ and $W_1 = (\alpha_1, \beta_1)$. Then using the equalities $x_j = \alpha_j + e_j$ and $y_j = \beta_j + f_j$, for $j = 0, 1$, where $|e_j|, |f_j| < \Delta$ leads to the following polynomial system:

$$\begin{cases} f(e_0, e_1, f_0) = 0 \bmod p \\ g(e_0, e_1, f_0, f_1) = 0 \bmod p \ . \end{cases}$$

where $f(z_1, z_2, z_3) = A_1 z_1 + A_2 z_2 + A_3 z_3 + A_4 z_1^2 + A_5 z_1 z_2 + z_1^3 + z_1^2 z_2 - z_3^2 + A_6$ and $g(z_1, z_2, z_3, z_4) = B_1 z_1 + B_2 z_2 + B_3 z_3 + B_4 z_4 + z_1 z_4 + z_2 z_3 + B_5$ are polynomials whose coefficients A_i and B_i are functions of x_G, and the approximations values $\alpha_0, \alpha_1, \beta_0, \beta_1$. If we fix $u = z_1^3 + z_1^2 z_2 - z_3^2$ and $v = z_1 z_4 + z_2 z_3$, then the polynomial f becomes $f_1(z_1, z_2, z_3, u) = A_1 z_1 + A_2 z_2 + A_3 z_3 + A_4 z_1^2 + A_5 z_1 z_2 + u + A_6$ and g becomes $g_1(z_1, z_2, z_3, z_4, v) = B_1 z_1 + B_2 z_2 + B_3 z_3 + B_4 z_4 + v + B_5$.

Description of the attack The adversary is therefore looking for the small solutions of the following modular multivariate polynomial system:

$$\begin{cases} f_1(z_1, z_2, z_3, u) = 0 \bmod p \\ g_1(z_1, z_2, z_3, z_4, v) = 0 \bmod p \ . \end{cases}$$

With $|z_j| < \Delta$, $|u| < X = \Delta^3$ and $|v| < Y = \Delta^2$. The attack consists in applying Coppersmith's methods for multivariate polynomials with one modulo. From now, we use the following collection of polynomials (parameterized by some integer $t \in \mathbb{N}$):

$$\mathfrak{P} = \left\{ z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2} \bmod p^{i_1+i_2} \, : \, i_1 + i_2 > 0 \, \text{and} \, j_1 + \dots + j_4 + 2i_1 + i_2 < 2t \right\}$$

The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ z_1^{i_1} z_2^{i_2} z_3^{i_3} z_4^{i_4} u^{i_5} v^{i_6} \bmod \Delta^{i_1+i_2+i_3+i_4} X^{i_5} Y^{i_6} : i_1 + \cdots + i_4 + 2i_5 + i_6 < 2t \right\}.$$

If we use for instance the monomial order lex (with $z_i < u < v$) on the set of monomials, then the leading monomial of f_1 is $LM(f_1) = u$ and $LM(g_1) = v$. Then the polynomials in $\mathfrak{P}$ are linearly independent since we have prohibited the multiplication by u and v.

Bounds for the polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2}) = j_1 + \cdots + j_4 + 2i_1 + i_2$ and the parameter function $\chi(z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2}) = i_1 + i_2$. The degree of each variable z_i, u, v is 1, whereas the degree of f_1 is 2 and the degree of g_1 is 1. For the sake of simplicity, we can consider $0 \leqslant i_1 + i_2$, since the parameter function equals 0 for elements $z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2}$ with $i_1 + i_2 = 0$.
We can described $\mathfrak{P}$ as:

$$\prod_{i=1}^{4} \textsc{Seq}(Z) \times \textsc{Seq}(uZ^2) \times \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dummy value z_0.
This leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-z}\right)^5 \times \frac{1}{1-uz^2} \times \frac{1}{1-uz}.$$

We have

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} = \frac{z^2(1-z) + z(1-z^2)}{(1-z)^7(1-z^2)^2},$$

as $z \to 1$, $1 - z^n \sim n(1-z)$ leads to:

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} \underset{z\to 1}{\sim} \frac{3(1-z)}{4(1-z)^9} \sim \frac{3}{4(1-z)^8},$$

since $2t \sim 2t - 1$, this leads to:

$$\chi_{<2t}(\mathfrak{P}) \sim \frac{3}{4} \times \frac{(2t)^7}{7!}$$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{i_1} \dots z_4^{i_4} u^{i_5} v^{i_6}) = i_1 + \cdots + i_4 + 2i_5 + i_6$ and the parameter function $\chi(z_1^{i_1} \dots z_4^{i_4} u^{i_5} v^{i_6}) = i_1 + \cdots + i_4$. As z_1, z_2, z_3, z_4, u, v "count for" $1, 1, 1, 1, 2$ and 1 respectively in the condition of the set, we can described $\mathfrak{M}$ as:

$$\textsc{Seq}(Z^2) \times \textsc{Seq}(Z) \times \prod_{i=1}^{4} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dummy value z_0.
Which leads to the generating function:

$$F(z,u) = \frac{1}{(1-z^2)(1-z)^2} \times \left(\frac{1}{1-uz}\right)^4.$$

We have

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} = \frac{4z}{(1-z)^7(1-z^2)},$$

as $z \to 1$, $1-z^n \sim n(1-z)$ leads to:

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} \underset{z\to 1}{\sim} \frac{2}{(1-z)^8},$$

since $2t \sim 2t-1$, this leads to:

$$\chi_{<2t,\Delta}(\mathfrak{M}) \sim \frac{2(2t)^7}{7!}$$

Bounds for the monomials modulo X. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{i_1}\dots z_4^{i_4}u^{i_5}v^{i_6}) = i_1+\dots+i_4+2i_5+i_6$ and the parameter function $\chi(z_1^{i_1}\dots z_4^{i_4}u^{i_5}v^{i_6}) = i_5$. As z_1,z_2,z_3,z_4,u,v "count for" $1,1,1,1,2$ and 1 respectively in the condition of the set, we can described $\mathfrak{M}$ as:

$$\prod_{i=1}^{5} \textsc{Seq}(Z) \times \textsc{Seq}(uZ^2) \times \textsc{Seq}(Z),$$

where the last one is for the dummy value z_0.
Which leads to the generating function:

$$F(z,u) = \frac{1}{(1-z)^6} \times \left(\frac{1}{1-uz^2}\right).$$

This leads to:

$$\chi_{<2t,X}(\mathfrak{M}) \sim \frac{(2t)^7}{4\times 7!}$$

Bounds for the monomials modulo Y. We consider again the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{i_1}\dots z_4^{i_4}u^{i_5}v^{i_6}) = i_1+\dots+i_4+2i_5+i_6$ and the parameter function $\chi(z_1^{i_1}\dots z_4^{i_4}u^{i_5}v^{i_6}) = i_6$. As z_1,z_2,z_3,z_4,u,v "count for" $1,1,1,1,2$ and 1 respectively in the condition of the set, we can described $\mathfrak{M}$ as:

$$\prod_{i=1}^{4} \textsc{Seq}(Z) \times \textsc{Seq}(Z^2) \times \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dummy value z_0.
Which leads to the generating function:

$$F(z,u) = \frac{1}{(1-z)^5(1-z^2)} \times \left(\frac{1}{1-uz}\right).$$

This leads to:

$$\chi_{<2t,Y}(\mathfrak{M}) \sim \frac{(2t)^7}{2\times 7!}$$

Condition. If we denote by $\nu_1 = \chi_{<2t,\Delta}(\mathfrak{M})$, $\nu_2 = \chi_{<2t,X}(\mathfrak{M})$, $\nu_3 = \chi_{<2t,Y}(\mathfrak{M})$ and $\varepsilon = \chi_{<2t}(\mathfrak{P})$, the condition for Coppersmith's method is $p^{\varepsilon} > \Delta^{\nu_1} X^{\nu_2} Y^{\nu_3}$, ie $\Delta < p^{\frac{\varepsilon}{\nu_1+3\nu_2+2\nu_3}}$, where:

$$\frac{\varepsilon}{\nu_1 + 3\nu_2 + 2\nu_3} \sim \frac{\chi_{<2t}(\mathfrak{P})}{\chi_{<2t,\Delta}(\mathfrak{M}) + 3\chi_{<2t,X}(\mathfrak{M}) + 2\chi_{<2t,Y}(\mathfrak{M})} \sim \frac{1}{5},$$

this leads to the expecting bound:

$$\Delta < p^{\frac{1}{5}}.$$

Complexity of the attack. The dimensions of the matrix used in Coppersmith's methods depend on the cardinalities of the set of polynomials and monomials. To compute the cardinalities of the sets $\mathfrak{P}$ and $\mathfrak{M}$, we make used of the parameters functions $\chi(z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2}) = 1$ and $\chi(z_1^{i_1} \dots z_4^{i_4} u^{i_5} v^{i_6}) = 1$. This leads to the generating functions:

$$F_1(z) = \left(\frac{1}{1-z}\right)^5 \times \left(\frac{1}{1-z^2} \times \frac{1}{1-z} - 1\right)$$

and

$$F_2(z) = \left(\frac{1}{1-z}\right)^5 \times \frac{1}{1-z^2} \times \frac{1}{1-z},$$

for $\mathfrak{P}$ and $\mathfrak{M}$ respectively.

Asymptotic bounds. We have:

$$F_1(z), F_2(z) \underset{z\to 1}{\sim} \frac{1}{2(1-z)^7},$$

which leads when t goes to infinity to the asymptotic bound:

$$\frac{1}{2} \times \frac{(2t)^6}{6!}$$

Concrete bounds. We give in the table below the cardinalities of the sets $\mathfrak{P}$ and $\mathfrak{M}$ for smaller t.

t	1	2	3	4	5	6	7	8	9
number of polynomials	1	27	188	776	2393	6111	13664	27672	51897
number of monomials	6	62	314	1106	3108	7476	16044	31548	57882

□

This bound improves the known bound $\Delta < p^{1/6}$. Next we further improve the previous bound and we show that the generator is insecure if at least a proportion of 8/11 of the most significant bits of an infinite consecutive values U_i of the sequence is output.

Experimental Results.

We have implemented the attack in Sage 7.6 on an elliptic curve over $\mathbb{F}_p$ for a 256-bit prime p on a MacBook Air laptop computer (2,2 GHz Intel Core i7, 4 Gb RAM 1600 MHz DDR3, Mac OSX 10.10.5). Our theoretical bound is $\delta_{\text{theo}} = \frac{1}{5}$ ($\delta < p^{\delta}$) and we denote the experimental bound by δ_{exp}. We consider the family of polynomials $\mathfrak{P}_t$ with $t = 2$ which corresponds to a lattice of dimension 89 and we could not increase the value of t (which allows us to obtain a better theoretical bound) because of the large dimension of the corresponding lattice. After the computations of the LLL reduced basis and Gram-Schmidt orthogonalized basis which takes a few seconds, we obtain:

- a polynomial system over the integers of dimension 0 if $\delta_{\text{exp}} < 0.08$ and we can then recover the seed U_0
- a polynomial system over the integers of dimension 1 if $\delta_{\text{exp}} \leqslant 0.1$. After the computation of the Gröbner basis, we add the two polynomials $u - z_1^3 + z_1^2 z_2 - z_3^2$ and $v - z_1 z_4 + z_2 z_3$ to the polynomials obtained with the computation of the Gröbner basis and we are able from the new system to obtain the seed U_0.

Theorem 6.2.2. ***(more consecutive outputs)***
Given Δ-approximations W_0, W_1,...,W_n (for some integer $n > 1$) to $n+1$ consecutive affine values U_0, U_1,...,U_n produced by the EC-LCG, and given the value of the composer $G = (x_G, y_G)$. Under the heuristic assumption that all created polynomials we get by applying Coppersmith's method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 in polynomial time in $\log p$ as soon as $\Delta < p^{\frac{3n}{11n+4}}$

Proof. Let us assume, for instance that the adversary has access to $n+1$ Δ-approximations W_0, W_1,...,W_n of U_0, U_1,...,U_n produced by the EC-LCG. Then using the equalities $x_j = \alpha_j + e_j$ and $y_j = \beta_j + f_j$, for $j = 0, \ldots, n$, where $|e_j|, |f_j| < \Delta$ and $W_j = (\alpha_j, \beta_j)$ and $U_j = (x_j, y_j)$ leads to the following polynomial system:

$$\begin{cases} f_1'(e_0, e_1, f_0) = 0 \bmod p \\ g_1'(e_0, e_1, f_0, f_1) = 0 \bmod p \\ \vdots \\ f_n'(e_{n-1}, e_n, f_{n-1}) = 0 \bmod p \\ g_n'(e_{n-1}, e_n, f_{n-1}, f_n) = 0 \bmod p \ . \end{cases}$$

Where for $i = 1, \ldots, n$, $f_i'(z_{i-1}, z_i, z_{n+i}) = A_1 z_{i-1} + A_2 z_i + A_3 z_{n+i} + A_4 z_{i-1}^2 + A_5 z_{i-1} z_i + z_{i-1}^3 + z_{i-1}^2 z_i - z_{n+i}^2 + A_6$ and $g_i'(z_{i-1}, z_i, z_{n+i}, z_{n+i+1}) = B_1 z_{i-1} + B_2 z_i + B_3 z_{n+i} + B_4 z_{n+i+1} + z_{i-1} z_{n+i+1} + z_i z_{n+i} + B_5$ are polynomials whose coefficients A_i and B_i are functions of x_G, and the approximations values α_k, β_k, $(k = i-1, i)$. If we fix $u_i = z_{i-1}^3 + z_{i-1}^2 z_i - z_{n+i}^2$ and $v_i = z_{i-1} z_{n+i+1} + z_i z_{n+i}$, then the polynomial f_i' becomes $f_i(z_{i-1}, z_i, z_{n+i}, u_i) = A_1 z_{i-1} + A_2 z_i + A_3 z_{n+i} + A_4 z_{i-1}^2 + A_5 z_{i-1} z_i + u_i + A_6$ and g_i' becomes $g_i(z_{i-1}, z_i, z_{n+i}, z_{n+i+1}, v_i) = B_1 z_{i-1} + B_2 z_i + B_3 z_{n+i} + B_4 z_{n+i+1} + v_i + B_5$. The adversary is then looking for the solutions of the modular multivariate polynomial system:

$$\begin{cases} f_1(z_0, z_1, z_{n+1}, u_1) = 0 \bmod p \\ g_1(z_0, z_1, z_{n+1}, z_{n+2}, v_1) = 0 \bmod p \\ \vdots \\ f_n(z_{n-1}, z_n, z_{2n}, u_n) = 0 \bmod p \\ g_n(z_{n-1}, z_n, z_{2n}, z_{2n+1}, v_n) = 0 \bmod p \ . \end{cases}$$

With $|z_j| < \Delta$, $j = 0, \ldots, 2n+1$, $|u_i| < X = \Delta^3$ and $|v_i| < Y = \Delta^2$, $i = 1, \ldots, n$. We consider the following collection of polynomials:

$$\mathfrak{P} = \left\{ \begin{array}{l} \tilde{f}_{j_0,\ldots,j_{2n+1},i_1,\ldots,i_n,l_1,\ldots,l_n} = z_0^{j_0} \ldots z_{2n+1}^{j_{2n+1}} f_1^{i_1} \ldots f_n^{i_n} g_1^{l_1} \ldots g_n^{l_n} \bmod p^{i_1+l_1\cdots+i_n+l_n} \\ \text{s.t. } \ i_1 + l_1 + \cdots + i_n + l_n > 0 \,\text{and}\ j_0 + \cdots + j_{2n+1} + 2(i_1 + \cdots + i_n) + l_1 + \cdots + l_n < 2t \end{array} \right\}.$$

The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ \begin{array}{l} z_0^{j_0} \dots z_{2n+1}^{j_{2n+1}} u_1^{i_1} v_1^{l_1} \dots u_n^{i_n} v_n^{l_n} \bmod \Delta^{j_0+\dots+j_{2n+1}} X^{i_0+\dots+i_n} Y^{l_0+\dots+l_n} \\ \text{s.t. } j_0 + \dots + j_{2n+1} + 2(i_1 + \dots + i_n) + l_1 + \dots + l_n < 2t \end{array} \right\}.$$

Bounds for the Polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(\tilde{f}_{j_0,\dots,j_{2n+1},i_1,\dots,i_n,l_1,\dots,l_n}) = j_0+\dots+j_{2n+1}+2(i_1+\dots+i_n)+l_1+\dots+l_n$ and the parameter function $\chi(\tilde{f}_{j_0,\dots,j_{2n+1},i_1,\dots,i_n,l_1,\dots,l_n}) = i_1 + l_1 + \dots + i_n + l_n$. We can described $\mathfrak{P}$ as:

$$\prod_{i=0}^{2n+1} \textsc{Seq}(Z) \times \prod_{j=1}^{n} \textsc{Seq}(uZ^2) \times \prod_{k=1}^{n} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dummy variable.
This leads to the generating function:

$$F(z,u) = \frac{1}{(1-z)^{2n+3}} \times \frac{1}{(1-uz^2)^n} \times \frac{1}{(1-uz)^n}.$$

We have

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{3n}{2^{n+1}(1-z)^{4n+4}},$$

since $2t \sim 2t-1$, we get:

$$\chi_{<2t}(\mathfrak{P}) \sim \frac{3n}{2^{n+1}} \times \frac{(2t)^{4n+3}}{(4n+3)!}$$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function
$S(z_0^{j_0} \dots z_{2n+1}^{j_{2n+1}} u_1^{i_1} v_1^{l_1} \dots u_n^{i_n} v_n^{l_n}) = j_0 + \dots + j_{2n+1} + 2(i_1 + \dots + i_n) + l_1 + \dots + l_n$ and the parameter function
$\chi(z_0^{j_0} \dots z_{2n+1}^{j_{2n+1}} u_1^{i_1} v_1^{l_1} \dots u_n^{i_n} v_n^{l_n}) = j_0 + \dots + j_{2n+1}$. We can described $\mathfrak{M}$ as:

$$\prod_{i=1}^{n} \textsc{Seq}(Z^2) \times \prod_{i=1}^{n} \textsc{Seq}(Z) \times \prod_{i=0}^{2n+1} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dummy value y_0.
Which leads to the generating function:

$$F(z,u) = \frac{1}{(1-z^2)^n (1-z)^{n+1}} \times \frac{1}{(1-uz)^{2n+2}}.$$

As $z \to 1$, $1 - z^n \sim n(1-z)$ leads to:

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{2n+2}{2^n(1-z)^{4n+4}},$$

since $2t \sim 2t-1$, this leads to:

$$\chi_{<2t,\Delta}(\mathfrak{M}) \sim \frac{2n+2}{2^n} \times \frac{(2t)^{4n+3}}{(4n+3)!}$$

Bounds for the monomials modulo X. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function
$S(z_0^{j_0}\dots z_{2n+1}^{j_{2n+1}}u_1^{i_1}v_1^{l_1}\dots u_n^{i_n}v_n^{l_n}) = j_0+\dots+j_{2n+1}+2(i_1+\dots+i_n)+l_1+\dots+l_n$ and the parameter function
$\chi(z_0^{j_0}\dots z_{2n+1}^{j_{2n+1}}u_1^{i_1}v_1^{l_1}\dots u_n^{i_n}v_n^{l_n}) = i_1+\dots+i_n$. We can described $\mathfrak{M}$ as:

$$\prod_{i=0}^{2n+1}\textsc{Seq}(Z)\times\prod_{i=1}^{n}\textsc{Seq}(Z)\times\prod_{i=}^{n}\textsc{Seq}(uZ^2)\times\textsc{Seq}(Z),$$

where the last one is for the dummy value y_0.
Which leads to the generating function:

$$F(z,u)=\frac{1}{(1-z)^{3n+3}}\times\frac{1}{(1-uz^2)^n}.$$

This leads to:

$$\chi_{<2t,X}(\mathfrak{M})\sim\frac{n}{2^{n+1}}\times\frac{(2t)^{4n+3}}{(4n+3)!}$$

Bounds for the monomials modulo Y. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function
$S(z_0^{j_0}\dots z_{2n+1}^{j_{2n+1}}u_1^{i_1}v_1^{l_1}\dots u_n^{i_n}v_n^{l_n}) = j_0+\dots+j_{2n+1}+2(i_1+\dots+i_n)+l_1+\dots+l_n$ and the parameter function
$\chi(z_0^{j_0}\dots z_{2n+1}^{j_{2n+1}}u_1^{i_1}v_1^{l_1}\dots u_n^{i_n}v_n^{l_n}) = l_1+\dots+l_n$. We can described $\mathfrak{M}$ as:

$$\prod_{i=0}^{2n+1}\textsc{Seq}(Z)\times\prod_{i=1}^{n}\textsc{Seq}(Z^2)\times\prod_{i=}^{n}\textsc{Seq}(uZ)\times\textsc{Seq}(Z),$$

where the last one is for the dummy value y_0.
Which leads to the generating function:

$$F(z,u)=\frac{1}{(1-z)^{2n+3}(1-z^2)^n}\times\frac{1}{(1-uz)^n}.$$

This leads to:

$$\chi_{<2t,Y}(\mathfrak{M})\sim\frac{n}{2^{n}}\times\frac{(2t)^{4n+3}}{(4n+3)!}$$

Condition. If we denote by $\nu_1=\chi_{<2t,\Delta}(\mathfrak{M})$, $\nu_2=\chi_{<2t,X}(\mathfrak{M})$, $\nu_3=\chi_{<2t,Y}(\mathfrak{M})$ and $\varepsilon=\chi_{<2t}(\mathfrak{P})$, the condition for Coppersmith's method is $p^{\varepsilon}>\Delta^{\nu_1}X^{\nu_2}Y^{\nu_3}$, ie $\Delta<p^{\frac{\varepsilon}{\nu_1+3\nu_2+2\nu_3}}$, where:

$$\frac{\varepsilon}{\nu_1+3\nu_2+2\nu_3}\sim\frac{\chi_{<2t}(\mathfrak{P})}{\chi_{<2t,\Delta}(\mathfrak{M})+3\chi_{<2t,X}(\mathfrak{M})+2\chi_{<2t,Y}(\mathfrak{M})}\sim\frac{3n}{11n+4},$$

this leads to the expecting bound:

$$\Delta<p^{\frac{3n}{11n+4}}\underset{n\to\infty}{\to}\Delta<p^{3/11}.$$

□

To prevent the attacks of [GI07] and the previous attacks on the EC-LCG, one could output only the most significant bits of the abscissa of consecutive multiple values U_{kn} (for some fixed integer k) of the sequence. We consider this setting here and use summation polynomials to infer the EC-LCG. These polynomials were used to solve elliptic curve discrete logarithm problem and we use it below to infer the EC-LCG when the attacks of [GI07] and the previous attacks on the EC-LCG cannot work. In the first time, we show that the generator is insecure if at least a proportion of 7/8 of the most significant bits of two values $X(U_0)$ and $X(U_k)$ is output, $X(P)$ denoting the abscissa of the point P.

Theorem 6.2.3. ***(two outputs)*** *Given Δ-approximations w_0, w_k to two values $X(U_0)$, $X(U_k)$ produced by the EC-LCG. Under the heuristic assumption that all created polynomials we get by applying Coppersmith's methods with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 in polynomial time in $\log p$ as soon as $\Delta < p^{1/8}$.*

Proof. We set $U_0 = (x_0, y_0)$, $U_k = (x_k, y_k)$ and $G = (x_G, y_G)$. We then have the equalities:

$$x_i = w_i + e_i \quad \text{where} \quad |e_i| < \Delta,\ i = 0, k.$$

We have $U_0 \oplus (-U_k) = -kG$, thus $U_0 \oplus (-U_k) \oplus kG = O$. Hence:

$$f_3(x_0, x_k, X(kG)) = 0,$$

where f_3 is the polynomial defined in section 2.3.3. Using the equalities $x_i = w_i + e_i$, $i = 0, k$ we obtain the polynomial equation:

$$f(e_0, e_k) = 0,$$

where $f(y_1, y_2) = f_3(w_0 + y_1, w_0 + y_2, X(kG))$ is a polynomial of degree 4. We consider monomials with respect to a monomial ordering such that $LM(f) = y_1^2 y_2^2$. We consider the following collection of polynomials:

$$\mathfrak{P} = \{\tilde{f}_{j_1,j_2,i} = y_1^{j_1} y_2^{j_2} f^i \bmod p^i : i > 0 \quad \text{and} \quad j_1 + j_2 + 4i < 4t \quad \text{and } (j_1 < 2 \vee j_2 < 2)\},$$

One can check that the polynomials $\tilde{f}_{j_1,j_2,i}$ are linearly independent since $LM(f) \neq y_1^{j_1} y_2^{j_2}$ for each $\tilde{f}_{j_1,j_2,i}$ from $\mathfrak{P}$. The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ z_1^{j_1} z_2^{j_2} \bmod \Delta^{j_1+j_2} : j_1 + j_2 < 4t \right\},$$

Bounds for the Polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(\tilde{f}_{j_1,j_2,i}) = j_1 + j_2 + 4i$ and the parameter function $\chi(\tilde{f}_{j_1,j_2,i}) = i$. Since the degree of each variable z_i is 1 and the degree of f is 4, we can described $\mathfrak{P}$ as:

$$\textsc{Seq}(uZ^4) \times \Big((\varepsilon + Z)(\textsc{Seq}(Z) + Z^2\textsc{Seq}(Z))\Big) \times \textsc{Seq}(Z),$$

where the last one is for the dumming value y_0.
This leads to the generating function:

$$F(z, u) = \frac{(1+z)(1+z^2)}{(1-z)^2} \times \frac{1}{1-uz^4}.$$

We have

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} = \frac{z^4(1+z)(1+z^2)}{(1-z)^2(1-z^4)^2}$$

as $z \to 1$, $1-z^4 \sim 4(1-z)$ leads to:

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} \underset{z\to 1}{\sim} \frac{1}{4(1-z)^4},$$

since $4t \sim 4t-1$, this leads to:

$$\chi_{<4t}(\mathfrak{P}) \sim \frac{1}{4} \times \frac{(4t)^3}{3!}$$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(y_1^{j_1}y_2^{j_2}) = j_1 + j_2$ and the parameter function $\chi(y_1^{j_1}y_2^{j_2}) = j_1 + j_2$. Since the degree of each z_i is 1, we can then described $\mathfrak{M}$ as:

$$\prod_{i=1}^{2} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dumming value y_0.
Which leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-uz}\right)^2 \times \frac{1}{1-z}.$$

We have

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} = \frac{2z}{(1-z)^4},$$

as $z \to 1$, $1-z^n \sim n(1-z)$ leads to:

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} \underset{z\to 1}{\sim} \frac{2}{(1-z)^4},$$

since $4t \sim 4t-1$, this leads to:

$$\chi_{<4t}(\mathfrak{M}) \sim \frac{2(3t)^3}{3!}$$

Condition. If we denote by $\nu = \chi_{<4t}(\mathfrak{P})$, and $\varepsilon = \chi_{<4t}(\mathfrak{M})$, the condition for Coppersmith's method is $p^\nu > \Delta^\varepsilon$, ie $\Delta < p^{\frac{\nu}{\varepsilon}}$, where:

$$\frac{\nu}{\varepsilon} \sim \frac{\chi_{<4t}(\mathfrak{P})}{\chi_{<4t}(\mathfrak{M})} \sim \frac{1}{8},$$

this leads to the expecting bound:

$$\Delta < p^{\frac{1}{8}}.$$

□

Experimental Results.
We have implemented the attack in Sage 7.6 on an elliptic curve over $\mathbb{F}_p$ for a 256-bit prime p on a MacBook Air laptop computer (2,2 GHz Intel Core i7, 4 Gb RAM 1600 MHz DDR3, Mac OSX 10.10.5). Our theoretical bound is $\delta_{\text{theo}} = \frac{1}{8}$ ($\delta < p^{\delta}$) and we denote the experimental bound by δ_{exp}. We consider the family of polynomials $\mathfrak{P}_t$ with $t = 2$ which corresponds to a lattice of dimension 40. After the computations of the LLL reduced basis and Gram-Schmidt orthogonalized basis which takes a few seconds, we obtain a polynomial system over the integers of dimension 0 if $\delta_{\text{exp}} < 0.085$ and we can then recover the seed U_0.

We further improve the previous bound and we show that the EC-LCG is insecure if at least a proportion of 3/4 of the most significant bits of an infinite consecutive multiple values U_{kn} of the sequence is output.

Theorem 6.2.4. *(**more outputs**) Given Δ-approximations w_0, w_k,...,w_{kn} to $n+1$ values $X(U_0)$, $X(U_k)$,...,$X(U_{kn})$ produced by the EC-LCG. Under the heuristic assumption that all created polynomials we get by applying Coppersmith's method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 in polynomial time in $\log p$ as soon as $\Delta < p^{\frac{n}{4(n+1)}}$.*

Proof. We set $U_{kt} = (x_{kt}, y_{kt})$, for $t = 0, \ldots, n$ and $G = (x_G, y_G)$. We then have the equalities:

$$x_i = w_i + e_i \quad \text{where} \quad |e_i| < \Delta,\ i = 0, k, \ldots, nk.$$

We have $U_{kt} - U_{k(t+1)} = -kG$, for $t = 0, \ldots, n-1$. Thus $U_{kt} - U_{k(t+1)} + kG = O$. Hence:

$$f_3(x_{tk}, x_{k(t+1)}, X(kG)) = 0,$$

where f_3 is like in the previous proof the polynomial defined in section 2.3.3. Using the equalities $x_i = w_i + e_i$, $i = 0, k, \ldots, kn$ we obtain the polynomial system:

$$f_j(e_{(j-1)k}, e_{jk}) = 0, \quad j = 1, \ldots, n$$

where $f_j(y_{j-1}, y_j) = f_3(w_{(j-1)k} + y_{j-1}, w_{jk} + y_j, X(kG))$ is a polynomial of degree 4. We consider monomials with respect to a monomial ordering such that $LM(f_k) = y_{j-1}^2 y_j^2$. We consider the following collection of polynomials:

$$\mathfrak{P} = \{\tilde{f}_{j_0,\ldots,j_n,i_k} = y_0^{j_0} \ldots y_n^{j_n} f_k^{i_k} \bmod p^{i_k} : \quad k = 1, \ldots, n;\ (j_{k-1} < 2 \vee j_k < 2) \\ (i_k > 0) \text{ and } (j_0 + \cdots + j_n + 4i_k) < 4t\},$$

One can check that the polynomials $\tilde{f}_{j_0,\ldots,j_n,i_k}$ are linearly independent. The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ z_0^{j_0} \ldots z_n^{j_n} \bmod \Delta^{j_0+\cdots+j_n} : j_0 + \cdots + j_n < 4t \right\},$$

Bounds for the Polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(\tilde{f}_{j_0,\ldots,j_n,i_k}) = j_0 + \cdots + j_n + 4i_k$ and the parameter function $\chi(\tilde{f}_{j_0,\ldots,j_n,i_k}) = i_k$. Since the degree of each variable z_i is 1 and the degree of f is 4, we can described $\mathfrak{P}$ as:

$$\sum_{k=1}^{n} \textsc{Seq}(uZ^4) \times \Big((\varepsilon + Z)(\textsc{Seq}(Z) + Z^2\textsc{Seq}(Z))\Big) \times \prod_{j=0, j\neq k-1, k}^{n} \textsc{Seq}(Z) \times \textsc{Seq}(Z),$$

where the last one is for the dumming value z_0.
This leads to the generating function:

$$F(z,u) = \sum_{k=1}^{n} \frac{(1+z)(1+z^2)}{(1-z)^{n+1}} \times \frac{1}{1-uz^4}.$$

We have

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} = \frac{nz^4(1+z)(1+z^2)}{(1-z)^{n+1}(1-z^4)^2}$$

as $z \to 1$, $1-z^4 \sim 4(1-z)$ leads to:

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} \underset{z\to 1}{\sim} \frac{n}{4(1-z)^{n+3}},$$

since $4t \sim 4t-1$, this leads to:

$$\chi_{<4t}(\mathfrak{P}) \sim \frac{n}{4} \times \frac{(4t)^{n+2}}{(n+2)!}$$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(y_1^{j_1}y_2^{j_2}) = j_0 + \cdots + j_n$ and the parameter function $\chi(y_0^{j_0}\dots y_n^{j_n}) = j_0 + \cdots + j_n$. Since the degree of each z_i is 1, we can then described $\mathfrak{M}$ as:

$$\prod_{i=0}^{n} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dumming value z_0.
Which leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-uz}\right)^{n+1} \times \frac{1}{1-z}.$$

We have

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} = \frac{2z}{(1-z)^{n+3}},$$

as $z \to 1$, $1-z^n \sim n(1-z)$ leads to:

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} \underset{z\to 1}{\sim} \frac{n+1}{(1-z)^{n+3}},$$

since $4t \sim 4t-1$, this leads to:

$$\chi_{<4t}(\mathfrak{M}) \sim \frac{(n+1)(4t)^{n+2}}{(n+2)!}$$

Condition. If we denote by $\nu = \chi_{<4t}(\mathfrak{P})$, and $\varepsilon = \chi_{<4t}(\mathfrak{M})$, the condition for Coppersmith's method is $p^\nu > \Delta^\varepsilon$, ie $\Delta < p^{\frac{\nu}{\varepsilon}}$, where:

$$\frac{\nu}{\varepsilon} \sim \frac{\chi_{<4t}(\mathfrak{P})}{\chi_{<4t}(\mathfrak{M})} \sim \frac{n}{4(n+1)},$$

this leads to the expecting bound:

$$\Delta < p^{\frac{n}{4(n+1)}} \underset{n\to\infty}{\to} \Delta < p^{\frac{1}{4}}.$$

□

6.3. Predicting EC-LCG Sequences for Unknown Composer

In this section, we infer the EC-LCG in the case where the composer G is unknown and the curve is kept secret. In the following, We show that the generator is insecure if at least a proportion of 23/24 of the most significant bits of three consecutive values U_0 and U_1 and U_2 of the sequence is output.

Theorem 6.3.1. *(**three consecutive outputs**) Given Δ-approximations W_0, W_1, W_2 to three consecutive affine values U_0, U_1, U_2 produced by the EC-LCG. Under the heuristic assumption that all created polynomials we get by applying Coppersmith's method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 and the composer G in polynomial time in $\log p$ as soon as $\Delta < p^{1/24}$.*

Proof. We set $U_0 = (x_0, y_0)$, $U_1 = (x_1, y_1)$, $U_2 = (x_2, y_2)$, $W_0 = (\alpha_0, \beta_0)$, $W_1 = (\alpha_1, \beta_1)$ and $W_2 = (\alpha_2, \beta_2)$. We then have the equalities:

$$x_i = \alpha_i + e_i,\ y_j = \beta_j + f_j, \quad \text{where} \quad |e_i|, |f_i| < \Delta,\ i = 0, 1, 2. \tag{6.1}$$

We also have:

$$\begin{cases} y_0^2 = x_0^3 + ax_0 + b \\ y_1^2 = x_1^3 + ax_1 + b \\ y_2^2 = x_2^3 + ax_2 + b \ . \end{cases}$$

Eliminating the curve parameters a, b and assuming that $U_2 \neq \pm U_1$(that is, $x_2 \neq x_1$), we obtain the following equation:

$$y_2^2(x_0 - x_1) + x_2^3(x_1 - x_0) + x_0^3(x_2 - x_1) + y_0^2(x_1 - x_2) + x_1^3(x_0 - x_2) + y_1^2(x_2 - x_0) = 0$$

Using the equalities 6.1, leads to the equation:

$$f(e_0, e_1, e_2, f_0, f_1, f_2) = 0 \bmod p$$

where f is a polynomial of degree 4 whose coefficients are functions of $\alpha_0, \alpha_1, \alpha_2$, β_0, β_2, and β_2.

Description of the attack The adversary is therefore looking for the solutions smaller than Δ of the following modular multivariate polynomial equation:

$$f(z_1, \ldots, z_6) = 0 \bmod p$$

The attack consists in applying Coppersmith's methods as in the former subsection. We consider monomials with respect to a monomial ordering such that $LM(f) = z_1^3 z_2$. From now on, we use the following collection of polynomials:

$$\mathfrak{P} = \{\tilde{f}_{j_1,\ldots,j_6,i} = z_1^{j_1} \ldots z_6^{j_6} f^i \bmod p^i : i > 0 \quad \text{and} \quad j_1 + \cdots + j_6 + 4i < 4t \quad \text{and } (0 \leqslant j_1 < 3 \vee j_2 = 0)\},$$

One can check that the polynomials $\tilde{f}_{j_1,\ldots,j_6,i}$ are linearly independent since $LM(f) \neq z_1^{j_1} \ldots z_6^{j_6}$ for each $\tilde{f}_{j_1,\ldots,j_6,i}$ from $\mathfrak{P}$. The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{z_1^{j_1} \ldots z_6^{j_6} \bmod \Delta^{j_1+\cdots+j_6} : j_1 + \cdots + j_6 < 4t\right\}.$$

Bounds for the Polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(\tilde{f}_{j_1,\ldots,j_6,i}) = j_1 + \cdots + j_6 + 4i$ and the parameter function $\chi(\tilde{f}_{j_1,\ldots,j_6,i}) = i$. Since the degree of each variable z_i is 1 and the degree of f is 4, we can described $\mathfrak{P}$ as:

$$\prod_{i=1}^{4} \textsc{Seq}(Z) \times \textsc{Seq}(uZ^4) \times \left((\underbrace{\varepsilon + Z + Z^2}_{z_1})(\underbrace{\varepsilon + Z\textsc{Seq}(Z)}_{z_2}) + \underbrace{Z^3\textsc{Seq}(Z)}_{z_1} \right) \times \textsc{Seq}(Z),$$

where the last one is for the dummy value z_0.
This leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-z}\right)^5 \times \frac{1}{1-uz^4} \times \left((1+z+z^2)(1+z/(1-z)) + \frac{z^3}{1-z} \right).$$

We have

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} = \frac{1+z+z^2+z^3}{(1-z)^6} \times \frac{z^4}{(1-z^4)^2}$$

as $z \to 1$, $1 - z^4 \sim 4(1-z)$ leads to:

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{1}{4(1-z)^8},$$

since $4t \sim 4t - 1$, this leads to:

$$\chi_{<4t}(\mathfrak{P}) \sim \frac{1}{4} \times \frac{(4t)^7}{7!}$$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{j_1} \ldots z_6^{j_6}) = j_1 + \cdots + j_6$ and the parameter function $\chi(z_1^{j_1} \ldots z_6^{j_6}) = j_1 + \cdots + j_6$. Since the degree of each z_i is 1, we can then described $\mathfrak{M}$ as:

$$\prod_{i=1}^{6} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dummy value z_0.
Which leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-uz}\right)^6 \times \frac{1}{1-z}.$$

We have

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} = \frac{6z}{(1-z)^8},$$

as $z \to 1$, $1 - z^n \sim n(1-z)$ leads to:

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{6}{(1-z)^8},$$

since $4t \sim 4t - 1$, this leads to:

$$\chi_{<4t}(\mathfrak{M}) \sim \frac{6(3t)^7}{7!}$$

Condition. If we denote by $\nu = \chi_{<4t}(\mathfrak{P})$, and $\varepsilon = \chi_{<4t}(\mathfrak{M})$, the condition for Coppersmith's method is $p^\nu > \Delta^\varepsilon$, ie $\Delta < p^{\frac{\nu}{\varepsilon}}$, where:

$$\frac{\nu}{\varepsilon} \sim \frac{\chi_{<4t}(\mathfrak{P})}{\chi_{<4t}(\mathfrak{M})} \sim \frac{1}{24},$$

this leads to the expecting bound:

$$\Delta < p^{\frac{1}{24}}.$$

This bound improves the known bound $\Delta < p^{1/46}$.

6.3.1. Complexity of the attack

To compute the cardinalities of the sets $\mathfrak{P}$ and $\mathfrak{M}$, we make used of the parameters functions $\chi(\tilde{f}_{j_1,\ldots,j_6,i}) = 1$ and $\chi(z_1^{j_1} \ldots z_6^{j_6}) = 1$. This leads to the generating functions:

$$F_1(z) = \left(\frac{1}{1-z}\right)^5 \times \frac{z^4}{1-z^4} \times \frac{1+z+z^2+z^3}{1-z}$$

and

$$F_2(z) = \frac{1}{(1-z)^7},$$

for $\mathfrak{P}$ and $\mathfrak{M}$ respectively.
Asymptotic bounds. We have:

$$F_1(z), F_2(z) \underset{z\to 1}{\sim} \frac{1}{(1-z)^7},$$

which leads when t goes to infinity to the asymptotic bound:

$$\frac{(4t)^6}{6!}$$

Concrete bounds. We give in the table below the cardinalities of the sets $\mathfrak{P}$ and $\mathfrak{M}$ for smaller t.

t	1	2	3	4	5
number of polynomials	0	84	1716	12376	54264
number of monomials	84	1716	12376	54264	177100

□

Next, we further improve the previous bound and we show that the generator is insecure if at least a proportion of 7/8 of the most significant bits of an infinite consecutive values U_i of the sequence is output.

Theorem 6.3.2. ***(more consecutive outputs)***
Given Δ-approximations W_0, W_1,...,W_{n+1} (for some integer $n > 1$) to $n+2$ consecutive affine values U_0, U_1,...,U_{n+1} produced by the EC-LCG. Under the heuristic assumption that all created polynomials we get by applying Coppersmith's method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 and the composer G in polynomial time in $\log p$ as soon as $\Delta < p^{n/4(2n+4)}$.

Proof. Let us assume, for instance that the adversary has access to $n+1$ Δ-approximations W_0, W_1,...,W_{n+1} of U_0, U_1,...,U_{n+1} produced by the EC-LCG. Then using the equalities $x_j = \alpha_j + e_j$ and $y_j = \beta_j + f_j$, for $j = 0, \ldots, n$, where $|e_j|, |f_j| < \Delta$ and $W_j = (\alpha_j, \beta_j)$ and $U_j = (x_j, y_j)$ and the fact that $y_j^2 = x_j^3 + ax_j + b$, $j = 0, \ldots, n+1$ and eliminating the curve parameters from three consecutive points U_j, U_{j+1}, U_{j+2},$j = 0, \ldots, n-1$ leads to the following polynomial system:

$$\begin{cases} f_1(e_0, e_1, e_2, f_0, f_1, f_2) = 0 \bmod p \\ \vdots \\ f_n(e_{n-1}, e_n, e_{n+1}, f_{n-1}, f_n, f_{n+1}) = 0 \bmod p \ . \end{cases}$$

Where f_j are polynomials of degrees 4 and $LM(f_i) = z_{i-1}^3 z_i$. The adversary is then looking for the solutions of the modular multivariate polynomial system:

$$\begin{cases} f_1(z_0, z_1, z_2, z_{n+2}, z_{n+3}, z_{n+4}) = 0 \bmod p \\ \vdots \\ f_n(z_{n-1}, z_n, z_{n+1}, z_{2n+1}, z_{2n+2}, z_{2n+3}) = 0 \bmod p \ . \end{cases}$$

We consider the following collection of polynomials:

$$\mathfrak{P} = \left\{ \begin{array}{l} \tilde{f}_{j_0,\ldots,j_{2n+3},\alpha_i} = z_0^{j_0} \ldots z_{2n+3}^{j_{2n+3}} f_i^{\alpha_i} \bmod p^{\alpha_i} \\ \text{s.t. } i = 1, \ldots, n;\ (j_{i-1} < 3 \vee j_i = 0) \\ (\alpha_i > 0) \text{ and } (j_0 + \cdots + j_{2n+3} + 4\alpha_i) < 4t \end{array} \right\}.$$

The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ z_0^{j_0} \ldots z_{2n+3}^{j_{2n+3}} \bmod \Delta^{j_0+\cdots+j_{2n+3}} : j_0 + \cdots + j_{2n+3} < 4t \right\},$$

Bounds for the Polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(\tilde{f}_{j_0,\ldots,j_{2n+3},\alpha_i}) = j_0 + \cdots + j_{2n+3} + 4\alpha_i$ and the parameter function $\chi(\tilde{f}_{j_0,\ldots,j_{2n+3},\alpha_i}) = \alpha_i$. We can described $\mathfrak{P}$ as:

$$\sum_{i=1}^{n} \prod_{\substack{j=0 \\ j \notin \{i-1,i\}}}^{2n+3} \textsc{Seq}(Z) \times \textsc{Seq}(uZ^4) \times \left((\varepsilon + Z + Z^2)(\varepsilon + Z\textsc{Seq}(Z)) + Z^3\textsc{Seq}(Z) \times \varepsilon\right) \times \textsc{Seq}(Z),$$

where the last one is for the dummy variable.
This leads to the generating function:

$$F(z,u) = \left(\frac{1}{(1-z)^{2n+3}} \times \frac{1}{1-uz^4} \right) \times n \left((1+z+z^2)(1+z/(1-z)) + \frac{z^3}{1-z} \right).$$

We have

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{n}{4(1-z)^{2n+6}},$$

since $4t \sim 4t - 1$, this leads to:

$$\chi_{<4t}(\mathfrak{P}) \sim \frac{n}{4} \times \frac{(4t)^{2n+5}}{(2n+5)!}$$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_0^{j_0} \dots z_{2n+3}^{j_{2n+3}}) = j_0 + \cdots + j_{2n+3}$ and the parameter function $\chi(z_0^{j_0} \dots z_{2n+3}^{j_{2n+3}}) = j_0 + \cdots + j_{2n+3}$. We can described $\mathfrak{M}$ as:

$$\prod_{i=1}^{2n+4} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dummy variable.
This leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-uz}\right)^{2n+4} \times \frac{1}{1-z}.$$

We get

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} \underset{z\to 1}{\sim} \frac{2n+4}{(1-z)^{2n+6}},$$

since $4t \sim 4t-1$, this leads to:

$$\chi_{<4t}(\mathfrak{M}) \sim (2n+4) \times \frac{(4t)^{2n+5}}{(2n+5)!}$$

Condition. If we denote by $\nu = \chi_{<4t}(\mathfrak{P})$, and $\varepsilon = \chi_{<4t}(\mathfrak{M})$, the condition for Coppersmith's method is $p^\nu > \Delta^\varepsilon$, ie $\Delta < p^{\frac{\nu}{\varepsilon}}$, where:

$$\frac{\nu}{\varepsilon} \sim \frac{\chi_{<4t}(\mathfrak{P})}{\chi_{<4t}(\mathfrak{M})} \sim \frac{n}{4(2n+4)},$$

This leads to the expecting bound:

$$\Delta < p^{\frac{n}{4(2n+4)}} \underset{n\to\infty}{\to} \Delta < p^{1/8}.$$

□

6.4. Predicting the Elliptic curve power generator

We infer the EC-PG in the case where the constants a, b and e are known. We show that this generator is insecure if at least a proportion of $1 - \frac{1}{2e^2}$ of the most significant bits of two consecutive values $X(V_0)$ and $X(V_1)$ is output.

Theorem 6.4.1. *(**two consecutive outputs**) Given Δ-approximations w_0, w_1 to two consecutive values $X(V_0)$, $X(V_0)$ produced by the EC-PG and under the heuristic assumption that all created polynomials we get by applying Coppersmith's method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed V_0 in heuristic polynomial time in $\log p$ as soon as $\Delta < p^{\frac{1}{2e^2}}$*

Proof. We put $V_0 = (x_0, y_0)$, $V_1 = (x_1, y_1)$. We have $x_1 = \frac{\theta_e(x_0)}{\psi_e^2(x_0)}$ (where the polynomials $\theta_e(X)$ and $\psi_e^2(X)$ are defined in section 2.3.2) since $V_1 = eV_0$. Using the equalities $x_0 = w_0 + \alpha_0$ and $x_1 = w_1 + \alpha_1$ with $\alpha_i < \Delta$, we have $f(\alpha_1, \alpha_0) = 0$, where $f(y_1, y_2) = (y_1 + w_1)\psi_e^2(y_2 +$

$w_0) - \theta_e(y_2 + w_0)$ s a polynomial of degree e^2. We are looking for small modular modulo p. We consider monomials with respect to a monomial ordering such that the leading monomial of f is $LM(f) = y_1 y_2^{e^2-1}$. We consider the following collection of polynomials (parameterized by some integer $t \in \mathbb{N}$):

$$\mathfrak{P} = \left\{ \begin{array}{l} \widetilde{f}_{j_1,j_2,i} = y_1^{j_1} y_2^{j_2} f^i \bmod p^i : i > 0 \,\text{and}\, j_1 + j_2 + e^2 i < e^2 t \\ \text{and}\, (j_1 = 0 \vee 0 \leqslant j_2 \leqslant e^2 - 2) \end{array} \right\}.$$

One can check that the polynomials $\widetilde{f}_{j_1,j_2,i}$ are linearly independent since $LM(\widetilde{f}_{j_1,j_2,i}) \neq y_1^{j_1} y_2^{j_2}$ for each $\widetilde{f}_{j_1,j_2,i}$. The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ y_1^{j_1} y_2^{j_2} \bmod \Delta^{j_1+j_2} : j_1 + j_2 < e^2 t \right\}.$$

Bounds for the Polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(\tilde{f}_{j_1,j_2,i}) = j_1 + j_2 + e^2 i$ and the parameter function $\chi(\tilde{f}_{j_1,j_2,i}) = i$. Since the degree of each variable z_i is 1 and the degree of f is e^2, we can described $\mathfrak{P}$ as:

$$\textsc{Seq}(uZ^{e^2}) \times \left((\underbrace{\varepsilon + Z + \cdots + Z^{e^2-2}}_{y_2})(\underbrace{\varepsilon + Z\textsc{Seq}(Z)}_{y_1}) + \underbrace{Z^{e^2-1}\textsc{Seq}(Z)}_{y_2} \right) \times \textsc{Seq}(Z),$$

where the last one is for the dumming value y_0.
This leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-z}\right)^2 \times \frac{1}{1-uz^{e^2}} \times \left(1 + z + \cdots + z^{e^2-1}\right).$$

We have

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} = \frac{z^{e^2}(1 + z + \cdots + + z^{e^2-1})}{(1-z)^2(1-z^{e^2})^2}$$

as $z \to 1$, $1 - z^{e^2} \sim e^2(1-z)$ leads to:

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{1}{e^2(1-z)^4},$$

since $e^2 t \sim e^2 t - 1$, this leads to:

$$\chi_{<e^2 t}(\mathfrak{P}) \sim \frac{1}{e^2} \times \frac{(e^2 t)^3}{3!}$$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{j_1} \ldots z_6^{j_6}) = j_1 + \cdots + j_6$ and the parameter function $\chi(y_1^{j_1} y_2^{j_2}) = j_1 + j_2$. Since the degree of each z_i is 1, we can then described $\mathfrak{M}$ as:

$$\prod_{i=1}^{2} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dumming value y_0.
Which leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-uz}\right)^2 \times \frac{1}{1-z}.$$

We have

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} = \frac{2z}{(1-z)^4},$$

which leads to:

$$\frac{\partial F}{\partial u}(u,z)\Big|_{u=1} \underset{z\to 1}{\sim} \frac{2}{(1-z)^4},$$

since $e^2t \sim e^2t - 1$, this leads to:

$$\chi_{<e^2t}(\mathfrak{M}) \sim \frac{2(e^2t)^3}{3!}$$

Condition. If we denote by $\nu = \chi_{\leq e^2t}(\mathfrak{P})$, and $\varepsilon = \chi_{<e^2t}(\mathfrak{M})$, the condition for Coppersmith's method is $p^\nu > \Delta^\varepsilon$, ie $\Delta < p^{\frac{\nu}{\varepsilon}}$, where:

$$\frac{\nu}{\varepsilon} \sim \frac{\chi_{<e^2t}(\mathfrak{P})}{\chi_{<e^2t}(\mathfrak{M})} \sim \frac{1}{2e^2},$$

this leads to the expecting bound:

$$\Delta < p^{\frac{1}{2e^2}}.$$

□

Theorem 6.4.2. *(**more consecutive outputs**) Given Δ-approximations w_0, w_1,...,w_n (for some integer $n > 1$) to $n+1$ consecutive values $X(V_0)$, $X(V_1)$,..., $X(V_1)$ produced by the EC-PG and under the heuristic assumption that all created polynomials we get by applying Coppersmith's method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed V_0 in heuristic polynomial time in $\log p$ as soon as $\Delta < p^{\frac{n}{(n+1)e^2}}$*

Proof. We put $V_i = (x_i, y_i)$, $i = 0, \ldots, n$. We have $x_{j+1} = \frac{\theta_e(x_j)}{\psi_e^2(x_j)}$ since $V_{j+1} = eV_j$ for $j = 0, \ldots, n-1$. Using the equalities $x_i = w_i + \alpha_i$, $i = 0, \ldots, n$ with $\alpha_i < \Delta$, we have $f_j(\alpha_j, \alpha_{j-1}) = 0$, for $j = 1, \ldots, n$ where $f_j(y_{j-1}, y_j) = (y_{j-1}+w_j)\psi_e^2(y_j+w_{j-1}) - \theta_e(y_j+w_{j-1})$. We are then looking for small modular modulo p. We use the Coppersmith's methods to recover the desired solution in polynomial time. The monomials are ordered with respect to a monomial ordering such that the leading monomial of each f_j is $y_{j-1}y_j^{e^2-1}$. f_j is a polynomial of degree e^2. We consider the following collection of polynomials (parameterized by some integer $t \in \mathbb{N}$):

$$\mathfrak{P} = \left\{ \begin{array}{l} \widetilde{f}_{j_1,\ldots,j_n,i_k} = y_1^{j_1} \ldots y_n^{j_n} f_k^{i_k} \bmod p^{i_k} : i_k > 0, \quad k \in \{1, \ldots, n\}, \quad \text{and} \\ j_1 + \cdots + j_n + e^2 i_k < e^2 t \quad \text{and} \quad (j_{k-1} = 0 \vee 0 \leqslant j_k \leqslant e^2 - 2) \end{array} \right\}.$$

One can check that the polynomials $\widetilde{f}_{j_1,\ldots,j_n,i_k}$ are linearly independent since $LM(\widetilde{f}_{j_1,\ldots,j_n,i_k}) \neq y_1^{j_1} \ldots y_n^{j_n}$ for each $\widetilde{f}_{j_1,\ldots,j_n,i_k}$. The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ y_0^{j_0} \ldots y_n^{j_n} \bmod \Delta^{j_0+\cdots+j_n} : j_0 + \cdots + j_n < e^2 t \right\}.$$

Bounds for the Polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(\tilde{f}_{j_1,\ldots,j_n,i_k}) = j_1 + \cdots + j_2 + e^2 i_k$ and the parameter function $\chi(\tilde{f}_{j_1,\ldots,j_n,i_k}) = i_k$. Since the degree of each variable z_i is 1 and the degree of f_k is e^2, we can described $\mathfrak{P}$ as:

$$\sum_{k=1}^{n} \textsc{Seq}(uZ^{e^2}) \times \left((\underbrace{\varepsilon + Z + \cdots + Z^{e^2-2}}_{y_k})(\underbrace{\varepsilon + Z\textsc{Seq}(Z)}_{y_{k-1}}) + \underbrace{Z^{e^2-1}\textsc{Seq}(Z)}_{y_k} \right) \times \prod_{\substack{j=0 \\ j \notin \{k-1,k\}}}^{n} \textsc{Seq}(Z) \times \textsc{Seq}(Z),$$

where the last one is for the dumming value z_0.
This leads to the generating function:

$$F(z,u) = \sum_{k=1}^{n} \left(\frac{1}{1-z}\right)^{n+1} \times \frac{1 + z + \cdots + z^{e^2-1}}{1 - uz^{e^2}}.$$

We have

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} = \frac{nz^{e^2}(1 + z + \cdots + +z^{e^2-1})}{(1-z)^{n+1}(1-z^{e^2})^2}$$

as $z \to 1$, $1 - z^{e^2} \sim e^2(1-z)$ leads to:

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{n}{e^2(1-z)^{n+3}},$$

since $e^2 t \sim e^2 t - 1$, this leads to:

$$\chi_{<e^2 t}(\mathfrak{P}) \sim \frac{n}{e^2} \times \frac{(e^2 t)^{n+2}}{(n+2)!}$$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{j_1} \ldots z_6^{j_6}) = j_1 + \cdots + j_6$ and the parameter function $\chi(y_0^{j_0} \ldots y_n^{j_2}) = j_1 + \cdots + j_n$. Since the degree of each y_i is 1, we can then described $\mathfrak{M}$ as:

$$\prod_{i=0}^{n} \textsc{Seq}(uZ) \times \textsc{Seq}(Z),$$

where the last one is for the dumming value z_0.
Which leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-uz}\right)^{(n+1)} \times \frac{1}{1-z}.$$

We have

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} = \frac{(n+1)z}{(1-z)^{n+3}},$$

which leads to:

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{n+1}{(1-z)^(n+3)},$$

since $e^2 t \sim e^2 t - 1$, this leads to:

$$\chi_{<e^2 t}(\mathfrak{M}) \sim \frac{(n+1)(e^2 t)^(n+2)}{(n+2)!}$$

Condition. If we denote by $\nu = \chi_{<e^2t}(\mathfrak{P})$, and $\varepsilon = \chi_{<e^2t}(\mathfrak{M})$, the condition for Coppersmith's method is $p^\nu > \Delta^\varepsilon$, ie $\Delta < p^{\frac{\nu}{\varepsilon}}$, where:

$$\frac{\nu}{\varepsilon} \sim \frac{\chi_{<e^2t}(\mathfrak{P})}{\chi_{<e^2t}(\mathfrak{M})} \sim \frac{n}{(n+1)e^2},$$

this leads to the expecting bound:

$$\Delta < p^{\frac{n}{(n+1)e^2}} \underset{n\to\infty}{\to} \Delta < p^{\frac{1}{e^2}}.$$

□

Chapter 7.

Lattice Attacks on Pairing-Based Signatures

The present Chapter deals with lattice attacks on some well-known Pairing-Based signatures. Practical implementations of cryptosystems often suffer from critical information leakage through side-channels (such as their power consumption or their electromagnetic emanations). For public-key cryptography on embedded systems, the core operation is usually group exponentiation – or scalar multiplication on elliptic curves – which is a sequence of group operations derived from the private-key that may reveal secret bits to an attacker (on an unprotected implementation). We present lattice-based polynomial-time (heuristic) algorithms that recover the signer's secret in popular pairing-based signatures when used to sign several messages under the assumption that blocks of consecutive bits of the corresponding exponents are known by the attacker. Our techniques relies upon Coppersmith's methods and apply to all signatures in the so-called *exponent-inversion* framework in the standard security model (*i.e.* Boneh-Boyen and Gentry signatures) as well as in the random oracle model (*i.e.* Sakai-Kasahara signatures). The Chapter is organized as follows: we start by recalling the Sakai-Kasahara, Boneh-Boyen and Gentry's Pairing-Based Signatures Schemes. We then present the attack on Gentry's signatures. Next, we present the attack on Boneh-Boyen's signatures and we conclude by the attack on Sakai-Kasahara's signature.

Contents

7.1. Sakai-Kasahara, Boneh-Boyen and Gentry's Pairing-Based Signatures Schemes . . . 101
7.2. Lattice Attack On Gentry Signatures . . . 102
7.2.1. Gentry Signatures . . . 102
7.2.2. Description of the Attack . . . 103
7.2.3. Experimental Results . . . 106
7.3. Concrete Attack Examples against Gentry signatures . . . 107
7.4. Lattice Attack on Boneh-Boyen Signatures . . . 109
7.4.1. Boneh-Boyen Signatures . . . 109
7.4.2. Description of the Attack . . . 109
7.4.3. Experimental results . . . 111
7.5. Lattice Attack on Sakai-Kasahara Signatures . . . 111
7.5.1. Sakai-Kasahara Signatures . . . 111
7.5.2. Description of the Attack . . . 112
7.5.3. Experimental results . . . 113

7.1. Sakai-Kasahara, Boneh-Boyen and Gentry's Pairing-Based Signatures Schemes

An identity-based encryption (IBE) scheme is a public key encryption scheme in which a user public key is its identity which may be an arbitrary string such as an email address, a phone number or any other identifier and the user private key is generated by a trusted authority called the private-key generator. In their seminal paper proposing the first IBE scheme, Boneh and Franklin [BF01] mentioned an interesting transform from an IBE scheme to a signature scheme (whose observation was attributed to Naor). The transformation is as follows: the private-key generator public key and secret key correspond to the public key and secret key of the signature scheme and the user private key generation correspond to signatures generation. The well-known short signature scheme proposed by Boneh, Lynn and Shacham [BLS01; BLS04] can be seen as an application of Naor transformation to Boneh and Franklin IBE [BF01].

Pairings (or bilinear maps) are powerful mathematical constructs which have been used since 2000 to design numerous complex cryptographic protocols. There are three known pairing-based approaches to design identity-based encryption schemes [Boy08]: *full-domain-hash* [BF01], *commutative-blinding* [BB04b] and *exponent-Inversion* [BB04b; BB04a; BB08]. In this Chapter, we focus on the latter framework which gives rise to several short signature schemes thanks to Naor transformation. We consider several pairing-based signature schemes in the exponent-inversion framework. In [SK03], Sakai and Kasahara presented the first such scheme (whose security was analyzed in the random oracle model by Zhang, Safavi-Naini and Susilo in [ZSS04]). Boneh and Boyen [BB04a] then presented the first pairing-based signature whose security can be proven in the standard security model. In 2006, Gentry [Gen06] proposed yet another scheme using the exponent-inversion paradigm, with a tighter security proof than the earlier proposals.

These schemes can be described in a general simplified form as follows. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of the same prime order p and let g be a generator of $\mathbb{G}$. We suppose that $(\mathbb{G}, \mathbb{G}_T)$ are equipped with an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a collision-resistant hash function. Let $f, g \in \mathbb{Z}_p[X, Y, M, R]$ be two polynomials of degree at most one in X and Y. The key generation picks uniformly at random two integers $(x, y) \in \mathbb{Z}_p$ as the signing secret key and outputs $(g^x, g^y) \in \mathbb{G}^2$ as the public-key. To sign a message $m \in \{0,1\}^*$, the signer picks uniformly at random $r \in \mathbb{Z}_p$, computes

$$\sigma = g^{f(x,y,\mathcal{H}(m),r)/g(x,y,\mathcal{H}(m),r)}$$

and outputs the pair (σ, r) as the signature. The validity of a signature is checked by verifying whether the following equality holds:

$$e(\sigma, g^{g(x,y,\mathcal{H}(m),r)} = e(g^{f(x,y,\mathcal{H}(m),r)}, g)$$

where the elements $g^{f(x,y,\mathcal{H}(m),r)}$ and $g^{g(x,y,\mathcal{H}(m),r)}$ can be computed publicly from g^x, g^y, m and r. The three schemes use the following specific polynomials:

- **Sakai-Kasahara**[1] **[SK03]:** $f(X, Y, M, R) = 1$, $g(X, Y, M, R) = X + M$
- **Boneh-Boyen [BB04a]:** $f(X, Y, M, R) = 1$, $g(X, Y, M, R) = X + M + YR$

[1] Sakai-Kasahara scheme actually does not use the secret key y and is deterministic.

- **Gentry [Gen06]:** $f(X, Y, M, R) = Y + R$, $g(X, Y, M, R) = X + M$

We present lattice-based polynomial-time algorithms that recover the signer's secret $(x, y) \in \mathbb{Z}_p^2$ in these pairing-based signatures when used to sign a constant number of messages under the assumption that blocks of consecutive bits of the corresponding exponents $f(x, y, \mathcal{H}(m), r)/g(x, y, \mathcal{H}(m), r) \bmod p$ are known by the attacker. We consider known-message attacks and chosen-message attacks (*i.e.* where the atacker is allowed to choose the message m). The method of this paper is heuristic and uses Coppersmith's lattice technique. Let ℓ denote the bit-length of p and N denote the number of unknown blocks of each signing exponent. In a nutshell, we show that one can recover the secret key if the number of consecutive bits of each unknown block is smaller than the following theoretical values:

- **Sakai-Kasahara:** $\ell/2N^2$
- **Boneh-Boyen:** $\ell/2N^2$
- **Gentry:** ℓ/N

provided that the number of signatures is sufficiently large (see the corresponding sections in the chapter for more precise bounds). It is interesting to note, that Gentry scheme which provides the best classical security (tight security reduction in the standard security model), is the weakest against our class of attacks.

More generally, our lattice-based algorithms can be seen as methods to solve variants of the *modular inversion hidden number problem* which was introduced by Boneh, Halevi and Howgrave-Graham in 2001 [BHH01]. This problem is to find a hidden number given several integers and partial bits of the corresponding modular inverse integers of the sums of the known integers and that unknown integer. It was used in [BHH01] to built a pseudo-random number generator and a message authentication code scheme. In [LSSW12], the authors mentioned that it is interesting to study a general problem of recovering of an unknown rational function. One can see our results as a first step towards solving this problem.

The efficiency of our (heuristic) attacks has been validated experimentally.

7.2. Lattice Attack On Gentry Signatures

7.2.1. Gentry Signatures

Gentry introduced in [Gen06] an IBE scheme without random oracles with short public parameters and tight security reduction in the standard security model. In this paragraph, we describe the signature scheme obtained by applying Naor transformation to Gentry's IBE. The resulting scheme achieves existential unforgeability under chosen-message attacks in the standard security model.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of the same prime order p (where $p > 2^{2\lambda}$ for a security parameter λ) and let g be a generator of $\mathbb{G}$. We suppose that $(\mathbb{G}, \mathbb{G}_T)$ are equipped with an efficient computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a collision-resistant hash function. Gentry signature scheme is defined by the three following algorithms:

- **Key generation.** The user picks uniformly at random $(x, y) \in \mathbb{Z}_p^2$, computes $h_1 = g^x$ and $h_2 = g^y$ and sets $\mathsf{sk} = (x, y)$ and $\mathsf{pk} = (h_1, h_2) \in \mathbb{G}^2$.

- **Signature generation.** Given a message $m \in \{0,1\}^*$, the user computes its hash value $\mathcal{H}(m)$, and picks uniformly at random $r \in \mathbb{Z}_p$. It computes the *signing exponent* $\sigma = (y+r)/(x+\mathcal{H}(m)) \bmod p$ and the group element $s = g^\sigma$. The signature is the pair $(r,s) \in Z_p \times \mathbb{G}$.
- **Signature verification.** Given $(r,s) \in \mathbb{Z}_p \times \mathbb{G}$, a verifier accepts it as a signature on $m \in \{0,1\}^*$ if and only if the following equality holds:

$$e(s, h_1 g^{\mathcal{H}(m)}) \stackrel{?}{=} e\,(g, h_2 g^r)$$

7.2.2. Description of the Attack

In this section, we use Coppersmith's methods to attack Gentry's signatures when the attacker learns some blocks of consecutive bits of the signing exponents.

Let $n \geqslant 1$ be some integer. We suppose that the attacker is given $(n+2)$ message/signature pairs $(m_i,(r_i,s_i))_{i\in\{0,\ldots,n+1\}}$ as described above (where n does not depend on the security parameter λ). To simplify the notation in the following, instead of the hash values $\mathcal{H}(m_i)$, we assume that the m_i belongs to $\mathbb{Z}_p$ (for $i \in \{0,\ldots,n+1\}$).

We assume that the attacker knows some blocks of consecutive bits of the corresponding signing exponents σ_i for $i \in \{0,\ldots,n+1\}$ and its goal is to recover the secret keys x and y. From the knowledge of two different signing exponents σ_i and σ_j for integers $i,j \in \{0,\ldots,n+1\}$ with $i \neq j$, the attacker can actually recover the secrets x and y. Its goal is therefore to recover the hidden bits of two σ_i's in order to obtain x and y.

We have $\sigma_i = (y+r_i)/(x+m_i) \bmod p$ for $i \in \{0,\ldots,n+1\}$ which can be rewritten as:

$$\sigma_i(x+m_i) - y - r_i = 0 \bmod p, \quad i \in \{0,\ldots,n+1\}.$$

We consider a chosen-message attack where the attacker uses an arbitrary unique message m for all signatures (*i.e.* $m_i = m$ for all $i \in \{0,\ldots,n+1\}$). Eliminating x and y, in the previous equation, we obtain for $a,b,i \in \{0,\ldots,n+1\}$ with $0 \leqslant a < b < i \leqslant n+1$:

$$(r_a - r_b)\sigma_i + (r_i - r_a)\sigma_b + (r_b - r_i)\sigma_a = 0 \bmod p \tag{7.1}$$

Putting $\sigma_i = \sum_{j=1}^{N} x_{i,j} 2^{k_{i,j}} + \gamma_i$, $i \in \{0,\ldots,n+1\}$, where γ_i is known to the attacker and $x_{i,j}$, $j \in \{1,\ldots,N\}$ are unknown and $|x_{i,j}| < 2^{\mu_{i,j}}$ for some integer $\mu_{i,j}$ and with the choice $a=0$, $b=1$, we obtain a polynomial

$$f_i(z_{0,1},\ldots,z_{0,N},\ldots,z_{n+1,1},\ldots,z_{n+1,N})$$

having as root $X_0 = (x_{0,1},\ldots,x_{0,N},\ldots,x_{n+1,1},\ldots,x_{n+1,N})$ modulo p with:

$$f_i = z_{i,N} + \sum_{j=1}^{N-1} a_{i,j} z_{i,j} + \sum_{j=1}^{N} b_{i,j} z_{1,j} + \sum_{j=1}^{N} c_{i,j} z_{0,j} + \gamma_i(r_0 - r_1) + d_i \bmod p \tag{7.2}$$

for $i \in \{2,\ldots,n+1\}$, where

$$\begin{cases} a_{i,j} &= 2^{k_{i,j}}/2^{k_{i,N}} \bmod p \\ b_{i,j} &= 2^{k_{1,j}}(r_i - r_0)/((r_0 - r_1)2^{k_{i,N}}) \bmod p \\ c_{i,j} &= 2^{k_{0,j}}(r_1 - r_i)/((r_0 - r_1)2^{k_{i,N}}) \bmod p \\ d_i &= (\gamma_i(r_0 - r_1) + \gamma_1(r_i - r_0) + \gamma_0(r_1 - r_i))/((r_0 - r_1)2^{k_{i,N}}) \bmod p \end{cases}$$

for $i \in \{2, \ldots, n+1\}$ and $j \in \{1, \ldots, N\}$.

We consider the following collection of polynomials (parameterized by some integer $m \in \mathbb{N}$ that does not depend on the security parameter λ):

$$\mathfrak{P}_m = \left\{ f_{i_{0,1},\ldots,i_{n+1,1},i_{0,2},\ldots,i_{n+1,2},\ldots,i_{0,N},\ldots,i_{n+1,N}} \right\},$$

for all vectors of integers $(i_{0,1}, \ldots, i_{n+1,1}, i_{0,2}, \ldots, i_{n+1,2}, \ldots, i_{0,N}, \ldots, i_{n+1,N})$ verifying

$$0 \leqslant i_{0,1} + \cdots + i_{n+1,1} + \cdots + i_{0,N}, \ldots, i_{n+1,N} \leqslant m$$

and where the polynomial $f_{i_{0,1},\ldots,i_{n+1,1},i_{0,2},\ldots,i_{n+1,2},\ldots,i_{0,N},\ldots,i_{n+1,N}}$ is defined by:

$$z_{0,1}^{i_{0,1}} \cdots z_{n+1,1}^{i_{n+1,1}} \cdots z_{0,N-1}^{i_{0,N-1}} \cdots z_{n+1,N-1}^{i_{n+1,N-1}} z_{0,N}^{i_{0,N}} z_{1,N}^{i_{1,N}} f_2^{i_{2,N}} \cdots f_{n+1}^{i_{n+1,N}} p^{m-(i_{2,N}+\cdots+i_{n+1,N})}.$$

One can see that $f_{i_{0,1},\ldots,i_{n+1,1},i_{0,2},\ldots,i_{n+1,2},\ldots,i_{0,N},\ldots,i_{n+1,N}}(X_0) = 0 \bmod p^m$ for all such vector of integers.

If we use for instance the lexicographical monomial order (with $z_{i,j} < z_{i',j'}$ if ($j < j'$ or ($j = j'$ and $i < i'$)) on the set of monomials, we can define an order over the set of polynomials as:

$$f_{i_{0,1},\ldots,i_{n+1,1},i_{0,2},\ldots,i_{n+1,2},\ldots,i_{0,N},\ldots,i_{n+1,N}} < f_{i'_{0,1},\ldots,i'_{n+1,1},i'_{0,2},\ldots,i'_{n+1,2},\ldots,i'_{0,N},\ldots,i'_{n+1,N}}$$

if $z_{0,1}^{i_{0,1}} \cdots z_{n+1,1}^{i_{n+1,1}} \cdots z_{0,N}^{i_{0,N}} \cdots z_{n+1,N}^{i_{n+1,N}} < z_{0,1}^{i'_{0,1}} \cdots z_{n+1,1}^{i'_{n+1,1}} \cdots z_{0,N}^{i'_{0,N}} \cdots z_{n+1,N}^{i'_{n+1,N}}$.

Using this order, we can write $\mathfrak{P}_m = \{\tilde{f}_i,\ i \in \{1, \ldots, \omega\}\}$, with $\tilde{f}_1 < \tilde{f}_2 < \cdots < \tilde{f}_\omega$ where ω is the number of polynomials. Putting $U = 2^{\max_{i,j} \mu_{i,j}}$, we define the lattice $\mathcal{L}$ generated by $b_1, \ldots, b_\omega$, where for $i \in \{1, \ldots, \omega\}$, b_i is the coefficient vector of the polynomial $\tilde{f}_i(Uz_{0,1}, \ldots, Uz_{n+1,1}, \ldots, Uz_{0,N}, \ldots, Uz_{n+1,N})$.

One can easily verify that the basis matrix is lower triangular and the diagonal elements are $U^a p^{m-(i_{2,N}+\cdots+i_{n+1,N})}$, where the integer a is equal to $i_{0,1} + \cdots + i_{n+1,1} + i_{0,N} + \cdots + i_{n+1,N}$. The number of variables is $N(n+2)$ and the success condition of Coppersmith's method is $\det(\mathcal{L}) < p^{m(\omega - N(n+2))}$, where $\omega = \sum_{\mathbf{i} \in I} 1$ is the dimension of the lattice with

$$I = \{\mathbf{i} = (i_{0,1}, \ldots, i_{0,N}, \ldots, i_{n+1,N}) | 0 \leqslant i_{0,1} + \cdots + i_{n+1,N} \leqslant m\}.$$

We have $\det(\mathfrak{L}) = U^\eta p^{m\omega} p^{-\mu}$ with

$$\mu = \sum_{\mathbf{i} \in I} i_{2,N} + \cdots + i_{n+1,N} \text{ and } \eta = \sum_{\mathbf{i} \in I} i_{0,1} + \cdots + i_{n+1,N}.$$

If m is large, we can neglect the $N(n+2)$ term in Coppersmith success condition and the asymptotic condition becomes:

$$U^\eta < p^\mu.$$

Using analytic combinatorics methods (see below for details), one can verify that when m tends to ∞, we have $\eta = N(n+2)\beta(m, N, n)$ and $\mu = n\beta(m, N, n)$, with

$$\beta(m, N, n) = \frac{m^{N(n+2)+1}}{(N(n+2)+1)!} + o(m^{N(n+2)+1}).$$

Therefore, the attacker can recover x and y as long as the sizes of each unknown block in the signatures σ_i for $i \in \{0, \ldots, n+1\}$ satisfies:

$$U < p^{\frac{n}{(n+2)N}} \underset{n \to \infty}{\to} p^{\frac{1}{N}}.$$

We can thus heuristically recover (using large[2] constant parameters n and m) the secret key (x, y) if the number of consecutive bits of each unknown block is smaller than $\lceil \log_2(p) \rceil / N$.

- In order to compute η, we consider $\mathscr{M}$ (the set of monomials appearing in the collection $\mathfrak{P}_m$) as a combinatorial class, with the size function $S(z_{0,1}^{i_{0,1}} \dots z_{n+1,1}^{i_{n+1,1}} \dots z_{0,N}^{i_{0,N}} \dots z_{n+1,N}^{i_{n+1,N}}) = i_{0,1} + \dots + i_{n+1,1} + \dots + i_{0,N} + \dots + i_{n+1,N}$ and the parameter function $\chi(z_{0,1}^{i_{0,1}} \dots z_{n+1,1}^{i_{n+1,1}} \dots z_{0,N}^{i_{0,N}} \dots z_{n+1,N}^{i_{n+1,N}}) = i_{0,1} + \dots + i_{n+1,1} + \dots + i_{0,N} + \dots + i_{n+1,N}$.
  We describe $\mathscr{M}$ as
  $$\prod_{k=1}^{N(n+2)} \textsc{Seq}(u\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z})$$
  the last $\textsc{Seq}(\mathcal{Z})$ being for the dummy value. This then leads to the OGF
  $$F(z,u) = \left(\frac{1}{1-uz}\right)^{N(n+2)} \left(\frac{1}{1-z}\right) .$$
  We get
  $$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{N(n+2)}{(1-z)^{N(n+2)+2}},$$
  this leads to:
  $$\chi_{<m}(\mathscr{M}) = \eta \sim N(n+2) \times \frac{(m)^{N(n+2)+1}}{(N(n+2)+1)!}.$$
- To compute μ, we consider $\mathscr{M}$ as a combinatorial class, with the size function $S(z_{0,1}^{i_{0,1}} \dots z_{n+1,1}^{i_{n+1,1}} \dots z_{0,N}^{i_{0,N}} \dots z_{n+1,N}^{i_{n+1,N}}) = i_{0,1} + \dots + i_{n+1,1} + \dots + i_{0,N} + \dots + i_{n+1,N}$ and the parameter function $\chi(z_{0,1}^{i_{0,1}} \dots z_{n+1,1}^{i_{n+1,1}} \dots z_{0,N}^{i_{0,N}} \dots z_{n+1,N}^{i_{n+1,N}}) = i_{2,N} + \dots + i_{n+1,N}$.
  We describe $\mathscr{M}$ as
  $$\prod_{k=1}^{N(n+2)-n} \textsc{Seq}(\mathcal{Z}) \times \prod_{k=1}^{n} \textsc{Seq}(u\mathcal{Z}) \times \textsc{Seq}(\mathcal{Z})$$
  the last $\textsc{Seq}(\mathcal{Z})$ being for the dummy value. This then leads to the OGF
  $$F(z,u) = \left(\frac{1}{1-uz}\right)^{n} \left(\frac{1}{1-z}\right)^{N(n+2)-n+1} .$$
  We get
  $$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{n}{(1-z)^{N(n+2)+2}},$$
  this leads to:
  $$\chi_{<m}(\mathscr{M}) = \mu \sim n \times \frac{(m)^{N(n+2)+1}}{(N(n+2)+1)!}.$$

[2] In order to reach this asymptotic bound, the constructed matrix is of huge dimension and the resulting polynomial system has a very large number of variables and the computation which is theoretically polynomial-time becomes in practice prohibitive.

N	n	δ_{theo}	δ_{exp}	dimension	m	LLL time(s)	Gröbner basis time(s)
1	1	0.333	0.32	35	4	3.804	4.603
1	3	0.6	0.49	21	2	0.250	0.699
1	5	0.714	0.49	36	2	0.871	38.374
2	1	0.166	0.16	28	2	1.438	0.650
2	5	0.33	0.29	91	2	191.906	556.715

Table 7.1.: Lattice Attack on Gentry signatures. Average running time (in seconds) of the LLL algorithm and the Gröbner basis computation.

Remark 7.2.1. *In [GV14], Galindo and Vivek analyzed the security of an ElGamal-based public-key encryption scheme (with stateful decryption) proposed by Kiltz and Pietrzak [KP10] which was conjectured to resists lunch-time chosen ciphertext attacks in the so-called* only computation leaks *model. They disprove the conjecture by proposing an algorithm to solve a new computational problem, deemed* Hidden Shares - Hidden Number Problem*, which is less general but has some similarities with the problem investigated in this section.*

7.2.3. Experimental Results

We have implemented the attack in Sage 7.6 on a MacBook Air laptop computer (2,2 GHz Intel Core i7, 4 Gb RAM 1600 MHz DDR3, Mac OSX 10.10.5). Table 7.1 lists the theoretical bound $\delta_{\text{theo}} = \frac{n}{(n+2)N}$ and an experimental bound δ_{exp} for a 512-bit prime p (corresponding to a 256-bit security level) with $(n+2)$ signatures (for $n \in \{1, 3, 5\}$) and a few number of unknown blocks ($N \leqslant 2$). We consider the family of polynomials $\mathfrak{P}_m$ with $m = 4$ and $m = 2$. We ran 2^7 experiments for all parameters and Table 7.1 gives the average running time (in seconds) of the LLL algorithm and the Gröbner basis computation.

We denote α the maximum number of least significant bits that the attacker knows in each signature σ_j, for all $j \neq 0$ (for instance $\alpha = 0$ means that it does not know any least significant bits of the signatures σ_j, for all $j \in \{1, \ldots, n+1\}$). If we know at least $\delta_{\text{exp}}\lceil \log_2(p) \rceil + \alpha$ least significant bits of the signature σ_0 then the Gröbner basis always gives us a system of dimension 0 and we are able to find the N unknown block of sizes $p^{\delta_{\text{exp}}}$ in each signature σ_i for $i \in \{0, \ldots, n+1\}$. Otherwise, Gröbner basis computations gives us a system of dimension 1 and we are *a priori* unable to find the unknown blocks (though it is possible in some cases to obtain additional information). This system of dimension 1 occurs because the constructed system admits a large number of "small" solutions. We give an example of this in Appendix 7.3. However, If the condition mentioned above is satisfied, we obtain for $N = 1$ and $n + 2 = 3$, the success rates given in Table 7.2 (over 250 attacks performed for each parameter pair (m, δ_{exp})).

	$m = 2$	$m = 3$	$m = 4$
$\delta_{\text{exp}} = 0.3225$	100	100	100
$\delta_{\text{exp}} = 0.3250$	98.4	98.4	99.2
$\delta_{\text{exp}} = 0.3275$	90.4	92.8	94.4
$\delta_{\text{exp}} = 0.3300$	66.0	65.2	72.8
$\delta_{\text{exp}} = 0.3325$	10.0	15.2	17.2
$\delta_{\text{exp}} = 0.3350$	0	0	0

Table 7.2.: Lattice Attack on Gentry signatures. Success rates (over 250 attacks performed for each parameter pair (m, δ_{exp})).

7.3. Concrete Attack Examples against Gentry signatures

In this section, we present two attack examples on Gentry signatures for a 256-bit prime p with 3 signatures (r_0, σ_0), (r_1, σ_1) and (r_2, σ_2) and one T-bit unknown block in each signature, with $T = \lfloor 0.3 \log_2(p) \rfloor$.

We recall that for $i \in \{0, 1, 2\}$, $\sigma_i = g^{s_i}$ where $s_i = (y + r_i)/(x + m) \bmod p$, x and y are the secret keys and p, m and r_i, $i \in \{0, 1, 2\}$ are public information. In this example, we took the following random values:

- $p =$ `9b814891e89496e776bfeeebcac5c74130862914fe2b928d40c3a88323dcbaaf`
- $m =$ `440f4a9df2936c4aad3856ed0ea5cf3d131ef658fc36c2fa56763373288d5519`
- $x =$ `57a7b0913f5202e31555ec9538ff90f38a5e6c53b359edfe1106c8ee9518029a`
- $y =$ `259b67be7de53e0546860379bc31ab9bb30caf68c314a956a1719e18d4a24ae2`
- $r_0 =$ `75c471becf6a9d86aa5480985a95702617892ba84b7662d6bdf3a3c1931abf3b`
- $r_1 =$ `675e28ffbf96b29365ebda463c3a0a4290a284f9fed9ddd0ccdada587c1f0152`
- $r_2 =$ `7961b0df3f0a286547f25da59a7c2a7c28764f4335a0aa2cd5a72ba2393a6cd3`
- $s_0 =$ `45f185a8ce35c2b95b3e1aef9fc516ec9e840c9a5b6b36c70532b10145790401`
- $s_1 =$ `8f63fe87fd0d67f6594ff44ba86a2755b2b6ad6a0b7ab4aafecae41fca50c713`
- $s_2 =$ `57de02b444bb7716c021d21162c3727ba904ae6e4d44aca2ad9f4406669e8744`

and $T = \lfloor 0.3 \log_2(p) \rfloor = 76$.

In the first case, we suppose that we do not know any least significant bits of each signature and show that we are unable to find the unknown blocks since the Gröbner basis gives us a system of dimension 1.

In the second case, we suppose that we know $T + 2$ least significant bits of σ_0 but do not know any least significant bits of s_1, and s_2. We also suppose that we do not know T intermediate bits of s_0 and we show that in this case we are able to find the unknown blocks since the Gröbner basis gives us a system of dimension 0.

7.3.0.1. First case

- We can write the signatures as:

$$\begin{aligned} s_0 &= 2^T \cdot \texttt{45f185a8ce35c2b95b3e1aef9fc516ec9e840c9a5b6b3} + z_0, \\ s_1 &= 2^T \cdot \texttt{8f63fe87fd0d67f6594ff44ba86a2755b2b6ad6a0b7ab} + z_1, \\ s_2 &= 2^T \cdot \texttt{57de02b444bb7716c021d21162c3727ba904ae6e4d44a} + z_2, \end{aligned}$$

  where the T-bit numbers z_0, z_1 and z_2 are the unknown blocks.

- We get the polynomial $f(y_0, y_1, y_2)$ defined by:

$$\begin{aligned} y_2 &+ \texttt{86acc2de9d15dab4df6a8114243623f246376c1103c29ee97a0dd7490f87eb33}\, y_1 \\ &+ \texttt{14d485b34b7ebc3297556dd7a68fa34eea4ebd03fa68f3a3c6b5d13a1454cf7b}\, y_0 \\ &+ \texttt{11f10fbe97565b062acfb71c6d98f596de6c1e236edaa9168d891d78d66e8c4a} \end{aligned}$$

  having as root (z_0, z_1, z_2) modulo p.

- Constructing the lattice with $m = 4$, after the LLL reduction and the Gröbner basis computation, we obtain the system of polynomials

$$\begin{cases} f_1(y_0, y_1, y_2) &= y_2 - y_0 - \texttt{5dba86c930521258343} \\ f_2(y_0, y_1, y_2) &= y_1 - y_0 + \texttt{21c0667cce17b283cee} \end{cases}$$

  having indeed (z_0, z_1, z_2) as root over the integers. However, the dimension of the system is 1 and then we are *a priori* unable to find the unknown blocks.

7.3.0.2. Second case

- We can write the signatures as:

$$\begin{aligned} s_0 &= \texttt{36c70532b10145790401} + 2^{79} \cdot z_0 + 2^{79+T} \cdot \texttt{8be30b519c6b8572b67c35df3} \\ s_1 &= 2^T \cdot \texttt{8f63fe87fd0d67f6594ff44ba86a2755b2b6ad6a0b7ab} + z_1 \\ s_2 &= 2^T \cdot \texttt{57de02b444bb7716c021d21162c3727ba904ae6e4d44a} + z_2 \end{aligned}$$

  where the T-bit numbers z_0, z_1 and z_2 are the unknown blocks.

- If one proceeds like in the attack, we obtain the polynomial $f(y_0, y_1, y_2)$ defined by

$$\begin{aligned} y_2 &+ \texttt{86acc2de9d15dab4df6a8114243623f246376c1103c29ee97a0dd7490f87eb33}\, y_1 \\ &+ \texttt{78836c7dbcc6bee53ea07b359a07fa111e09607336b452976acd0f0ec2a0c985}\, y_0 \\ &+ \texttt{77b82eec348f27f19cb7a6c1cc895cf7261093b80d067ea4eb7b8da90e1ae306} \end{aligned}$$

  having as root (z_0, z_1, z_2) modulo p.

- Constructing the lattice with $m = 4$, after the LLL reduction and the Gröbner basis computation, one obtains the system of polynomials

$$\begin{cases} f_1(y_0, y_1, y_2) &= y_2 - \texttt{ca2ad9f4406669e8744} \\ f_2(y_0, y_1, y_2) &= y_1 - \texttt{4aafecae41fca50c713} \\ f_3(y_0, y_1, y_2) &= y_0 - \texttt{f8a2dd93d081934b6d6} \end{cases}$$

  having (z_0, z_1, z_2) as root over the integers. The dimension of the system is 0 and one finds readily the unknown blocks.

7.4. Lattice Attack on Boneh-Boyen Signatures

7.4.1. Boneh-Boyen Signatures

Two years before the proposal of Gentry's IBE, Boneh and Boyen proposed two IBE schemes in [BB04a] and described one signature scheme obtained using the Naor transformation in [BB04b]. Their scheme has comparable efficiency properties and also achieves existential unforgeability under chosen-message attacks in the standard security model.

With the same notation as above, Boneh-Boyen signature scheme is defined by the three following algorithms:

- **Key generation.** The user picks uniformly at random $(x, y) \in \mathbb{Z}_p^2$, computes $h_1 = g^x$ and $h_2 = g^y$ and sets $\mathsf{sk} = (x, y)$ and $\mathsf{pk} = (h_1, h_2) \in \mathbb{G}^2$.
- **Signature generation.** Given a message $m \in \{0,1\}^*$, the user computes its hash value $\mathcal{H}(m)$, and picks uniformly at random $r \in \mathbb{Z}_p$. It computes the *signing exponent* $s = 1/(x + \mathcal{H}(m) + yr) \bmod p$ and the group element $\sigma = g^s$. The signature is the pair $(r, \sigma) \in Z_p \times \mathbb{G}$.
- **Signature verification.** Given $(r, \sigma) \in \mathbb{Z}_p \times \mathbb{G}$, a verifier accepts it as a signature on $m \in \{0,1\}^*$ if and only if the following equality holds:

$$e(\sigma, h_1 \cdot g^{\mathcal{H}(m)} \cdot h_2^r) \stackrel{?}{=} e\,(g, g)$$

7.4.2. Description of the Attack

In this section, we use the Coppersmith's methods to attack Boneh-Boyen's signature. Let $n \geqslant 1$ be some integer. We suppose that the attacker is given $(n+2)$ message/signature pairs $(m_i, (r_i, s_i))_{i \in \{0,\ldots,n+1\}}$ as described above (where n does not depend on the security parameter λ). As above, to simplify the notation, we replace $\mathcal{H}(m_i)$ by $m_i \in \mathbb{Z}_p$ (for $i \in \{0, \ldots, n+1\}$). We assume that the attacker knows some blocks of consecutive bits of the corresponding signing exponents $\sigma_i = 1/(x + m + yr_i) \bmod p$, for $i \in \{0, \ldots, n\}$, where p, r_i and m_i are known to the attacker and x and y are kept secret.
As for Gentry signatures, from the knowledge of two different signing exponents, the attacker can actually recover the secrets x and y and its goal is to recover the hidden bits of two σ_i's in order to recover x and y.

We have $\sigma_i = 1/(x + m_i + yr_i) \bmod p$ for $i \in \{0, \ldots, n+1\}$ and we have:

$$x + m_i + yr_i - \frac{1}{\sigma_i} = 0 \bmod p, \quad i \in \{0, \ldots, n+1\}.$$

Eliminating x and y and assuming again that the attacker chooses a unique message m (namely $m_i = m$, for all $i \in \{0, \ldots, n+1\}$), we obtain, for $a, b, i \in \{0, \ldots, n+1\}$ with $0 \leqslant a < b < i \leqslant n+1$:

$$(r_b - r_i)\sigma_i\sigma_b + (r_i - r_a)\sigma_i\sigma_a + (r_a - r_b)\sigma_a\sigma_b = 0 \bmod p. \tag{7.3}$$

Putting $\sigma_i = \sum_{j=1}^{N} x_{i,j} 2^{k_{i,j}} + \gamma_i$, $i \in \{0, \ldots, n+1\}$, where γ_i is known to the attacker and $x_{i,j}$, $j \in \{1, \ldots, N\}$ are unknown with $|x_{i,j}| < 2^{\mu_{i,j}}$ for some integer $\mu_{i,j}$ and $a = 0$,

we obtain a polynomial $f_{0,b,i}(z_{0,1},\ldots,z_{0,N},\ldots,z_{n+1,1},\ldots,z_{n+1,N})$ having as "small" root $X_0=(x_{0,1},\ldots,x_{0,N},\ldots,x_{n+1,1},\ldots,x_{n+1,N})$ modulo p, where :

$$\begin{aligned} f_{0,b,i} &= \sum_{j=1}^{N}\sum_{k=1}^{N}\alpha_{b,i,j,k}z_{i,j}z_{b,k}+\sum_{j=1}^{N}\sum_{k=1}^{N}\alpha_{0,i,j,k}z_{i,j}z_{0,k}+\sum_{j=1}^{N}\sum_{k=1}^{N}\alpha_{0,b,j,k}z_{b,j}z_{0,k} \\ &\quad +\sum_{j=1}^{N}\alpha_{0,b,i,j}z_{i,j}+\sum_{j=1}^{N}\beta_{0,b,i,j}z_{b,j}+\sum_{j=1}^{N}\gamma_{0,b,i,j}z_{0,j}+\delta_{0,b,i} \bmod p \end{aligned} \tag{7.4}$$

for $b,i\in\{1,\ldots,n+1\}$, $b<i$ and with known coefficients , where $\alpha_{b,i,N,N}=1$. The set of monomials appearing in the polynomials $f_{0,b,i}$ is:

$$\mathfrak{M}=\left\{1, z_{a,j}z_{b,k}, z_{i,j} : i\in\{0,\ldots,n+1\} \;\middle|\; \begin{array}{l} a,b\in\{0,\ldots,n+1\}; a<b \\ j,k\in\{0,\ldots,N\}\end{array}\right\}.$$

We consider the following set of polynomials:

$$\mathfrak{P}=\{p\tilde{m},\tilde{m}\in\mathfrak{M}_1\}\cup\{f_{0,b,i} : b,i\in\{1,\ldots,n+1\}; b<i\},$$

where $\mathfrak{M}_1=\mathfrak{M}\setminus\mathfrak{M}_2$ with $\mathfrak{M}_2=\{z_{b,N}z_{i,N} : b,i\in\{1,\ldots,n+1\}; b<i\}$. One can see that for any polynomial $\tilde{f}\in\mathfrak{P}$, $\tilde{f}(X_0)=0 \bmod p$. We can define an order on the set of monomials such that all the monomials in $\mathfrak{M}_1$ are smaller than any monomial in $\mathfrak{M}_2$ and for $z_{b,N}z_{i,N}, z_{b',N}z_{i',N}\in\mathfrak{M}_2$, $z_{b,N}z_{i,N}<z_{b',N}z_{i',N}$ if $(b<b'$ or $(b=b'$ and $i<i'))$.

Using that order, we can order the set of polynomials from the smallest element to the greatest as follows:

$$\begin{aligned}\mathfrak{P} &= \{p\tilde{m}_1,\ldots,p\tilde{m}_{\omega_1},f_{0,1,2},\ldots,f_{0,1,n+1},f_{0,2,3},\ldots,f_{0,2,n+1},\ldots,f_{0,n,n+1}\} \\ &= \{\tilde{f}_1,\ldots,\tilde{f}_\omega\}\end{aligned}$$

where $\tilde{m}_1<\cdots<\tilde{m}_{\omega_1}$, ω_1 is the cardinality of $\mathfrak{M}_1$ and ω is the cardinality of $\mathfrak{M}$.

Putting $U=2^{\max_{i,j}\mu_{i,j}}$, we define the lattice $\mathcal{L}$ generated by $b_1,\ldots,b_\omega$, where for each $i\in\{1,\ldots,\omega\}$, b_i is the coefficient vector of the polynomial

$$\tilde{f}_i(Uz_{0,1},\ldots,Uz_{0,N},\ldots,Uz_{n+1,1},\ldots,Uz_{n+1,N}).$$

One can verify that the basis matrix is lower triangular. The number of variables is $N(n+2)$ and the success condition for the Coppersmith's method is:

$$\det(\mathcal{L})<p^{\omega-N(n+2)+1},\ \text{with}\ \omega=\sharp\mathfrak{M}=N^2\frac{(n+1)(n+2)}{2}+(n+2)N+1.$$

We have $\det(\mathcal{L})=U^{2N^2\frac{(n+1)(n+2)}{2}+(n+2)N}p^{\omega-\frac{n(n+1)}{2}}$ and the success condition becomes:

$$U<p^{\frac{\frac{n(n+1)}{2}-N(n+2)+1}{2N^2\frac{(n+1)(n+2)}{2}+(n+2)N}}.$$

If n is large and since N is small, we can neglect $-N(n+2)+1$ which contribute to a small error term. So the attacker can recover x and y as long as the sizes of each unknown block in the signatures σ_i, $i\in\{0,\ldots,n+1\}$ satisfies:

$$U<p^{\frac{n(n+1)}{2N^2(n+1)(n+2)+2(n+2)N}}\underset{n\to\infty}{\to}p^{\frac{1}{2N^2}}.$$

We can thus heuristically recover the secret key if the number of consecutive bits of each unknown block is smaller than $\lceil\log_2(p)\rceil/(2N^2)$.

N	n	δ_{theo}	δ_{exp}	dimension	LLL time(s)	Gröbner basis time(s)
1	4	0.277	0.293	22	0.205	0.048
1	6	0.306	0.31	29	1.961	1.008
1	10	0.382	0.38	79	75.086	39.669
2	4	0.076	0.08	73	9.185	3.078
2	6	0.087	0.09	129	232.698	397.900

Table 7.3.: Lattice Attack on Boneh-Boyen signatures. Average running time (in seconds) of the LLL algorithm and the Gröbner basis computation.

7.4.3. Experimental results

Table 7.3 lists the theoretical bound $\delta_{\text{theo}} = \frac{n(n+1)}{2N^2(n+1)(n+2)+2(n+2)N}$ and an experimental bound δ_{exp} for a 512-bit prime p with $(n+2)$ signatures for a few values of $n \in \{4, 6, 10\}$ and one or two unknown blocks per signatures.

We ran 2^7 experiments for all parameters and in all cases (for the bound δ_{exp}), the assumption that the created polynomials define an algebraic variety of dimension 0 was verified. The constructed system was solved using Gröbner basis and the desired root recovered. Table 7.3 gives the average running time (in seconds) of the LLL algorithm and the Gröbner basis computation (using the same configuration as above).

7.5. Lattice Attack on Sakai-Kasahara Signatures

7.5.1. Sakai-Kasahara Signatures

In [SK03], Sakai and Kasahara presented the first pairing-based signature scheme in the exponent-inversion framework. Their scheme is very close to Boneh-Boyen signature schemes but produces shorter signatures (at the cost of relying on the random oracle heuristic [ZSS04]).

With the same notation as above, Sakai-Kasahara signature scheme is defined by the three following algorithms:

- **Key generation.** The user picks uniformly at random $x \in \mathbb{Z}_p$, computes $h = g^x$ and sets $\mathsf{sk} = x$ and $\mathsf{pk} = h \in \mathbb{G}$.
- **Signature generation.** Given a message $m \in \{0,1\}^*$, the user computes its hash value $\mathcal{H}(m)$. It computes the *signing exponent* $s = 1/(x + \mathcal{H}(m)) \bmod p$ and the group element $\sigma = g^s$. The signature is the group element $\sigma \in \mathbb{G}$.
- **Signature verification.** Given $\sigma \in \mathbb{G}$, a verifier accepts it as a signature on $m \in \{0,1\}^*$ if and only if the following equality holds:

$$e(\sigma, h \cdot g^{\mathcal{H}(m)}) \stackrel{?}{=} e(g, g)$$

We present in the following an attack on this scheme when the attacker learns some blocks of consecutive bits of the signing exponents. This computational problem is related to the Modular Inversion Hidden Number Problem which was introduced in 2001 by Boneh, Halevi and Howgrave-Graham [BHH01]. In this problem, the attacker does not know exactly one

block of least significant bits of the signing exponents σ_i while our attack considers the setting where the attacker does not know $N \geqslant 1$ different blocks in each σ_i (for any N).

7.5.2. Description of the Attack

In this section, we use the Coppersmith's methods to attack Sakai-Kasahara signatures. Let $n \geqslant 1$ be some integer. We suppose that the attacker is given $(n+1)$ message/signature pairs $(m_i, s_i)_{i\in\{0,\ldots,n\}}$ as described above (where n does not depend on the security parameter λ). Again, to simplify the notation, we replace $\mathcal{H}(m_i)$ by $m_i \in \mathbb{Z}_p$ (for $i \in \{0,\ldots,n\}$). We assume that the attacker knows some blocks of consecutive bits of the corresponding signing exponents $\sigma_i = 1/(x+m_i) \bmod p$ for $i \in \{0,\ldots,n\}$ and its goal is to recover x. One can see that from the knowledge of a value σ_i, the attacker can actually recover the hidden number x and it is thus sufficient to recover the hidden bits of a single σ_i's in order to recover x.

We have $\sigma_i = 1/(x+m_i) \bmod p$ for $i \in \{0,\ldots,n\}$ which can be rewritten as:

$$x + m_i - \frac{1}{\sigma_i} = 0 \bmod p, \quad i \in \{0,\ldots,n\}.$$

Eliminating x, we obtain:

$$(m_i - m_a)\sigma_i\sigma_a + \sigma_i - \sigma_a = 0 \bmod p \quad a, i \in \{0,\ldots,n\}, 0 \leqslant a < i \leqslant n. \tag{7.5}$$

Putting, for $i \in \{0,\ldots,n+1\}$, $\sigma_i = \sum_{j=1}^{N} x_{i,j}2^{k_{i,j}} + \gamma_i$, where γ_i is known to the attacker and $x_{i,j}$ for $j \in \{1,\ldots,N\}$ are unknown with $|x_{i,j}| < 2^{\mu_{i,j}}$ for some integer $\mu_{i,j}$, we obtain a polynomial $f_{a,i}(z_{0,1},\ldots,z_{0,N},\ldots,z_{n,1},\ldots,z_{n,N})$ having as root $X_0 = (x_{0,1},\ldots,x_{0,N},\ldots,x_{n,1},\ldots,x_{n,N})$ modulo p with:

$$f_{a,i} = \sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_{a,i,j,k} z_{i,j} z_{a,k} + \sum_{j=1}^{N} \beta_{a,i,j} z_{i,j} + \sum_{j=1}^{N} \gamma_{a,i,j} x_{a,j} + \delta_{a,i} \bmod p \tag{7.6}$$

for $a, i \in \{0,\ldots,n\}$, $a < i$ and with known coefficients, where $\alpha_{a,i,N,N} = 1$. The set of monomials appearing in the polynomials $f_{a,i}$ is:

$$\mathfrak{M} = \{1, z_{a,j}z_{b,k}, z_{i,j} : i \in \{0,\ldots,n\}; a, b \in \{0,\ldots,n\}; a < b; j, k \in \{1,\ldots,N\}\}.$$

We consider the following set of polynomials:

$$\mathfrak{P} = \{p\tilde{m}, \tilde{m} \in \mathfrak{M}_1\} \cup \{f_{a,i} : a, i \in \{0,\ldots,n\}; a < i\},$$

where $\mathfrak{M}_1 = \mathfrak{M} \setminus \mathfrak{M}_2$ with $\mathfrak{M}_2 = \{z_{a,N}z_{i,N} : a, i \in \{0,\ldots,n\}; a < i\}$. One can see that for any polynomial $\tilde{f} \in \mathfrak{P}$, $\tilde{f}(X_0) = 0 \bmod p$. We can define an order on the set of monomials such that all the monomials in $\mathfrak{M}_1$ are smaller than any monomial in $\mathfrak{M}_2$ and for $z_{a,N}z_{i,N}, z_{a',N}z_{i',N} \in \mathfrak{M}_2$, $z_{a,N}z_{i,N} < z_{a',N}z_{i',N}$ if $(a < a'$ or $(a = a'$ and $i < i'))$.

Using that order, we can order the set of polynomials from the smallest element to the greatest as follows:

$$\mathfrak{P} = \{p\tilde{m}_1,\ldots,p\tilde{m}_{\omega_1}, f_{0,1},\ldots,f_{0,n},f_{1,2},\ldots,f_{1,n},\ldots,f_{n-1,n}\} = \{\tilde{f}_1,\ldots,\tilde{f}_\omega\}$$

where $\tilde{m}_1 < \cdots < \tilde{m}_{\omega_1}$, ω_1 is the cardinality of $\mathfrak{M}_1$ and ω is the cardinality of $\mathfrak{M}$. Putting $U = 2^{\max_{i,j}\mu_{i,j}}$, we define the lattice $\mathcal{L}$ generated by $b_1,\ldots,b_\omega$, where b_i is the coefficient

N	n	δ_{theo}	δ_{exp}	dimension	LLL time(s)	Gröbner basis time(s)
1	4	0.4	0.39	16	0.015	0.009
1	6	0.4285	0.425	29	0.934	0.267
1	10	0.4545	0.45	67	5.082	4.247
2	4	0.1111	0.1111	51	0.728	0.292
2	6	0.1153	0.1153	99	15.308	14.482

Table 7.4.: Lattice Attack on Sakai-Kasahara signatures. Average running time (in seconds) of the LLL algorithm and the Gröbner basis computation.

vector of $\tilde{f}_i(Uz_{0,1}, \ldots, Uz_{0,N}, \ldots, Uz_{n,1}, \ldots, Uz_{n,N})$ for $i \in \{1, \ldots, \omega\}$,. One can easily verify that the basis matrix is lower triangular. The number of variables is $N(n+1)$ and the success condition for the Coppersmith's method is:

$$\det(\mathcal{L}) < p^{\omega - N(n+1)+1},$$

with $\omega = \sharp\mathfrak{M} = N^2\frac{n(n+1)}{2} + (n+1)N + 1$ and $\det(\mathcal{L}) = U^{2N^2\frac{n(n+1)}{2}+(n+1)N} p^{\omega - \frac{n(n+1)}{2}}$. The success condition then becomes:

$$U < p^{\frac{\frac{n(n+1)}{2} - N(n+1)+1}{2N^2\frac{n(n+1)}{2}+(n+1)N}}.$$

If n is large and since N is small, we can neglect $-N(n+1)+1$ which contributes to a small error. The attacker can recover x and y as long as the sizes of each unknown block in the signatures σ_i, $i \in \{0, \ldots, n\}$ satisfies:

$$U < p^{\frac{n(n+1)}{2N^2 n(n+1))+2(n+1)N}} \underset{n\to\infty}{\to} p^{\frac{1}{2N^2}}.$$

We can heuristically recover the secret key of Sakai-Kasahara signatures if the number of consecutive bits of each unknown block is smaller than $\lceil \log_2(p) \rceil / (2N^2)$.

7.5.3. Experimental results

Table 7.4 gives the theoretical bound $\delta_{\text{theo}} = \frac{n(n+1)}{2N^2 n(n+1))+2(n+1)N}$ and an experimental bound δ_{exp} for a 512-bit prime p with $(n+1)$ signatures for a few values of $n \in \{4, 6, 10\}$ and one or two unknown blocks per signatures.

We ran 2^7 experiments for all parameters . As in the attack on Boneh-Boyen signatures, the assumption that the created polynomials define an algebraic variety of dimension 0 was verified (in all cases for the bound δ_{exp}) and the constructed system was solved using Gröbner basis and the desired root recovered. Table 7.4 gives the average running time (in seconds) of the LLL algorithm and the Gröbner basis computation (using the same configuration as above).

Chapter 8.

Conclusion and Open Questions

8.1. Conclusion

In this thesis:

- We proved lower bounds on the degree and weight of multivariate polynomial representations of the Naor-Reingold function over a finite field and over the group of points on an elliptic curve over a finite field for fixed secret keys and variable secret keys. For fixed secret keys, we showed that a low-weight or low-degree multivariate polynomial cannot reveal information on the functions values over finite fields and that a low-degree univariate and bivariate polynomial cannot reveal information on the functions values over elliptic curves. For variable secret keys, we showed that a low-weight or low-degree multivariate polynomial cannot reveal information on the functions values over finite fields and that a low-degree multivariate polynomial cannot reveal information on the functions values over elliptic curves in certain cases.

- We studied the distribution of the Dodis-Yampolskiy pseudo-random function values over finite fields and over elliptic curves. We showed that for almost all values of parameters, the Dodis-Yampolskiy pseudo-random function produces a uniformly distributed sequence. We also proved lower bounds on the degree of polynomials interpolating the values of these functions in these two settings of practical interest. We showed that a low-weight or low-degree univariate polynomial cannot reveal the secret key x when evaluated at $V_x(m)$ (for some integer $m \in \{1, \ldots, d\}$) for all x over finite fields and that a low-degree univariate cannot reveal the secret key x when evaluated at $V_x(m)$ over elliptic curves.

- We analyzed the security of the elliptic curve linear congruential generator (EC-LCG) and of the elliptic curve power generator (EC-PG). In the case where the *composer* is known, we showed that the EC-LCG is insecure if at least a proportion of 8/11 of the most significant bits of an arbitrary large number of consecutive values U_i of the sequence is output. We also tackled the case where the most significant bits of an arbitrary large number of non consecutive values (namely the most significant bits of the abscissa of values U_{ki} for some fixed integer k) of the sequence is output and we showed that the EC-LCG is insecure if at least a proportion of 3/4 of the most significant bits is output. Furthermore, we consider the cryptographic setting where the *composer* is unknown and we showed that this generator is insecure if at least a proportion of 7/8

of the most significant bits of an arbitrary large number of consecutive values U_i of the sequence is output. Finally, we showed that the EC-PG is insecure if a proportion of at least $1 - 1/e^2$ of the most significant bits of the abscissa of an arbitrary large number of consecutive values V_i of the sequence is output. Our results are theoretical since in practice, the performance of Coppersmith's method in our attacks is bad because of large dimension of the constructed lattice but they are good evidences of the weaknesses of these generators. These generators should then be used with great care.

- We presented lattice-based polynomial-time algorithms that recover the signer's secret in popular pairing-based signatures (Gentry signature, Boneh-Boyen signature and Sakai-Kasahara signature) when used to sign several messages under the assumption that N blocks of consecutive bits of the corresponding exponents are known by the attacker. This partial information can be obtained in practice easily through side-channels (such as the power consumption or the electromagnetic emanations of the device generating the signature). We considered known-message attacks and chosen-message attacks. We show that one can recover the secret key if the number of consecutive bits of each unknown block is smaller than the following values:
  - **Sakai-Kasahara:** $\lceil \log_2(p) \rceil / 2N^2$
  - **Boneh-Boyen:** $\lceil \log_2(p) \rceil / 2N^2$
  - **Gentry:** $\lceil \log_2(p) \rceil / N$

  provided that the number of signatures is sufficiently large. The efficiency of our (heuristic) attacks has been validated experimentally.

8.2. Open questions

Many open questions still remain:

- For the Naor-Reingold pseudo-random functions: the first question which is natural is
  Question 8.1. *Can we generalize our bounds to smaller interpolating sets?*

  the second question is
  Question 8.2. *Can we obtain lower bounds on the weight of multivariate polynomial representations of the Naor-Reingold functions over elliptic curves?*

  and the third question
  Question 8.3. *Can we obtain lower bounds on the non-linear complexity of the Naor-Reingold functions over elliptic curves?*

- For the Dodis-Yampolskiy pseudo-random functions: the first question is
  Question 8.4. *Can we study the distribution of k-tuples $(V_x(m), \ldots, V_x(m+k))_m$?*

  and the second one
  Question 8.5. *Can we study linear complexity, non-linear complexity and minimal polynomials of the sequence generated by the Dodis-Yampolskiy functions over finite fields and over elliptic curves?*

- For lattice attacks on pairings-based schemes: In order to prevent the leakage of partial information on the exponent, it is customary to use a probabilistic algorithm to encode the sensitive values such that the cryptographic operations only occur on randomized

data. In [Cor99], Coron proposed notably to randomize the exponent and the projective coordinates of the base point. The first question is:

Question 8.6. *Can we extend our attacks in such setting (as it was done recently for ECDSA in [GRV17])?*

Our attacks are heuristic and it would be very interesting to give proven version of our attacks (as it was done in [NS02; NS03] for ECDSA signatures). It is also interesting to study further the attack against Gentry signatures when the unknown blocks of consecutive bits overlap. Finally, it would be nice to improve our attacks on Boneh-Boyen and Sakai-Kasahara signatures and to show that one can recover the secret key if the number of consecutive bits of each unknown block is smaller than $\lceil \log_2(p) \rceil / (cN)$ for some constant c.

Bibliography

[Adl79] L.M. Adleman. "A Subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography". In: *Proceedings of the 20th Annual Symposium on Foundations of Computer Science, SFCS 1979*. IEEE Computer Society, Washington, DC, USA. 1979, pp. 55–60 (cit. on p. 22).

[BB04a] Dan Boneh and Xavier Boyen. "Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles". In: *Advances in Cryptology – EUROCRYPT 2004*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. Lecture Notes in Computer Science. Interlaken, Switzerland: Springer, Heidelberg, Germany, May 2004, pp. 223–238 (cit. on pp. 3, 4, 14, 101, 109).

[BB04b] Dan Boneh and Xavier Boyen. "Short Signatures Without Random Oracles". In: *Advances in Cryptology – EUROCRYPT 2004*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. Lecture Notes in Computer Science. Interlaken, Switzerland: Springer, Heidelberg, Germany, May 2004, pp. 56–73 (cit. on pp. 101, 109).

[BB08] Dan Boneh and Xavier Boyen. "Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups". In: *Journal of Cryptology* 21.2 (Apr. 2008), pp. 149–177 (cit. on p. 101).

[BCP07] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. "Provably secure authenticated group Diffie-Hellman key exchange". In: *ACM Trans. Inf. Syst. Secur.* 10.3 (2007), p. 10 (cit. on p. 35).

[BCTV16] Fabrice Benhamouda, Céline Chevalier, Adrian Thillard, and Damien Vergnaud. "Easing Coppersmith Methods Using Analytic Combinatorics: Applications to Public-Key Cryptography with Weak Pseudorandomness". In: *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part II*. Ed. by Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang. Vol. 9615. Lecture Notes in Computer Science. Taipei, Taiwan: Springer, Heidelberg, Germany, Mar. 2016, pp. 36–66. DOI: 10.1007/978-3-662-49387-8_3 (cit. on pp. 63, 67, 78).

[BD00] Dan Boneh and Glenn Durfee. "Cryptanalysis of RSA with private key d less than $N^{0.292}$". In: *IEEE Trans. Information Theory* 46.4 (2000), pp. 1339–1349 (cit. on p. 63).

[BF01] Dan Boneh and Matthew K. Franklin. "Identity-Based Encryption from the Weil Pairing". In: *Advances in Cryptology – CRYPTO 2001*. Ed. by Joe Kilian. Vol. 2139. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 2001, pp. 213–229 (cit. on p. 101).

[BGLS00] William D. Banks, Frances Griffin, Daniel Lieman, and Igor Shparlinski. "Non-linear Complexity of the Naor-Reingold Pseudo-random Function". In: *ICISC 99: 2nd International Conference on Information Security and Cryptology*. Ed. by JooSeok Song. Vol. 1787. Lecture Notes in Computer Science. Seoul, Korea: Springer, Heidelberg, Germany, Dec. 2000, pp. 53–59 (cit. on pp. 11, 29, 32).

[BHH01] Dan Boneh, Shai Halevi, and Nick Howgrave-Graham. "The Modular Inversion Hidden Number Problem". In: *Advances in Cryptology – ASIACRYPT 2001*. Ed. by Colin Boyd. Vol. 2248. Lecture Notes in Computer Science. Gold Coast, Australia: Springer, Heidelberg, Germany, Dec. 2001, pp. 36–51 (cit. on pp. 102, 111).

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing". In: *Advances in Cryptology – ASIACRYPT 2001*. Ed. by Colin Boyd. Vol. 2248. Lecture Notes in Computer Science. Gold Coast, Australia: Springer, Heidelberg, Germany, Dec. 2001, pp. 514–532 (cit. on p. 101).

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing". In: *Journal of Cryptology* 17.4 (Sept. 2004), pp. 297–319 (cit. on p. 101).

[BM03] J. Blomer and A. May. "New partial key exposure attacks on RSA". In: *Dan Boneh, editor, Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference*. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 27–43 (cit. on p. 63).

[BM84] M. Blum and S. Micali. "How to Generate Cryptographically Strong Sequences of Pseudorandom Bits". In: *SIAM Journal on Computing* 13.4 (1984), pp. 850–864 (cit. on p. 6).

[Bom66] E. Bombieri. "On exponential sums in finite fields". In: *Amer. J. Math. 88(1966)* (1966), pp. 71–105 (cit. on p. 26).

[Bon98] D. Boneh. ""The Decision Diffie–Hellman Problem". In: *Proceedings of the Third Algorithmic Number Theory Symposium*. Vol. 1423. Lecture Notes in Computer Science. Springer, 1998, pp. 48–63 (cit. on p. 3).

[Boy08] Xavier Boyen. "A tapestry of identity-based encryption: practical frameworks compared". In: *IJACT* 1.1 (2008), pp. 3–21 (cit. on p. 101).

[Boy89] J. Boyar. "Inferring sequences produced by a linear congruential generator missing low–order bits". In: *J.Cryptology* 1 (1989), pp. 177–184 (cit. on p. 8).

[BS08] Jean Bourgain and Igor E. Shparlinski. "Distribution of consecutive modular roots of an integer." In: *Acta Arith.* 134.1 (2008), pp. 83–91 (cit. on p. 23).

[BSS99] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Elliptic curves in cryptography*. Cambridge: Cambridge University Press, 1999, pp. xv + 204. ISBN: 0-521-65374-6/pbk (cit. on p. 19).

[BV98] Dan Boneh and Ramarathnam Venkatesan. "Breaking RSA May Not Be Equivalent to Factoring". In: *Advances in Cryptology – EUROCRYPT'98*. Ed. by Kaisa Nyberg. Vol. 1403. Lecture Notes in Computer Science. Espoo, Finland: Springer, Heidelberg, Germany, May 1998, pp. 59–71 (cit. on p. 4).

[BVZ12] Aurélie Bauer, Damien Vergnaud, and Jean-Christophe Zapalowicz. “Inferring Sequences Produced by Nonlinear Pseudorandom Number Generators Using Coppersmith’s Methods”. In: *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*. Ed. by Marc Fischlin, Johannes Buchmann, and Mark Manulis. Vol. 7293. Lecture Notes in Computer Science. Darmstadt, Germany: Springer, Heidelberg, Germany, May 2012, pp. 609–626 (cit. on pp. 63, 78).

[CGS10] Marcos Cruz, Domingo Gómez, and Daniel Sadornil. “On the linear complexity of the Naor-Reingold sequence with elliptic curves.” In: *Finite Fields Appl.* 16.5 (2010), pp. 329–333. ISSN: 1071-5797 (cit. on pp. 11, 29).

[Che10] Jung Hee Cheon. “Discrete Logarithm Problems with Auxiliary Inputs”. In: *Journal of Cryptology* 23.3 (July 2010), pp. 457–476 (cit. on pp. 3, 51).

[CHK+06] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. “How to win the clonewars: Efficient periodic n-times anonymous authentication”. In: *ACM CCS 06: 13th Conference on Computer and Communications Security*. Ed. by Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati. Alexandria, Virginia, USA: ACM Press, Oct. 2006, pp. 201–210 (cit. on pp. 11, 51, 56).

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. “Compact E-Cash”. In: *Advances in Cryptology – EUROCRYPT 2005*. Ed. by Ronald Cramer. Vol. 3494. Lecture Notes in Computer Science. Aarhus, Denmark: Springer, Heidelberg, Germany, May 2005, pp. 302–321 (cit. on pp. 11, 51, 56).

[Cop96a] Don Coppersmith. “Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known”. In: *Advances in Cryptology – EUROCRYPT’96*. Ed. by Ueli M. Maurer. Vol. 1070. Lecture Notes in Computer Science. Saragossa, Spain: Springer, Heidelberg, Germany, May 1996, pp. 178–189 (cit. on pp. 63, 65).

[Cop96b] Don Coppersmith. “Finding a Small Root of a Univariate Modular Equation”. In: *Advances in Cryptology – EUROCRYPT’96*. Ed. by Ueli M. Maurer. Vol. 1070. Lecture Notes in Computer Science. Saragossa, Spain: Springer, Heidelberg, Germany, May 1996, pp. 155–165 (cit. on p. 63).

[Cor99] Jean-Sébastien Coron. “Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems”. In: *Cryptographic Hardware and Embedded Systems – CHES’99*. Ed. by Çetin Kaya Koç and Christof Paar. Vol. 1717. Lecture Notes in Computer Science. Worcester, Massachusetts, USA: Springer, Heidelberg, Germany, Aug. 1999, pp. 292–302 (cit. on p. 117).

[CS00] Don Coppersmith and Igor Shparlinski. “On Polynomial Approximation of the Discrete Logarithm and the Diffie-Hellman Mapping”. In: *Journal of Cryptology* 13.3 (2000), pp. 339–360 (cit. on pp. 12, 26, 27, 32).

[CS98] Ronald Cramer and Victor Shoup. “A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack”. In: *Advances in Cryptology – CRYPTO’98*. Ed. by Hugo Krawczyk. Vol. 1462. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 1998, pp. 13–25 (cit. on p. 3).

[DH76] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Trans Inf Theory* 22.6 (1976), pp. 644–654 (cit. on pp. 2, 3).

[Die11] C. Diem. "On the discrete logarithm problem in elliptic curves". In: *Compos. Math* 147 (2011), pp. 75–104 (cit. on p. 22).

[DT97] M. Drmota and R. Tichy. *discrepancies and applications*. Springer-Verlag, Berlin, 1997 (cit. on p. 53).

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. "A Verifiable Random Function with Short Proofs and Keys". In: *PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography*. Ed. by Serge Vaudenay. Vol. 3386. Lecture Notes in Computer Science. Les Diablerets, Switzerland: Springer, Heidelberg, Germany, Jan. 2005, pp. 416–431 (cit. on pp. 3, 11, 51).

[ElG85] T. ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *IEEE Trans Inf Theory* 31.4 (1985), pp. 469–472 (cit. on p. 3).

[FHK+88] Alan M. Frieze, Johan Håstad, Ravi Kannan, J. C. Lagarias, and Adi Shamir. "Reconstructing Truncated Integer Variables Satisfying Linear Congruences". In: *SIAM J. Comput.* 17.2 (1988), pp. 262–280 (cit. on p. 8).

[FPPR12] J-Ch. Faugere, L. Perret, Ch. Petit, and G. Renault. "Improving the complexity of index calculus algorithms in elliptic curves over binary fields". In: *EUROCRYPT 2012*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 27–44 (cit. on p. 22).

[FS09] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009 (cit. on pp. 63, 67, 72).

[Gau09] P. Gaudry. "Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem". In: *J. Symbolic Comput* 44 (2009), pp. 1690–1702 (cit. on p. 22).

[GBS99] Guang Gong, Thomas A. Berson, and Douglas R. Stinson. "Elliptic Curve Pseudorandom Sequence Generators". In: *Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99, Kingston, Ontario, Canada, August 9-10, 1999, Proceedings*. 1999, pp. 34–48 (cit. on p. 78).

[Gen06] Craig Gentry. "Practical Identity-Based Encryption Without Random Oracles". In: *Advances in Cryptology – EUROCRYPT 2006*. Ed. by Serge Vaudenay. Vol. 4004. Lecture Notes in Computer Science. St. Petersburg, Russia: Springer, Heidelberg, Germany, May 2006, pp. 445–464 (cit. on pp. 4, 14, 101, 102).

[GGI11] Domingo Gómez, Jaime Gutierrez, and Álvar Ibeas. "On the linear complexity of the Naor-Reingold sequence". In: *Inf. Process. Lett.* 111.17 (2011), pp. 854–856 (cit. on pp. 11, 29, 32).

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to Construct Random Functions (Extended Abstract)". In: *25th Annual Symposium on Foundations of Computer Science*. Singer Island, Florida: IEEE Computer Society Press, Oct. 1984, pp. 464–479 (cit. on p. 10).

[GI07] J. Gutierrez and A. Ibeas. “Inferring sequences produced by a linear congruential generator on elliptic curves missing high-order bits.” In: *Des. Codes Cryptography* 45 (2007), pp. 199–212 (cit. on pp. 8, 13, 86).

[GL01] G. Gong and C. C. Y. Lam. “Linear recursive sequences over elliptic curves.” In: *intern. conf. on sequences and their applications, Bergen 2001.* Springer-Verlag, London, 2001, pp. 182–196 (cit. on p. 78).

[GL89] Oded Goldreich and Leonid A. Levin. “A Hard-Core Predicate for all One-Way Functions”. In: *21st Annual ACM Symposium on Theory of Computing.* Seattle, Washington, USA: ACM Press, May 1989, pp. 25–32 (cit. on p. 10).

[Gol04] O. Goldreich. *Foundations of cryptography: Basic Applications.* Cambridge University Press, 2004 (cit. on p. 12).

[GRV17] Dahmun Goudarzi, Mathieu Rivain, and Damien Vergnaud. “Lattice Attacks against Elliptic-Curve Signatures with Blinded Scalar Multiplication”. In: *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John’s, NL, Canada, August 9-12, 2016, Revised Selected Papers.* Ed. by Roberto Avanzi and Howard Heys. Vol. to appear. Lecture Notes in Computer Science. Springer, 2017 (cit. on p. 117).

[GV14] David Galindo and Srinivas Vivek. “Limits of a conjecture on a leakage-resilient cryptosystem”. In: *Inf. Process. Lett.* 114.4 (2014), pp. 192–196 (cit. on p. 106).

[HM09] Mathias Herrmann and Alexander May. “Attacking Power Generators Using Unravelled Linearization: When Do We Output Too Much?” In: *Advances in Cryptology – ASIACRYPT 2009.* Ed. by Mitsuru Matsui. Vol. 5912. Lecture Notes in Computer Science. Tokyo, Japan: Springer, Heidelberg, Germany, Dec. 2009, pp. 487–504 (cit. on pp. 9, 63).

[HM10] Mathias Herrmann and Alexander May. “Maximizing Small Root Bounds by Linearization and Applications to Small Secret Exponent RSA”. In: *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography.* Ed. by Phong Q. Nguyen and David Pointcheval. Vol. 6056. Lecture Notes in Computer Science. Paris, France: Springer, Heidelberg, Germany, May 2010, pp. 53–69 (cit. on p. 63).

[How97] Nick Howgrave-Graham. “Finding Small Roots of Univariate Modular Equations Revisited”. In: *6th IMA International Conference on Cryptography and Coding.* Ed. by Michael Darnell. Vol. 1355. Lecture Notes in Computer Science. Cirencester, UK: Springer, Heidelberg, Germany, Dec. 1997, pp. 131–142 (cit. on pp. 63, 67).

[HS01] Nick Howgrave-Graham and Nigel P. Smart. “Lattice Attacks on Digital Signature Schemes”. In: *Des. Codes Cryptography* 23.3 (2001), pp. 283–290 (cit. on p. 14).

[HS05] F. Hess and I. E. Shparlinski. “On the linear complexity and multidimensional distribution of congruential generators over elliptic curves.” In: *Des. Codes Cryptography* 35 (2005), pp. 111–117 (cit. on p. 78).

[IK04] H. Iwaniec and E. Kowalski. *Analytic number theory.* Amer. Math.Soc., Providence, RI, 2004 (cit. on p. 53).

[JL99] L.A. Levin J. Hastad R. Impagliazzo and M. Luby. "A Pseudorandom Generator from any One-way Function". In: *SIAM Journal on Computing* 28.4 (1999), pp. 1364–1396 (cit. on p. 6).

[JM06] Ellen Jochemsz and Alexander May. "A Strategy for Finding Roots of Multivariate Polynomials with New Applications in Attacking RSA Variants". In: *Advances in Cryptology – ASIACRYPT 2006*. Ed. by Xuejia Lai and Kefei Chen. Vol. 4284. Lecture Notes in Computer Science. Shanghai, China: Springer, Heidelberg, Germany, Dec. 2006, pp. 267–282 (cit. on pp. 63, 64).

[JS98] A. Joux and J. Stern. "Lattice reduction: A toolbox for the cryptanalyst". In: *J. Cryptology* 11 (1998), pp. 161–185 (cit. on p. 8).

[Jut98] Charanjit S. Jutla. "On Finding Small Solutions of Modular Multivariate Polynomial Equations". In: *Advances in Cryptology – EUROCRYPT'98*. Ed. by Kaisa Nyberg. Vol. 1403. Lecture Notes in Computer Science. Espoo, Finland: Springer, Heidelberg, Germany, May 1998, pp. 158–170 (cit. on p. 65).

[JV12] Antoine Joux and Vanessa Vitse. "Cover and Decomposition Index Calculus on Elliptic Curves Made Practical - Application to a Previously Unreachable Curve over $\mathbb{F}_{p^6}$". In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. Lecture Notes in Computer Science. Cambridge, UK: Springer, Heidelberg, Germany, Apr. 2012, pp. 9–26 (cit. on p. 22).

[KAF+10] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Joppe W. Bos, Pierrick Gaudry, Alexander Kruppa, Peter L. Montgomery, Dag Arne Osvik, Herman J. J. te Riele, Andrey Timofeev, and Paul Zimmermann. "Factorization of a 768-Bit RSA Modulus". In: *Advances in Cryptology – CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 2010, pp. 333–350 (cit. on p. 4).

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. "Differential Power Analysis". In: *Advances in Cryptology – CRYPTO'99*. Ed. by Michael J. Wiener. Vol. 1666. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 1999, pp. 388–397 (cit. on p. 13).

[Knu85] D. E. Knuth. "Deciphering a linear congruential encryption". In: *IEEE Trans. Inf.Theory* 31 (1985), pp. 49–52 (cit. on p. 8).

[Koc96] Paul C. Kocher. "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems". In: *Advances in Cryptology – CRYPTO'96*. Ed. by Neal Koblitz. Vol. 1109. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 1996, pp. 104–113 (cit. on p. 13).

[KP10] Eike Kiltz and Krzysztof Pietrzak. "Leakage Resilient ElGamal Encryption". In: *Advances in Cryptology – ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. Lecture Notes in Computer Science. Singapore: Springer, Heidelberg, Germany, Dec. 2010, pp. 595–612 (cit. on p. 106).

[KW04] Eike Kiltz and Arne Winterhof. “On the interpolation of bivariate polynomials related to Diffie-Hellman mapping.” In: *Bull. Austral. Math. Soc.* 69 (2004), pp. 305–315 (cit. on pp. 12, 18, 28, 32).

[KW06] Eike Kiltz and Arne Winterhof. “Polynomial interpolation of cryptographic functions related to Diffie-Hellman and discrete logarithm problem.” In: *Discrete Appl. Math.* 154.2 (2006), pp. 326–336. ISSN: 0166-218X (cit. on p. 32).

[LLL82] Arjen K. Lenstra, Hendrik W. Jr. Lenstra, and László Lovász. “Factoring polynomials with rational coefficients.” In: *Math. Ann.* 261 (1982), pp. 515–534. ISSN: 0025-5831; 1432-1807/e (cit. on p. 66).

[LS05] Tanja Lange and Igor E. Shparlinskii. “Certain Exponential Sums and Random Walks on Elliptic Curves”. In: *Canad. J. Math.* 57.42 (2005), pp. 338–350 (cit. on p. 9).

[LS86] M. Blum L. Blum and M. Shub. “A Simple Unpredictable Pseudo-Random Number Generator”. In: *SIAM Journal on Computing* 15.2 (1986), pp. 364–383 (cit. on p. 6).

[LSSW12] San Ling, Igor E. Shparlinski, Ron Steinfeld, and Huaxiong Wang. “On the modular inversion hidden number problem”. In: *J. Symb. Comput.* 47.4 (2012), pp. 358–367 (cit. on p. 102).

[LSW14] San Ling, Igor E. Shparlinski, and Huaxiong Wang. “On the Multidimensional Distribution of the Naor-Reingold Pseudo-Random Function”. In: *Math. Comput.* 83.289 (2014), pp. 2429–2434 (cit. on pp. 11, 29).

[LW02] Tanja Lange and Arne Winterhof. “Polynomial Interpolation of the Elliptic Curve and XTR Discrete Logarithm”. In: *Computing and Combinatorics, 8th Annual International Conference, COCOON 2002, Singapore, August 15-17, 2002, Proceedings*. Ed. by Oscar H. Ibarra and Louxin Zhang. Vol. 2387. Lecture Notes in Computer Science. Springer, 2002, pp. 137–143. ISBN: 3-540-43996-X (cit. on pp. 18, 32).

[LW03a] Tanja Lange and Arne Winterhof. “Interpolation of the discrete logarithm in $\mathbb{F}_q$ by Boolean functions and by polynomials in several variables modulo a divisor of $q-1$.” In: *Discrete Appl. Math.* 128.1 (2003), pp. 193–206. ISSN: 0166-218X (cit. on p. 32).

[LW03b] Tanja Lange and Arne Winterhof. “Interpolation of the Elliptic Curve Diffie-Hellman Mapping”. In: *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 15th International Symposium, AAECC-15, Toulouse, France, May 12-16, 2003, Proceedings*. Ed. by Marc P. C. Fossorier, Tom Høholdt, and Alain Poli. Vol. 2643. Lecture Notes in Computer Science. Springer, 2003, pp. 51–60 (cit. on p. 45).

[M R10] M. Ritzenhofen. *On efficiently calculating small solutions of systems of polynomial equations: lattice-based methods and applications to cryptography*. Ph.D. thesis, Ruhr University Bochum, http://www-brs.ub.ruhr-uni-bochum., 2010 (cit. on p. 66).

[MD86] G. L. Mullen and D.White. “A polynomial representation for logarithms in GF(q)”. In: *Acta Arith* 47 (1986), pp. 255–261 (cit. on p. 26).

[Mef16] Thierry Mefenza. "Inferring Sequences Produced by a Linear Congruential on Elliptic Curves using Coppersmith's Methods". In: *Computing and Combinatorics - 22nd International Conference, COCOON 2016 Ho Chi Minh City, Vietnam, August 2-4, 2016 Proceeding*. Vol. 9797. Lecture Notes in Computer Science. Springer, 2016 (cit. on p. 13).

[MHMP13] Elke De Mulder, Michael Hutter, Mark E. Marson, and Peter Pearson. "Using Bleichenbacher's Solution to the Hidden Number Problem to Attack Nonce Leaks in 384-Bit ECDSA". In: *Cryptographic Hardware and Embedded Systems – CHES 2013*. Ed. by Guido Bertoni and Jean-Sébastien Coron. Vol. 8086. Lecture Notes in Computer Science. Santa Barbara, California, US: Springer, Heidelberg, Germany, Aug. 2013, pp. 435–452. DOI: 10.1007/978-3-642-40349-1_25 (cit. on p. 14).

[Mil86] Victor S. Miller. "Use of Elliptic Curves in Cryptography". In: *Advances in Cryptology – CRYPTO'85*. Ed. by Hugh C. Williams. Vol. 218. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 1986, pp. 417–426 (cit. on p. 2).

[MS01] Edwin El Mahassni and Igor Shparlinski. "Polynomial representations of the Diffie-Hellman mapping." In: *Bull. Austral. Math. Soc.* 63 (2001), pp. 467–473 (cit. on pp. 12, 28).

[MS02] E. Mahassni and I. E. Shparlinski. "On the uniformity of distribution of congruential generators over elliptic curves." In: *intern. conf. on sequences and their applications, Bergen 2001*. Springer-Verlag, London, 2002, pp. 257–264 (cit. on p. 78).

[MV16] Thierry Mefenza and Damien Vergnaud. "Distribution and Polynomial Interpolation of the Dodis-Yampolskiy Pseudo-Random Function". In: *6th International Workshop, WAIFI 2016, Ghent, Belgium, July 13-15, 2016*. Vol. 10064. Lecture Notes in Computer Science. Springer, 2016 (cit. on p. 12).

[MV17a] Thierry Mefenza and Damien Vergnaud. "Lattice Attacks on Pairing-Based Signatures". In: *16th IMA International Conference on Cryptography and Coding, IMACC 2017*. Vol. to appear. Lecture Notes in Computer Science. Springer, 2017 (cit. on p. 14).

[MV17b] Thierry Mefenza and Damien Vergnaud. "Polynomial Approximation of the Generalized Diffie- Hellman and Naor-Reingold Functions". In: *preprint* (2017) (cit. on p. 12).

[MV17c] Thierry Mefenza and Damien Vergnaud. "Polynomial interpolation of the Naor-Reingold pseudo-random function". In: *Applicable Algebra in Engineering, Communication and Computing* 28.3 (2017), pp. 237–255 (cit. on p. 12).

[MW08] Gerasimos C. Meletiou and Arne Winterhof. "Interpolation of the Double Discrete Logarithm". In: *Arithmetic of Finite Fields, 2nd International Workshop, WAIFI 2008, Siena, Italy, July 6-9, 2008, Proceedings*. Ed. by Joachim von zur Gathen, José Luis Imaña, and Çetin Kaya Koç. Vol. 5130. Lecture Notes in Computer Science. Springer, 2008, pp. 1–10 (cit. on p. 32).

[N K87] N. Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of Computation*. 48.177 (1987), pp. 203–209 (cit. on p. 2).

[NR04] Moni Naor and Omer Reingold. “Number-theoretic constructions of efficient pseudo-random functions”. In: *J. ACM* 51.2 (2004), pp. 231–262 (cit. on pp. 10, 11, 29, 31).

[NR97] Moni Naor and Omer Reingold. “Number-theoretic Constructions of Efficient Pseudo-random Functions”. In: *38th Annual Symposium on Foundations of Computer Science*. Miami Beach, Florida: IEEE Computer Society Press, Oct. 1997, pp. 458–467 (cit. on pp. 10, 29, 31).

[NS02] Phong Q. Nguyen and Igor Shparlinski. “The Insecurity of the Digital Signature Algorithm with Partially Known Nonces”. In: *Journal of Cryptology* 15.3 (2002), pp. 151–176 (cit. on pp. 14, 117).

[NS03] Phong Q. Nguyen and Igor E. Shparlinski. “The Insecurity of the Elliptic Curve Digital Signature Algorithm with Partially Known Nonces”. In: *Des. Codes Cryptography* 30.2 (2003), pp. 201–217 (cit. on pp. 14, 117).

[NW01] Harald Niederreiter and Arne Winterhof. “Incomplete exponential sums over finite fields and their applications to new inversive pseudorandom number generators”. In: *Acta. Arith. Part 4* (2001), pp. 387–399 (cit. on p. 23).

[OS11] Alina Ostafe and Igor E. Shparlinski. “Twisted exponential sums over points of elliptic curves.” English. In: *Acta Arith.* 148.1 (2011), pp. 77–92. ISSN: 0065-1036; 1730-6264/e. DOI: 10.4064/aa148-1-6 (cit. on pp. 22, 25).

[PJ02] P.Beelen and J.Doumen. “Pseudorandom sequences from elliptic curves. Finite fields with applications to coding theory.” In: *Cryptography and related areas. Springer-Verlag, Berlin,* (2002), pp. 37–52 (cit. on p. 78).

[RW06] Josef Pieprzyk Ron Steinfeld and Huaxiong Wang. “On the provable security of an efficient RSA-based pseudorandom generator”. In: *In Xuejia Lai and Kefei Chen, editors, ASIACRYPT*. Vol. 4284. Lecture Notes in Computer Science. Springer, 2006, pp. 48–63 (cit. on p. 8).

[S H94] S. Hallgren. “Linear congruential generators over elliptic curves.” In: *Preprint CS-94-143, Dept. of Comp. Sci.,* (1994) (cit. on pp. 8, 78).

[Sch90] Claus-Peter Schnorr. “Efficient Identification and Signatures for Smart Cards”. In: *Advances in Cryptology – CRYPTO’89*. Ed. by Gilles Brassard. Vol. 435. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 1990, pp. 239–252 (cit. on p. 3).

[Sem04] Igor Semaev. *Summation polynomials and the discrete logarithm problem on elliptic curves*. Cryptology ePrint Archive, Report 2004/031. http://eprint.iacr.org/2004/031. 2004 (cit. on p. 22).

[Sem15] Igor Semaev. *New algorithm for the discrete logarithm problem on elliptic curves*. Cryptology ePrint Archive, Report 2015/310. http://eprint.iacr.org/2015/310. 2015 (cit. on p. 22).

[Sho97] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM J. Comput.* 26.5 (1997), pp. 1484–1509 (cit. on p. 4).

[Shp00a] Igor E. Shparlinski. “Linear complexity of the Naor-Reingold pseudo-random function”. In: *Inf. Process. Lett.* 76.3 (2000), pp. 95–99 (cit. on pp. 11, 29).

[Shp00b] Igor E. Shparlinski. "On the Naor-Reingold Pseudo-Random Function from Elliptic Curves". In: *Appl. Algebra Eng. Commun. Comput.* 11.1 (2000), pp. 27–34 (cit. on pp. 11, 29).

[Shp03] I.E. Shparlinski. *Cryptographic applications of analytic number theory. Complexity lower bounds and pseudorandomness* . Birkhauser Verlag, Basel, 2003 (cit. on pp. 12, 28, 32).

[Shp08] I. E. Shparlinski. "Pseudorandom points on elliptic curves over finite fields". In: *Series on Number Theory and Its Applications* 5 (2008). Algebraic Geometry and Its Applications, pp. 116–134 (cit. on p. 78).

[Shp09a] Igor E. Shparlinski. "Exponential sums with consecutive modular roots of an integer". In: *Quart. J. Math* (2009), pp. 1–7 (cit. on pp. 22, 23).

[Shp09b] Igor E. Shparlinski. "Pseudorandom number generators from elliptic curves." In: *Recent trends in cryptography. UIMP-RSME Santaló summer school, July 11–15, 2005, Universidad Internacional Menéndez Pelayo, Santander, Spain.* Providence, RI: American Mathematical Society (AMS); Madrid: Real Sociedad Matemática Española, 2009, pp. 121–141. ISBN: 978-0-8218-3984-3/pbk (cit. on p. 78).

[SK03] Ryuichi Sakai and Masao Kasahara. *ID based Cryptosystems with Pairing on Elliptic Curve*. Cryptology ePrint Archive, Report 2003/054. http://eprint.iacr.org/2003/054. 2003 (cit. on pp. 3, 4, 14, 101, 111).

[SS01] Igor E. Shparlinski and Joseph H. Silverman. "On the Linear Complexity of the Naor-Reingold Pseudo-random Function from Elliptic Curves". In: *Des. Codes Cryptography* 24.3 (2001), pp. 279–289 (cit. on pp. 11, 29).

[SS06] Jaime Gutierrez Simon R. Blackburn Domingo Gomez-Perez and Igor Shparlinski. "Reconstructing noisy polynomial evaluation in residue rings". In: *J. Algorithms* 61.2 (2006), pp. 47–59 (cit. on p. 8).

[Was08] Lawrence C. Washington. *Elliptic curves. Number theory and cryptography. 2nd ed.* 2nd ed. Boca Raton, FL: Chapman and Hall/CRC, 2008, pp. xviii + 513. ISBN: 978-1-4200-7146-7/hbk (cit. on pp. 19, 20).

[Wei48] André Weil. "On some exponential sums". In: *Proc. Nat. Acad. Sci. U.S.A* 34 (1948), pp. 204–207 (cit. on p. 23).

[Win01] A. Winterhof. "A note on (the interpolation of the Diffie-Hellman Mapping." In: *Bull.Austral.Math.Soc* 64 (2001), pp. 475–477 (cit. on pp. 12, 28).

[WS99] D. Lieman W. D. Banks F. Griffin and I. E. Shparlinski, eds. *Non-linear Complexity of the Naor-Reingold Pseudo-random Function, Proc. 2nd Intern. Conf. on Information and Communication Security, ICICS1999, Sydney*. Vol. 1726. Lecture Notes in Computer Science. Springe-Verlag, Berlin, 1999 (cit. on p. 32).

[ZSS04] Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. "An Efficient Signature Scheme from Bilinear Pairings and Its Applications". In: *PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography*. Ed. by Feng Bao, Robert Deng, and Jianying Zhou. Vol. 2947. Lecture Notes in Computer Science. Singapore: Springer, Heidelberg, Germany, Mar. 2004, pp. 277–290 (cit. on pp. 101, 111).

Résumé

L'aléatoire est un ingrédient clé en cryptographie. Par exemple, les nombres aléatoires sont utilisés pour générer des clés, pour le chiffrement et pour produire des nonces. Ces nombres sont générés par des générateurs pseudo-aléatoires et des fonctions pseudo-aléatoires dont les constructions sont basées sur des problèmes qui sont supposés difficiles. Dans cette thèse, nous étudions certaines mesures de complexité des fonctions pseudo-aléatoires de Naor-Reingold et Dodis-Yampolskiy et étudions la sécurité de certains générateurs pseudo-aléatoires (le générateur linéaire congruentiel et le générateur puissance basés sur les courbes elliptiques) et de certaines signatures à base de couplage basées sur le paradigme d'inversion.

Nous montrons que la fonction pseudo-aléatoire de Dodis-Yampolskiy est uniformément distribué et qu'un polynôme multivarié de petit dégré ou de petit poids ne peut pas interpoler les fonctions pseudo-aléatoires de Naor-Reingold et de Dodis-Yampolskiy définies sur un corps fini ou une courbe elliptique. Le contraire serait désastreux car un tel polynôme casserait la sécurité de ces fonctions et des problèmes sur lesquels elles sont basées. Nous montrons aussi que le générateur linéaire congruentiel et le générateur puissance basés sur les courbes elliptiques sont prédictibles si trop de bits sont sortis à chaque itération.

Les implémentations pratiques de cryptosystèmes souffrent souvent de fuites critiques d'informations à travers des attaques par canaux cachés. Ceci peut être le cas lors du calcul de l'exponentiation afin de calculer la sortie de la fonction pseudo-aléatoire de Dodis-Yampolskiy et plus généralement le calcul des signatures dans certains schémas de signatures bien connus à base de couplage (signatures de Sakai-Kasahara, Boneh-Boyen et Gentry) basées sur le paradigme d'inversion. Nous présentons des algorithmes (heuristiques) en temps polynomial à base des réseaux qui retrouvent le secret de celui qui signe le message dans ces trois schémas de signatures lorsque plusieurs messages sont signés sous l'hypothèse que des blocs consécutifs de bits des exposants sont connus de l'adversaire.

Mots Clés

fonctions pseudo-aléatoires, générateurs pseudo-aléatoires, signature à base de couplage, discrépance, interpolation polynomiale, attaques à base de réseaux.

Abstract

Randomness is a key ingredient in cryptography. For instance, random numbers are used to generate keys, for encryption and to produce nonces. They are generated by pseudo-random generators and pseudo-random functions whose constructions are based on problems which are assumed to be difficult. In this thesis, we study some complexity measures of the Naor-Reingold and Dodis-Yampolskiy pseudo-random functions and study the security of some pseudo-random generators (the linear congruential generator and the power generator on elliptic curves) and some pairing-based signatures based on *exponent-inversion* framework.

We show that the Dodis-Yampolskiy pseudo-random functions is uniformly distributed and that a low-degree or low-weight multivariate polynomial cannot interpolate the Naor-Reingold and Dodis-Yampolskiy pseudo-random functions over finite fields and over elliptic curves. The contrary would be disastrous since it would break the security of these functions and of problems on which they are based. We also show that the linear congruential generator and the power generator on elliptic curves are insecure if too many bits are output at each iteration.

Practical implementations of cryptosystems often suffer from critical information leakage through side-channels. This can be the case when computing the exponentiation in order to compute the output of the Dodis-Yampolskiy pseudo-random function and more generally in well-known pairing-based signatures (Sakai-Kasahara signatures, Boneh-Boyen signatures and Gentry signatures) based on the *exponent-inversion* framework. We present lattice-based polynomial-time (heuristic) algorithms that recover the signer's secret in the pairing-based signatures when used to sign several messages under the assumption that blocks of consecutive bits of the exponents are known by the attacker.

Keywords

pseudo-random functions, pseudo-random generators, pairing-based signatures, discrepancy, polynomial interpolation, lattice attacks.