Algorithmic techniques in algebraic cryptanalysis and their applications

Makarim, R.H.

Part I
Introduction and Overview of the Results

1 Cryptography and Cryptanalysis

Contents

1.1 Cryptography

Communication is one of the most basic human activities, enabling two or more parties to exchange messages through various forms of media. Traditionally, communication could only be achieved within a very limited distance by means of meeting or gathering in one place. When the concept of writing was invented, especially with portable mediums such as animal skins, bones, papyrus, etc, information could reach longer distances by writing the message on a suitable material and send it to the destination using a courier. If the message contains sensitive information, the courier is assumed to be a trusted person that would not leak it to malicious parties. In order to liberate the sender and receiver from such assumptions, they need to devise techniques that guarantee the secrecy of their message.

Classically, cryptography is the practice and study of techniques for secure information processing in the presence of adversaries. This definition is relevant when it comes to the study of encryption schemes and digital signatures, both are used to ensure secrecy and authenticity of a communication channel. Modern cryptography, on the other hand, has a broader context. It is a study how to obtain cryptographic functionalities such as confidentiality, integrity, entity authentication, and data origin authentication [MvOV96], typically using mathematical techniques related to various aspects of information security. For instance, confidentiality is achieved using encryption primitives that protect the data by converting it into an incomprehensible form that can be recovered only by an authorized entity; integrity is ensured through mechanisms that detect any unauthorized modification of information; entity authentication verifies the identity of a communicating party and is commonly achieved using challenge-response protocols, digital certificates, or public-key infrastructures; data origin authentication ensures that a received message originates from the claimed source using message authentication codes or digital signatures. A further example includes the study of secure computation, which allows arbitrary computation to be distributed among multiple parties such that the computations remain confidential and performed correctly even if some parties involved behave maliciously.

Encryption is a function that transforms a message (plaintext) into its corresponding incomprehensible form of information (ciphertext). The correspondence between a plaintext and a ciphertext is necessary in order for the intended

recipient to recover the original plaintext using a *decryption* function. To provide security against adversaries, the encryption and decryption function of an encryption scheme require an additional secret information as input, which should only be available to the designated parties. We refer to this secret input as *key*. A classical construction of an encryption scheme is done storing every possible plaintext and their corresponding ciphertext in a look-up table. Another well-known example is Caesar cipher (named after Julius Caesar), that takes a secret number N as a key and the encryption function substitutes each letter in a message by the next N-th letter in a cyclic and alphabetical ordering (i.e., with $N = 3$, A is substituted by D, B by E, ..., W by Z, X by A, Y by B, and Z by C).

The encryption and decryption functions of an encryption scheme should be invertible and efficiently computable, provided that the key is known. In principle the security of an encryption scheme must only rely on the secrecy of the key, not the secrecy of the scheme itself. This is known as the *Kerckhoffs' principle* [Ker83]. The size of a key should be small enough to ensure low storage distribution cost. However, given a plaintext and its corresponding ciphertext, the number of all possible keys should be large enough such that no real-world adversary is able to exhaustively try all possible keys. The usage of keys is also beneficial whenever an adversary succeeds in revealing some information about the key. Instead of redesigning the entire scheme, the legitimate parties can simply replace the previous keys with the new ones.

Encryption schemes can be classified according to the relation between the encryption key and decryption key. There are two main categories. The first category is *symmetric-key* (or secret-key) cryptography. The name comes from the fact that the same key is used to perform encryption and decryption. Thus, the key must not be accessible by adversaries. The second category is *asymmetric-key* (or public-key) cryptography. The idea is that, given an encryption key, it should be computationally infeasible for an adversary to obtain the decryption key. This allows the knowledge of the encryption key to be known publicly. The idea of public-key cryptography was formally introduced in [Mer78, DH76].

There are a number of approaches to construct symmetric-key encryption schemes. One particular class of symmetric-key cryptography are *block ciphers*. As the name suggests, a block cipher operates on a fixed-length block of plaintext. They are widely used on applications that require encryption on a bulk of data. Modern block ciphers typically employ an iterated function called *round function*. It is constructed as a composition of a nonlinear function together with a linear function and key addition operation. The goal of employing a nonlinear function is to make the ciphertext statistics depend on the plaintext statistics in a way that makes it difficult to be exploited by an adversary. This design principle is known as *confusion*. A linear function and key addition are used so that each bit of the plaintext and each bit of the key influences many bits of the ciphertext. This design principle is known as *diffusion*. Both principles were first introduced by Shannon [Sha49]. Some notable examples of modern block ciphers are Data Encryption Standard (DES) [Nat77] and Advanced Encryption Standard (AES) [NIS01]. In order to encrypt an arbitrary length plaintext using a block cipher, one uses modes of operation, i.e., it is an algorithm that specifies how a block cipher should be applied to achieve this. Some examples of block cipher modes of operation include Electronic Codebook (ECB), Cipher Block Chaining (CBC), and Cipher Feedback (CFB) (see [Pre11]).

Another class of symmetric-key cryptography are *stream ciphers*, where encryption and decryption functions operate on one bit at a time. They are preferred over block ciphers in devices with limited memory and each plaintext bit must be processed individually. An example of a simple, yet secure, stream cipher is the *one time pad*. Given a sequence of bits $s_1, s_2, \ldots \in \mathbb{F}_2$ called *keystream*, the one-time pad encrypts a sequence of plaintext bits $p_1, p_2, \ldots \in \mathbb{F}_2$ into a sequence of ciphertext bits $c_1, c_2, \ldots \in \mathbb{F}_2$ as $c_i = p_i + s_i$ for $i \geq 1$. It achieves perfect secrecy in the sense that a ciphertext does not reveal any information about the plaintext provided the same keystream is not used twice to encrypt different plaintext. One drawback of the one-time pad is that the length of the keystream must be equal to the length of the plaintext. A solution for this drawback at the cost of losing perfect secrecy is to let a pseudo-random generator generate the keystream. It takes a relatively short key as its initial state together with some public parameters. The study of stream ciphers deals with techniques to construct secure pseudo-random generator that generates the keystream with low cost digital circuit operations.

One of the main building blocks of stream ciphers is a *Linear Feedback Shift Register* (LFSR). An LFSR of length ℓ over $\mathbb{F}_2$ is a regularly clocked finite state machine with internal state $(x_1, \ldots, x_\ell) \in \mathbb{F}_2^\ell$ and an update function $L(x_1, \ldots, x_\ell) = (x_2, \ldots, x_\ell, \Lambda(x_1, \ldots, x_\ell))$ where $\Lambda : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ is a linear map. At each clock, it outputs a specific entry of its internal state, followed by the update function. Let $S_t \in \mathbb{F}_2^\ell$ denote the internal state at time t. The internal state at time $t+1$ is computed as $S_{t+1} = L(S_t) = L^{t+1}(S_0)$ where S_0 denote the initial state of the LFSR. Note that L itself is a linear transformation. Thus, an LFSR can not be used directly as a keystream generator because the knowledge of the entire keystream can be efficiently recovered, provided that Λ is known and an attacker has access to χ consecutive keystream bits, where χ is the size of the smallest LFSR that generates the given sequence (also known as *linear complexity* of the sequence). A common technique to construct a stream cipher is to combine one or several LFSRs together with a nonlinear Boolean function in order to increase the complexity of attacks that exploit algebraic relations between the keystream bits and the initial state of the stream cipher.

The main problem of using symmetric-key encryption schemes for communication purposes is that the keys must be distributed to the intended recipients via a secure channel. This limitation was solved with the introduction of public-key cryptography. Recall that the encryption keys of a public-key encryption scheme can be publicly known and only the decryption keys that must be kept secret. This idea is possible due to the concept of a *trapdoor one-way function*. Informally, it is a function that is computationally feasible to compute in one direction yet it is computationally infeasible to compute its inverse unless one knows some secret information called the *trapdoor*. One prominent example of trapdoor one-way functions is RSA [RSA78], that is based on the difficulty of integer factorization. The trapdoor consists of prime factors of a sufficiently large positive composite integer. Other examples of trapdoor one-way functions include the knapsack problem [MH78], discrete logarithm problem [DH76, Gam84, Kra93], lattice problems [BDK+18, DKL+18, HPS98, PFH+] and multivariate quadratic (MQ) problem [Pat96, KPG99, PCDY17, CBD+09].

Such a trapdoor one-way function also enables a party to attest his approval of a message, analogous to the concept of signing a document in a real-world scenario. This mechanism is known as a *digital signature scheme*. A digital

signature scheme consists of three different algorithms: *key generation*, *signing*, and *verification*. The key generation algorithm is responsible for generating a private-key and its corresponding public-key. The signing algorithm produces a signature for a particular message using a private-key whereas the verification algorithm verifies the signature of a given message using a public-key.

1.2 Cryptanalysis

Cryptanalysis is the process of developing methods to attack cryptographic schemes by unveiling and exploiting their intrinsic properties. In the case of encryption schemes, the aim is to devise an algorithm that breaks one of the claimed security properties. The algorithm can be generic and applicable to all encryption schemes that belongs to a specific family, or it can be dedicated for a particular construction. Typically, an attack scenario assumes that the encryption or decryption algorithm is known to the adversaries, and the key used is chosen uniformly at random from the set of all possible keys.

There are different goals for cryptanalytic attacks. The strongest one is to extract partial or full information about the key used in an encryption. This would allow an adversary to recover any plaintext encrypted using the same key. A weaker goal of cryptanalytic attack is to learn non-random behaviours of a particular encryption scheme, known as *distinguishing attacks*. For block ciphers, distinguishing attacks try to exhibit some properties that deviates from a random permutation, while for the case of stream ciphers, it tries is to distinguish the keystream from a truly random sequence. In a digital signature scheme, an attack that tries to falsify a signature for a given message without an access the the actual signer's private key is called a *forgery attack*.

In any cryptanalytic attack, it is possible to assume that adversaries have access to various types of data (e.g., plaintext, ciphertext, etc). Not all data is equally beneficial. Thus it is crucial to classify different attack scenarios based on the data required to mount the attack. Let E_K and D_K denote encryption and decryption function under the key K.

Ciphertext-Only Attack In this model an adversary is assumed to only have access to a set of ciphertexts.

Known-Plaintext Attack In this attack model an adversary has access to a set of, say, n plaintext $p_1, p_2, \ldots, p_n$ and their corresponding ciphertext $c_1, c_2, \ldots, c_n$.

Chosen-(Plaintext/Ciphertext) Attack In a chosen-plaintext attack an adversary selects a set of plaintext $p_1, \ldots, p_n$ *a priori* and requests their corresponding ciphertext $c_i = E_K(p_i)$ for all $i = 1, 2, \ldots, n$. Conversely, in a chosen-ciphertext attack an adversary chooses a set of ciphertext $c_1, \ldots, c_n$ and requests their corresponding plaintext $p_i = D_K(c_i)$ for all $i = 1, 2, \ldots, n$.

Adaptively Chosen-(Plaintext/Ciphertext) Attack In an adaptively chosen-plaintext attack, an adversary chooses plaintexts for encryption based on previously selected plaintexts and the corresponding ciphertexts. In an adaptively chosen-ciphertext attack the plaintext and encryption oracle in the adaptively chosen-plaintext scenario are replaced by a ciphertext and decryption oracle, respectively.

Ciphertext-only and known-plaintext attacks are classified as *passive* attacks, i.e., the adversary only observes communications or data. The former is considered to be the most realistic attack scenario but it is also the most difficult one due to limited information available to the adversary. Thus, to recover partial or full-information about the key in a ciphertext-only attack, additional knowledge about the plaintext is often needed, such as some redundancy that occurs in the plaintext or knowledge of a valid plaintext.

On the other hand, chosen-(plaintext/ciphertext) and adaptively chosen-(plaintext/ciphertext) attacks are *active* attacks. These models assume that the adversary has black-box access to the encryption (or decryption) oracle with the user's key, and therefore viewed as somewhat less realistic in practice. In public-key cryptography, however, the encryption oracle comes for free since the encryption key is public.

As we move down the list above, the requirements of the attack model become increasingly difficult. This can be seen from the escalation in the type of data required and the capability of the adversary. In order to compare different types of attack, it is necessary to quantify their complexity in terms of all involved parameters. The cost metric for a cryptanalytic attack consists of the following parameters:

Time complexity In general, the time required to mount the attack is considered the most important cost metric in cryptanalysis. In symmetric-key cryptanalysis, it is typically quantitatively measured relative to the cost of an encryption or decryption call. In public-key cryptanalysis, the time complexity is typically measured in terms of the number of bit operations.

Data complexity Every attack model previously described requires certain information to be available. Not only the type of information, the amount must also be taken into account as a cost metric for cryptanalytic attacks. There is also a possible trade-off between data complexity and time complexity. For instance an attack might need relatively few encryption or decryption operations but it requires enormous amounts of plaintext or ciphertext pairs, or *vice versa*.

Memory complexity The memory complexity is a metric that represents the storage requirements to mount the attack. High memory complexities may turn out to be the main bottleneck that causes an attack to be impractical. It is also considered to be an expensive cost from practical point of view. In the case where there is a trade-off between time complexity and memory complexity, it is often more reasonable to reduce the memory complexity at the expense of increasing time complexity.

Success probability In all attack settings, it is plausible to include success probability as a cost metric to mount an attack. For instance, the adversary may reduce the data complexity at the expense of a smaller success probability. However, the probability should still be sufficiently high to be considered as a valid attack.

The most straightforward cryptanalysis technique that recovers the key is *exhaustive key search*. The goal is to recover the key used to encrypt a particular ciphertext, which will allow an adversary to recover the corresponding plaintext and any other ciphertexts encrypted with the same key. It requires only one

ciphertext and sufficient knowledge about the plaintext to eliminate wrong keys during the searching process. Exhaustive key search works for all encryption schemes independent from their specification. Thus, it constitutes as a baseline to determine whether a specific algorithm can be considered as a structural attack. More formally, an algorithm is classified as a break of an encryption scheme if its time complexity is strictly less than exhaustive-key search. Note that it does not necessarily mean that a scheme that is broken in such sense is considered to be broken in practice.

One of an early known documented cryptanalysis technique in history is attributed to *Al-Kindi* (Alkindus) [Sin00, MAAT02][vzG15, Section G.1]. His treatise, dated back in the 9-th century, described a method to recover the plaintext from a ciphertext encrypted using *monoalphabetic substitution cipher*, i.e., it is an encryption scheme that substitutes each plaintext character into another character/symbols and the rule of the substitution is unknown to adversaries. The only requirement of the attack is that the original language used in the communication is known. The main idea of the attack is the occurrences of a letter or combination of letters for a particular language varies. For instance, the letters 'E', 'T', and 'A' are the three most-frequent letters that appear in English language while occurrence of 'X', 'Q', or 'Z' is rare. An adversary then counts the frequency of each symbol in the ciphertext and associates each of them with a corresponding letter of the language used such that the occurrence falls within the same range. This can also be combined with more sophisticated technique such as counting a pair of letters (*bigram*) or triplet of letters (*trigram*). This technique, which is statistical in nature, is known as *frequency distribution analysis*.

Many of the cryptanalysis approaches in modern cryptography, especially in the domain of symmetric-key cryptanalysis, are also based on statistical techniques. For instance, *linear cryptanalysis* [Mat93] and *differential cryptanalysis* [BS90] are the two most-well known cryptanalysis techniques for block ciphers exploiting non-random statistical behaviour of the encryption function. In linear cryptanalysis, an adversary tries to identify linear relations among parity bits of the plaintext, the ciphertext, and the key* that holds with high or low probability. On the other hand, differential cryptanalysis studies how the relation between two plaintexts affects the probabilistic occurrence of a certain relation in the corresponding ciphertexts. Both attacks are also applicable to stream ciphers [Gol94, BD07]. Another type of statistical attack on stream ciphers consists of *correlation attacks* [Sie85]. It is applicable to any stream cipher where the keystream is generated by combining the output of several LFSRs using a nonlinear function. A correlation attack exploits possible statistical dependence between the keystream bits and the output of a single LFSR. A more efficient variant of it is described in [MS89a].

One of the main drawbacks of statistical cryptanalysis is that it often requires large amounts of plaintexts or ciphertexts, making it less realistic in practice to mount the attack. Moreover, some statistical assumptions are required in order to simplify the analysis. For instance, the assumption of independent round keys is often stated in an attack against block ciphers based on linear or differential cryptanalysis. This assumption does not hold true in general, but

* Concretely, the relation is of the form $\sum_{i=1}^{n}(\alpha_i P_i + \beta_i C_i) = \sum_{j=1}^{k} \gamma_j K_j$ where $\alpha_i, \beta_i, \gamma_j \in \mathbb{F}_2$ and P_i, C_i, K_j are the plaintext bits, the ciphertext bits, and the key bits respectively

often is an useful heuristic.

Another approach in cryptanalysis is *algebraic cryptanalysis* or *algebraic attack* [Bar09]. From a high-level point-of-view, algebraic cryptanalysis consists mainly of two steps. It first converts the encryption or decryption function of an encryption scheme into a system of multivariate equations with coefficients in some ring or field such as $\mathbb{F}_2$. The indeterminates of the system of equations represent the internal state of the encryption function and the key, both unknown to the adversaries. The second step of the attack is to solve the constructed system of equations for a given plaintext and its corresponding ciphertext. Solving the system of equations means recovering the key as well as the internal state of the cipher. Note that for a high-degree system of equations, a common strategy to reduce the degree is by substituting all quadratic terms with auxiliary variables. For instance, a term $x_i x_j$ is substituted by $y_{i,j}$ and the new equation $x_i x_j - y_{i,j} = 0$ is added to the system of equations [KS99, CKPS00]. By iteratively applying such strategy, one can always reduce the degree of a system of equations to a quadratic one at the expense of having more variables and equations compared to the initial system of equations.

Algebraic cryptanalysis serves as a complementary technique to statistical cryptanalysis. Its primary advantage is that it requires only few plaintext-ciphertext pairs to mount the attack. In contrast to the probabilistic nature of statistical cryptanalysis, algebraic cryptanalysis is a deterministic technique and the success probability of the attack is always guaranteed since it requires no *a priori* assumptions.

However, one drawback of algebraic cryptanalysis is that it is computationally hard to solve a system of quadratic polynomial equations over a finite field, even in the case of binary field $\mathbb{F}_2$. Solving a quadratic system of equations is a well-known hard problem, known as the *Multivariate Quadratic (MQ) problem*. The computational intractability of the MQ problem over a finite field is also used to construct trapdoor functions for public-key cryptography. We refer to the family of such construction as *multivariate public-key cryptography* (MPKC), which also includes digital signature schemes based on the hardness of MQ problem. The primary challenge in MPKC is to construct a public-key or digital signature scheme such that the complexity to solve the system of equations representing the public-key is as close as possible to the complexity of solving random quadratic polynomial equations.

MPKC is one of the mathematical platforms considered for post-quantum cryptography. The US National Institute for Standards and Technology (NIST) has initiated a competition for a post-quantum cryptography standardization process. The goal is to develop classical cryptographic schemes that are compatible with existing network/communication protocols and considered secure against attacks by both quantum and classical computers. The security of several submitted proposals are based on the hardness of the MQ problem over finite fields.

1.3 Contributions

1.3.1 Motivations

This thesis focuses on the development of Gröbner bases algorithms and their applications in the cryptanalysis of Multivariate Public-Key Cryptography (MPKC)

and in other domains. The main research question in this work is how efficient the Gröbner bases algorithms can be implemented in practice to solve a system of equations that represent the public-key of MPKC. In this section we shall discuss the motivations on why we focus on Gröbner bases algorithms and why we concentrate on their applications in the cryptanalysis of MPKC. While Gröbner bases algorithms and polynomial-solving methods are often associated with the cryptanalysis of MPKC, they are equally applicable in the analysis of symmetric-key primitives. These primitives can be expressed as systems of polynomial equations over finite fields, which allows the use of Gröbner bases algorithms to perform key-recovery attacks or to study the underlying algebraic structures under certain conditions. Chapter 2 provides an overview of these approaches.

In the view of algebraic cryptanalysis, there are two factors that influence its effectiveness to attack cryptographic primitives. The first factor is the representation of the primitive as a system of multivariate polynomial equations. An adversary needs to find an algebraic modelling that maximizes the efficiency of algorithms that solve it. For instance, when it comes to algebraic solvers such as Gröbner bases algorithms [Buc65, Fau99] or the XL algorithm [CKPS00], it is more desirable that the system contains as many linearly independent equations as possible (see Subsection 2.5.1 and Subsection 2.5.2). The efficiency of other methods may be influenced by other parameters, such as the Raddum-Semaev algorithm that takes the sparsity of the system and the order of the finite field into account (see Subsection 2.5.4). The second factor, which is arguably the most important one, is the algorithm used to solve the system of polynomial equations. The SAT-based approach is applicable for polynomials with coefficients in $\mathbb{F}_{2^e}$ where e is a positive integer (see Subsection 2.5.3 and Chapter 15 of [Bar09]). The Raddum-Semaev algorithm is applicable for sparse systems of polynomial equations defined over a finite field having small order. The MILP-based approach is so far limited only for polynomials defined over $\mathbb{F}_2$ (see Subsection 2.5.5). Gröbner bases algorithms and linearization-based algorithms remain as the most generic approaches. The latter approach arguably received the most attention in cryptographic literature. Since the introduction of the XL algorithm [CKPS00], several modified variants were proposed [CP02, CP03, Cou04]. Even after it was shown that the XL algorithm is a redundant variant of the F_4 algorithm [AFI+04], the development of XL-based algorithms still continued with the introduction of the concept of mutant and MutantXL algorithms [MMDB08, MCD+09, MDBW09]. Eventually, the MutantXL algorithm was also shown to be a redundant variant of the F_4 algorithm [ACFP12].

The primary motivation to develop a new Gröbner bases algorithm from a cryptanalysis standpoint is to shift the attention from linearization-based approaches to a family of algorithms with a well-established theory. There are several proposals of Gröbner bases algorithms dedicated for cryptanalysis purposes [FJ03, JV11]. However, their exposure are limited to high-level descriptions without much emphasis on the implementation aspects. This in contrast to the fact that the efficiency of Gröbner bases algorithms are often shown by their ability to solve computational problems in practice. For instance, the efficiency of F_4 [Fau99] was demonstrated by its implementation that solved the Cyclic

9-roots problem.* The paper, however, mentioned almost no direction on how to implement the algorithm. Moreover, two well-known implementations of the Gröbner bases algorithm [Fau10, BCP97] are provided as closed-source binaries. This thesis tries to address this problem: we develop a Gröbner bases algorithm for cryptanalysis purposes and present its high-level description together with its implementation in a unified theoretical framework.

We choose to focus on the cryptanalysis of MPKCs because of one main reason: the growing interest in their development as candidates for post-quantum cryptography. In December 2016, the US National Institute for Standards and Technology (NIST) has initiated a call for proposals for a post-quantum cryptography standardization process. The goal is to develop classical cryptographic schemes that are compatible with existing network/communication protocols and considered secure against attacks by both quantum and classical computers. The security of several submitted proposals are based on the hardness of the MQ problem over finite fields. These proposals are DME [Lue17], RAINBOW [DS05], LUOV [BP17], DUALMODEMS [FPR17a], GEMSS [FPR+17b], GUI [DCP+17], HIMQ-3 [SPK17] and MQDSS [CHR+17]. Among those proposals, RAINBOW was the one that advanced the furthest and emerged as one of the Round 3 finalists [NIS20]. However, in Crypto 2022 [Beu22] Beullens introduced a new attack on RAINBOW that reduces the security level for all the proposed parameters and was mounted in practice to recover the private key for the smallest parameter. Nonetheless, this does not eliminate the interest of having MPKC schemes in the NIST post-quantum standardization mainly due to their features of having short signature and fast verification. Both features are explicitly stated as the interests of NIST in its additional call for post-quantum digital signature schemes [NIS22] where 10 out of 40 submissions are classified under the category of multivariate signatures: 3WISE [Rod23a], DME SIGN [Ln23], HPPC [Rod23b], MAYO [BCC+23], PROV [GCF+23], QR-UOV [FIH+23], SNOVA [WCD+23], TUOV [DGG+23], UOV [BCD+23], and VOX [PCF+23].

In Section 2.2 we describe that the public-key of encryption or signature schemes based on the hardness of MQ problem consists of multivariate quadratic polynomials over a finite field. Given a ciphertext (in the case of encryption schemes) or a signature (in the case of digital signature schemes), the direct approach to perform a plaintext-recovery or signature forgery attack on MQ-based public-key and digital signature schemes is to compute a Gröbner basis of the ideal generated by the polynomials in the public key. In order to gain a better confidence in their security, it is crucial to understand the complexity of computing Gröbner bases in practice. Following the same motivation,

* Cyclic n-roots problem is the problem of finding a solution for the system of equations

$$\begin{aligned} &x_1 + x_2 + \cdots + x_n = 0,\\ &x_1x_2 + x_2x_3 + \cdots + x_{n-1}x_n + x_nx_1 = 0,\\ &x_1x_2x_3 + x_2x_3x_4 + \cdots + x_nx_1x_2 = 0,\\ &\vdots\\ &x_1x_2\cdots x_n - 1 = 0. \end{aligned}$$

This is a well-known benchmark problem in computer algebra, popularized by James Davenport [Dav87].

Yasuda *et al.* introduced an open challenge to solve concrete instances of the MQ problem [YDH+15a]. The main goal of the challenge is to understand the relation among the order of a finite field, the number of variables, and the number of equations with the hardness of solving MQ problem. The result of solved challenges may contribute in determining appropriate parameters for MQ-based public-key or digital signature schemes.

Gröbner bases and Gröbner bases algorithms are also known to have applications in other areas such as algebraic geometry [AL94, Section 2.5], coding theory [BP09, Sak98], graph-coloring [Bay82, Section 3.2], integer programming [MMR91], geometric theorem proving [Wan98], etc. In this thesis we also explore other possible new applications of Gröbner bases, particularly on problems that can be expressed as a *constraint satisfaction problem* [RN10, Chapter 6] and also in the cryptanalysis of symmetric-key primitives.

1.3.2 Technical Contributions

The contribution of this thesis is divided into two parts. The first contribution deals with the algorithmic and the implementation aspect of the Gröbner bases algorithm. The second contribution proposes several new applications of Gröbner bases in different domains. All these contributions are described in Chapter 5, Chapter 6, Chapter 7, and Chapter 8.

Part of the results in Chapter 5 and Chapter 6 have been published in the following conference proceedings

> Rusydi H. Makarim, Marc Stevens. *M4GB: An Efficient Gröbner-Basis Algorithm*. Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2017 [MS17].

Chapter 5 extends the results of [MS17] by providing a detailed rationale of the reduction strategy of the M4GB algorithm as an improvement over the reduction strategy of the F_4 algorithm [Fau99] (see Section 5.1). The same chapter also provides an example to better illustrate the M4GB reduction and its generalization that applies on term-polynomial multiplication (see Section 5.2). Chapter 6 also extends the results of [MS17] by giving an overview of the MQ challenge [YDH+15a] and the hybrid approach [BFP09, BFP12] in a greater details (see Section 6.1, Section 6.2 and Section 6.3). Moreover, Chapter 6 also provides cost estimations, in terms of CPU time and memory, of using the current implementation of the M4GB algorithm to solve other instances of the MQ challenge with higher parameters (see Section 6.5).

Part of the results in Chapter 7 have been published in the following journal

> Putranto H. Utomo, Rusydi H. Makarim. *Solving a Binary Puzzle*. Mathematics in Computer Science, Volume 11, Numbers 3-4 [UM17].

Chapter 7 extends the result of [UM17] in threefold. Firstly, it provides a new way to express the polynomial equation for one of the constraint of binary puzzles as a linear combination of the elementary symmetric polynomials over $\mathbb{F}_2$ (see Theorem 7.14 in Section 7.5). Secondly, it introduces a strategy to reduce the degree of the polynomials of binary puzzles over $\mathbb{F}_2$ at the expense of working over the rational field or the integer ring (see Section 7.6). Lastly, it provides a benchmark that compares the performance of the implementation of Gröbner

bases algorithms in Magma[BCP97], Singular[DGPS18], and PolyBori[BD09a] to solve systems of polynomial equations of binary puzzles over $\mathbb{F}_2$, the rational field, and the integer ring (see Section 7.7).

M4GB Gröbner Bases Algorithm

In Chapter 5, we propose a new variant of the Gröbner bases algorithm called M4GB algorithm. The algorithm is designed to consistently operate on and maintain polynomials in tail-reduced form with respect to the current intermediate basis. In contrast to the F_4 algorithm that computes the (reduced)-row echelon form of coefficient matrices that includes reducible terms in the tail of the corresponding polynomials, the strategy of M4GB can be seen as a way to perform similar computation on coefficient matrices that contain only non-reducible terms in the tail of each polynomial. We also introduce a new recursive reduction algorithm on polynomials of the form $\mathtt{t}f$ where $\mathtt{t}$ and f are a term and a polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ respectively. The M4GB algorithm is described in a way that unifies high-level description together with its implementation aspects. Although this is different from the usual tradition to describe a Gröbner bases algorithm, our goal here is to present it without obscuring the nature of its implementation.

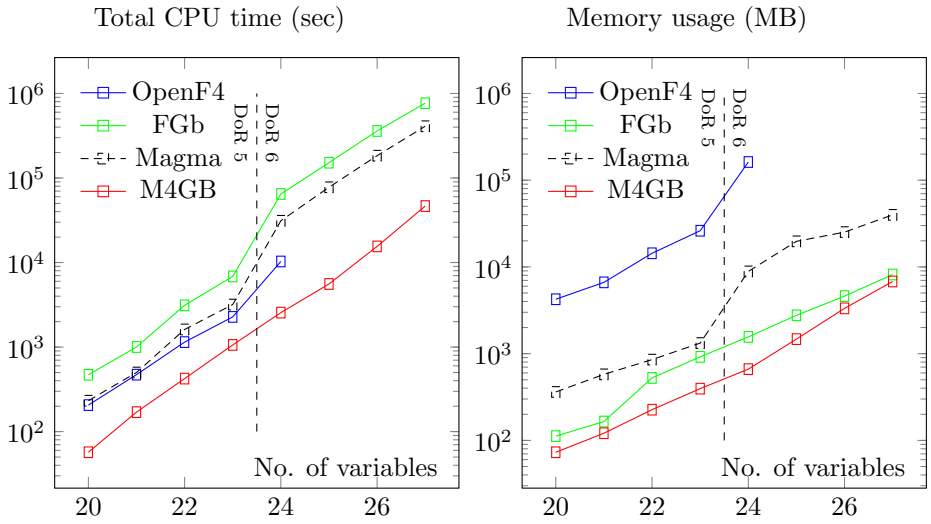

Figure 1: *The CPU time and memory comparison of* M4GB *with other Gröbner bases implementation for system of equations with* $2n$ *equations over* $\mathbb{F}_{31}$*, where* n *is the number of variables. The vertical dashed line separates the parameter with different degree of regularity (DoR).*

Following a similar approach as in [Fau99, Fau02], we demonstrate the efficiency of the M4GB algorithm by implementing it in practice and comparing its performance against other existing implementations of the Gröbner bases algorithms. The implementation is available in the following link

`https://github.com/cr-marcstevens/m4gb`.

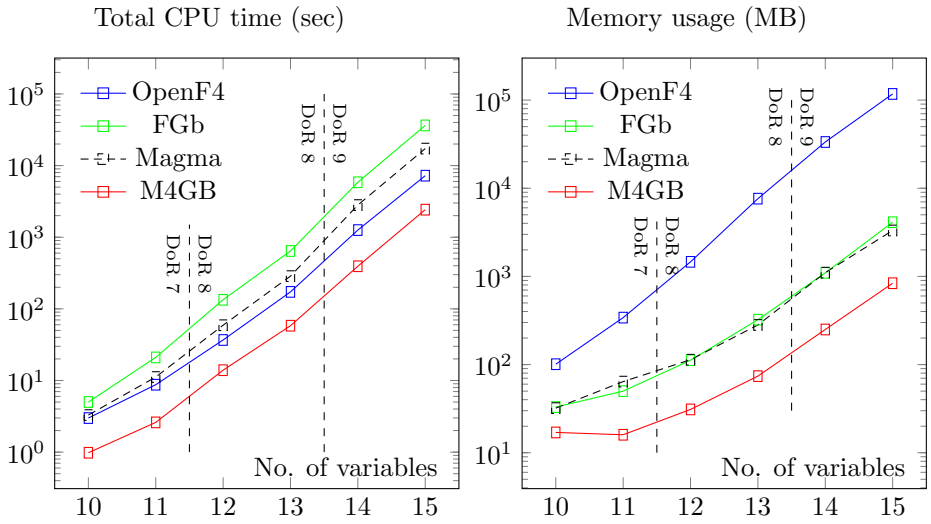

Figure 2: *The CPU time and memory comparison of* M4GB *with other Gröbner bases implementation for system of equations with* $n+1$ *equations over* $\mathbb{F}_{31}$*, where* n *is the number of variables. The vertical dashed line separates the parameter with different degree of regularity (DoR).*

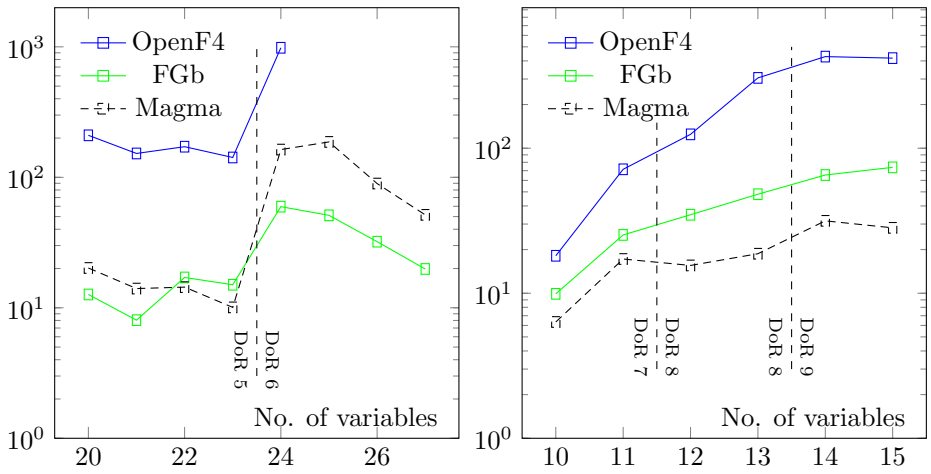

Figure 3: *The ratio of time-memory product of Magma, FGb, and OpenF4 relative to the time-memory product of* M4GB*. The left graph is the result on systems with* n *variables and* $2n$ *equations whereas the right one is the result on systems with* n *variables and* $n+1$ *equations. The vertical dashed line separates the parameter with different degree of regularity (DoR).*

We compare the implementation of our open-source M4GB with three state-of-the-art Gröbner bases implementations, namely Magma [BCP97], FGb [Fau10], and OpenF4 [CVJ]. We compare the performance and memory usage of our M4GB algorithm against existing Gröbner bases implementations for system of polynomial equations that represent a public-key of MPKC schemes, i.e., overdefined quadratic systems of equations over finite fields. The results of the benchmarks are given in Figure 1 and Figure 2. Although OpenF4, FGb, and Magma appear to provide a trade-off between CPU time and memory usage, our implementation of M4GB consistently has the best performance and the least memory usage in all benchmark parameters. For the case of $2n$ equations with n variables, M4GB performs 2.5 up to 4 times faster than OpenF4 (the second fastest implementation). M4GB also has the least memory usage by a factor between 1.2 and 2.35 compared to FGb (having the second smallest memory usage). For systems with $n+1$ equations and n variables, M4GB also emerges as the most efficient algorithm to solve such systems. It performs 2.6 up to 3.3 times faster than OpenF4 as the second fastest implementation and it uses 1.9 up to 4.4 times less memory compared to FGb and Magma.

The efficiency of the M4GB algorithm also contributes to the cryptanalysis of existing multivariate-based post-quantum digital signatures. Recently Beullens [Beu24] utilized our implementation of the M4GB algorithm to demonstrate a practical attack against SNOVA [WCD+23], one of the candidate of the NIST on-ramp post-quantum signature. The attack itself consists of multiple steps where the M4GB algorithm is used in the last one, which is also the bottleneck of the attack, that requires solving multiple systems of 14 polynomal equations in 13 variables over $\mathbb{F}_{16}$ (see page 13-15 of [Beu24]).

MQ Challenge Computational Records

Chapter 6 presents the application of M4GB as a Gröbner bases algorithm to solve some concrete challenges of MQ problems over finite fields. Particularly, M4GB set a record in the *Fukuoka MQ Challenge* for parameters that represent the public-key in MQ-based digital signature algorithms.* A short summary of challenges solved by M4GB is available in Table 1. Moreover, in the same chapter we provide an estimation of CPU time and memory to solve larger parameters of type V and VI of MQ challenge. The result is described in Table 2.

Solving Binary Puzzles using Gröbner Bases

Chapter 7 presents an application of Gröbner bases algorithms to find solutions for *binary puzzles*. A binary puzzle is a Sudoku-like puzzle with an entry in each cell taken from the set $\{0,1\}$ instead of $\{0,1,\ldots,9\}$. Let $n \geq 4$ be an even integer and $[n] = \{1,2,\ldots,n\}$. We define the *initial configuration of a binary puzzle* as a subset $M \subseteq [n]^2 \times \{0,1\}$. The set M tells that the value at row i and column j of the binary puzzle is equal to $v_{i,j}$, for all $((i,j),v_{i,j}) \in M$. An $n \times n$ binary puzzle with the initial configuration $M \subseteq [n]^2 \times \{0,1\}$ is solvable if there exists an assignment for each cell that satisfies the following constraints :

* The challenges are random instances of the MQ problem over finite fields that represent MQ-based public-key cryptography and digital signature. All challenges and existing records are available at `https://www.mqchallenge.org`.

Type	(n, m)	k	Duration	Search Space Covered
V	$(24, 16)$	1	$\approx$ 9.3 hours	29%
V	$(25, 17)$	1	$\approx$ 46.33 hours	46.5%
V	$(27, 18)$	1	$\approx$ 10.9 days	95.7%
V	$(28, 19)$	1	$\approx$ 69.75 days	71.5%
VI	$(24, 16)$	1	$\approx$ 1.2 hours	22.6%
VI	$(25, 17)$	1	$\approx$ 9.87 hours	64.5%
VI	$(27, 18)$	1	$\approx$ 31.48 hours	12.9%
VI	$(28, 19)$	1	$\approx$ 7.61 days	54.8%
VI	$(30, 20)$	2	$\approx$ 11.32 days	21.5%

Table 1: *Summary of solved MQ challenges with n variables and m equations using the implementation of* M4GB. *Type V and VI consist of polynomials defined over $\mathbb{F}_2$s and $\mathbb{F}_{31}$ respectively. For each challenge, we solved q^k subsystems by substituting k variables with all possible values in $\mathbb{F}_q$, where q is the order of the finite field. The subsystems are generated after fixing the value of $n - m$ chosen variables.*

m	$m-n$		$\approx$ CPU hours		$\approx$ Memory (GB)	
	V	VI	V	VI	V	VI
20	1	-	402045	-	210	-
21	1	3	$2.85 \cdot 10^6$	217452	764	5.47
22	1	2	$2.02 \cdot 10^7$	$1.39 \cdot 10^6$	2776	189
23	1	2	$1.44 \cdot 10^8$	$8.94 \cdot 10^6$	10087	654
24	1	3	$1.02 \cdot 10^9$	$1.43 \cdot 10^7$	36643	130
25	1	3	$7.23 \cdot 10^9$	$1.57 \cdot 10^8$	133110	528
26	1	3	$5.13 \cdot 10^{10}$	$8.15 \cdot 10^8$	483536	1631
27	1	3	$3.64 \cdot 10^{11}$	$4.24 \cdot 10^9$	$1.76 \cdot 10^6$	5042
28	1	3	$2.58 \cdot 10^{12}$	$2.20 \cdot 10^{10}$	$6.40 \cdot 10^6$	15584
29	1	3	$1.83 \cdot 10^{13}$	$1.15 \cdot 10^{11}$	$2.32 \cdot 10^7$	$4.81 \cdot 10^4$
30	1	3	$1.30 \cdot 10^{14}$	$5.95 \cdot 10^{11}$	$8.44 \cdot 10^7$	$1.49 \cdot 10^5$
31	1	3	$9.23 \cdot 10^{14}$	$3.10 \cdot 10^{12}$	$3.06 \cdot 10^8$	$4.61 \cdot 10^5$
32	1	3	$6.55 \cdot 10^{15}$	$1.61 \cdot 10^{13}$	$1.11 \cdot 10^9$	$1.42 \cdot 10^6$
33	1	3	$4.64 \cdot 10^{16}$	$8.37 \cdot 10^{13}$	$4.04 \cdot 10^9$	$4.40 \cdot 10^6$
34	1	3	$3.30 \cdot 10^{17}$	$4.35 \cdot 10^{14}$	$1.47 \cdot 10^{10}$	$1.36 \cdot 10^7$
35	1	5	$2.34 \cdot 10^{18}$	$1.95 \cdot 10^{15}$	$5.32 \cdot 10^{10}$	$8.37 \cdot 10^4$

Table 2: *Estimated cost to solve larger parameters of type V (top) and VI (bottom) using our implementation of* M4GB *algorithm running on Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30 GHz. Memory usage is for a single subsystem and the total system memory required depends on the number of subsystems being solved simultaneously.*

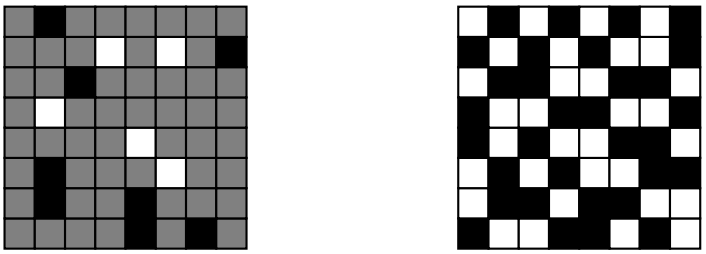

Figure 4: *An example of initial configuration of an 8×8 binary puzzle (left) and its corresponding solution (right). Note that multiple solutions may exist depending on the size and the initial configuration.*

1. There exists no three (3) consecutive cells, both vertically and horizontally, which are filled with the same value
2. The number of zeros and ones in each row and column must be equal, i.e., their number in each row and column must be equal to $n/2$
3. Every two distinct rows (resp. columns) must not be equal.

Solving a binary puzzle can then be viewed as a *constraint satisfaction problem*. Our approach is to see each constraint that must be satisfied as a polynomial equation. Since each entry in the puzzle is taken from the set $\{0, 1\}$, the first natural approach is to look at each constraint as a polynomial over $\mathbb{F}_2$.

Theorem 1.1. *An $n \times n$ Binary puzzle with the initial configuration M is solvable if and only if the following system of polynomial equations over $\mathbb{F}_2$ has a solution in $\mathbb{F}_2^{(n^2)}$:*

$$
\begin{aligned}
& x_{i,j+1}x_{i,j+2} + x_{i,j}x_{i,j+2} + x_{i,j+2} \\
& \qquad + x_{i,j}x_{i,j+1} + x_{i,j+1} + x_{i,j} + 1, && 1 \le i \le n, 1 \le j \le n-2 \\
& x_{i+1,j}x_{i+2,j} + x_{i,j}x_{i+2,j} + x_{i+2,j} \\
& \qquad + x_{i,j}x_{i+1,j} + x_{i+1,j} + x_{i,j} + 1, && 1 \le j \le n, 1 \le i \le n-2 \\
& 1 + \sum_{\substack{k=1 \\ n/2 \preceq k}}^{n} \sigma_{k,n}(x_{i,1}, \ldots, x_{i,n}), && 1 \le i \le n \\
& 1 + \sum_{\substack{k=1 \\ n/2 \preceq k}}^{n} \sigma_{k,n}(x_{1,j}, \ldots, x_{n,j}), && 1 \le j \le n \\
& \prod_{k=1}^{n} (x_{i,k} + x_{j,k} + 1), && 1 \le i \le n-1, i+1 \le j \le n \\
& \prod_{k=1}^{n} (x_{k,i} + x_{k,j} + 1), && 1 \le i \le n-1, i+1 \le j \le n \\
& x_{i,j}^2 + x_{i,j} && 1 \le i \le n, 1 \le j \le n. \\
& x_{i,j} + v_{i,j} && \forall ((i,j), v_{i,j}) \in M
\end{aligned}
$$

where $\sigma_{k,n}(x_1, \ldots, x_n)$ denotes the k-th elementary symmetric polynomial in n-variables $x_1, \ldots, x_n$ and $a \preceq b$, $(a, b \in \mathbb{Z}_{\ge 0})$ if the binary representation of a is covered by the binary representation of b, .

The second approach is to look at each constraint as polynomial over $\mathbb{Q}$ or $\mathbb{Z}$. One advantage of this approach is the polynomial representation for the second

constraint is linear, thus significantly reducing the degree of the same constraint at the expense of defining the polynomials over $\mathbb{Q}$ or $\mathbb{Z}$.

Theorem 1.2. *An $n \times n$ binary puzzle with the initial configuration M is solvable if and only if the following system of polynomial equations over $\mathbb{Q}$ (or $\mathbb{Z}$) has a solution in $\mathbb{Q}^{(n^2)}$ (or $\mathbb{Z}^{(n^2)}$):*

$$\begin{aligned}
&x_{i,j+1}x_{i,j+2} + x_{i,j}x_{i,j+2} - x_{i,j+2} && \\
&\qquad +x_{i,j}x_{i,j+1} - x_{i,j+1} - x_{i,j} + 1, && 1 \le i \le n, 1 \le j \le n-2 \\
&x_{i+1,j}x_{i+2,j} + x_{i,j}x_{i+2,j} - x_{i+2,j} && \\
&\qquad +x_{i,j}x_{i+1,j} - x_{i+1,j} - x_{i,j} + 1, && 1 \le j \le n, 1 \le i \le n-2 \\
&\sum_{j=1}^{n} x_{i,j} - n/2, && 1 \le i \le n \\
&\sum_{i=1}^{n} x_{i,j} - n/2, && 1 \le j \le n \\
&\prod_{k=1}^{n} (x_{i,k} + x_{j,k} - 1), && 1 \le i \le n-1, i+1 \le j \le n \\
&\prod_{k=1}^{n} (x_{k,i} + x_{k,j} - 1), && 1 \le i \le n-1, i+1 \le j \le n \\
&x_{i,j}^2 - x_{i,j} && 1 \le i \le n, 1 \le j \le n \\
&x_{i,j} - v_{i,j} && \forall((i,j), v_{i,j}) \in M.
\end{aligned}$$

The solutions for a binary puzzle can then be obtained by computing a Gröbner bases of the ideal generated by either the polynomials in Theorem 1.1 or Theorem 1.2.

In order to understand the impact of reducing the degree of some equations at the expense of changing the ring where coefficients of the polynomials are defined, we also compare the performance of Gröbner bases algorithm in practice to solve system of equations that represent binary puzzles over $\mathbb{F}_2$, $\mathbb{Q}$, and $\mathbb{Z}$.

New Characterizations of S-Boxes Criteria

Chapter 8 introduces a new-point-of-view to look at cryptographic properties of a vectorial Boolean functions from the settings of ideals in multivariate polynomial ring. We propose new characterization for notions related to the nonlinearity, differential, and autocorrelation from ideal-theoretic viewpoint. The primary contributions are summarized in the following theorems.

Theorem 1.3. *Let $S : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ and $\mathcal{R}$ be a polynomial ring over $\mathbb{F}_2$ in $n+m$ variables. For any $\boldsymbol{a} \in \mathbb{F}_2^n$ and $\boldsymbol{b} \in \mathbb{F}_2^m$ there exists an ideal I of $\mathcal{R}$ such that the bias of a linear approximation on S with input mask $\boldsymbol{a}$ and output mask $\boldsymbol{b}$ is equal to*

$$2^{-n} \cdot (\dim_{\mathbb{F}_2}(\mathcal{R}/I) - 2^{n-1}).$$

Theorem 1.4. *Let $S : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ and $\mathcal{R}$ be a polynomial ring over $\mathbb{F}_2$ in $2(n+m)$ variables. For any $\boldsymbol{\alpha} \in \mathbb{F}_2^n, \boldsymbol{\beta} \in \mathbb{F}_2^m$ there exists an ideal I of $\mathcal{R}$ such that the probability of a differential $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is equal to*

$$2^{-n} \cdot \dim_{\mathbb{F}_2}(\mathcal{R}/I).$$

Theorem 1.5. *Let $S : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ and $\mathcal{R}$ be a polynomial ring over $\mathbb{F}_2$ in $2(n+m)$ variables. For any $\boldsymbol{\alpha} \in \mathbb{F}_2^n, \boldsymbol{b} \in \mathbb{F}_2^m$ there exists an ideal I of $\mathcal{R}$ such that the autocorrelation of the component function $\boldsymbol{b} \cdot S(x)$ at $\boldsymbol{\alpha}$ is equal to*

$$2^n - 2 \cdot \dim_{\mathbb{F}_2}(\mathcal{R}/I).$$

In the same chapter we also exhibit new approaches to compute the bias of a linear approximation, the probability of a differential, and the autocorrelation of a component function that utilize Gröbner bases algorithms. To the best of our knowledge, this is the first work that establishes connections among cryptographic properties of vectorial Boolean functions and computational commutative algebra.

1.3.3 Thesis Outline

The remainder of this thesis consists of seven chapters.

Chapter 2 is a survey on various aspects of MQ problems and solving systems of polynomial equations in cryptology. It describes the intractability of the MQ problem and how it can be used to construct public-key and digital signature schemes. The chapter also describes the relevance of solving MQ problem and systems of polynomial equations in the cryptanalysis of symmetric-key primitives. It also provides a brief survey of existing techniques in the literatures to solve systems of polynomial equations.

Chapter 3 discusses the Gröbner bases and their applications. It serves as the preliminary chapter where we explain all notations and terminologies used throughout this thesis. In the same chapter we also describe two applications of Gröbner bases that are relevant within the scope of this thesis: solving the ideal membership problem and solving systems of multivariate polynomial equations.

Chapter 4 focuses on the Gröbner bases algorithms. We shall explain Buchberger [Buc65, Buc06] and the F_4 algorithm [Fau99] in Section 4.1 and Section 4.2 respectively. In the same chapter, we also discuss several improvement strategies for both algorithms, where their main goal is to detect unnecessary computations in advance. Moreover, we also describe the XL algorithm [CKPS00] in Section 4.3. Even though it was not originally proposed to compute Gröbner bases of an ideal, it can actually be viewed as a redundant variant of the F_4 algorithm.

Chapter 5 explains the M4GB algorithm. It covers the description of the algorithm, the reduction strategy, and the rationale behind its design. The performance comparison of M4GB against FGb [Fau10], Magma [BCP97], and OpenF4 [CVJ] are given in Section 5.7.

Chapter 6 describes the application of the M4GB algorithm to solve concrete challenges of MQ problems proposed in [YDH$^+$15a, YDH$^+$15b] for parameters that represent signature-type MPKC, i.e., an underdefined system of equations over a finite field. We give an overview of existing strategies to solve such a system of equations in Section 6.3. We also discuss the cost of solving larger parameters of signature-type MQ challenge in Section 6.5.

Chapter 7 explains the application of the Gröbner bases algorithm to solve binary puzzles. In particular, the proof of Theorem 1.1 and Theorem 1.2 are given in Section 7.5 and Section 7.6 respectively. We also compare the performance of different implementations of Gröbner bases algorithms to solve sys-

tems of polynomial equations over $\mathbb{F}_2$, $\mathbb{Q}$, and $\mathbb{Z}$ that represent a binary puzzle in Section 7.7.

Chapter 8 presents the relation of cryptographic properties of vectorial Boolean functions and their corresponding ideals. The proof of Theorem 1.3, Theorem 1.4, and Theorem 1.5 are given in Section 8.2, Section 8.3, and Section 8.4 respectively. In each of those sections we also present the algorithms based on the Gröbner bases that compute the bias of a linear approximation, the probability of a differential and the autocorrelation. In relation with autocorrelation, we also discuss a possible way to look at the existence of a linear structure as ideal membership problem in Subsection 8.4.1.

1.3.4 Publications

In addition to [MS17, UM17], the author has co-authored the following publication related to the study of the intractability of the MQ problem and multivariate public-key cryptosystems. The results, though related, are not covered in this dissertation.

[BMSV22] Emanuele Bellini, Rusydi H. Makarim, Carlo Sanna, Javier A. Verbel. *An Estimator for the Hardness of the MQ Problem*. 13th International Conference on Cryptology in Africa (AFRICACRYPT), Fez, Morocco, 2022, pp. 323-347.

Moreover, the author has also contributed to the following publications in the area of cryptanalysis of symmetric-key primitives. Hence, these results are not covered in this dissertation.

[HDRM23] Solane El Hirch, Joan Daemen, Raghvendra Rohit, Rusydi H. Makarim. *Twin Column Parity Mixers and Gaston - A New Mixing Layer and Permutation*. 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20–24, 2023, Proceedings, Part III, pp. 475-506.

[BGG+23b] Emanuele Bellini, David Gérault, Juan Grados, Rusydi H. Makarim, Thomas Peyrin. *Boosting Differential-Linear Cryptanalysis of ChaCha7 with MILP*. (2023). IACR Transactions on Symmetric Cryptology, Volume 2023, Issue 2, pp. 189-223.

[BGG+23a] Emanuele Bellini, David Gérault, Juan Grados, Yun Ju Huang, Rusydi H. Makarim, Mohamed Rachidi, Sharwan K. Tiwari. *CLAASP: a Cryptographic Library for the Automated Analysis of Symmetric Primitives*. Selected Areas in Cryptography (SAC), Fredericton, New Brunswick, Canada, 2023, pp. 387-408.

[BGG+23c] Emanuele Bellini, David Gérault, Juan Grados, Rusydi H. Makarim, Thomas Peyrin. *Fully Automated Differential-Linear Attacks Against ARX Ciphers*. Cryptographers' Track at the RSA Conference (CT-RSA), San Francisco, USA, 2023, pp 252–276.

[MR22] Rusydi H. Makarim and Raghvendra Rohit. *Towards Tight Differential Bounds of Ascon: A Hybrid Usage of SMT and MILP*. (2022). IACR Transactions on Symmetric Cryptology, Volume 2022, Issue 3, pp. 303-340.

[BPM+21] Emanuele Bellini, Alessandro De Piccoli, Rusydi H. Makarim, Sergio Polese, Lorenzo Riva, Andrea Visconti. *New Records of Pre-image Search of Reduced SHA-1 Using SAT Solvers*. Proceedings of the Seventh International Conference on Mathematics and Computing (ICMC), Shibpur, India, 2021, pp. 141-151.

[BGMS24] Emanuele Bellini, Juan Grados, Rusydi H. Makarim, Carlo Sanna. *Finding Differential Trails on ChaCha by Means of State Functions*. International Journal of Applied Cryptography, Volume 4, No. 3/4, 2024, pp. 156-175.

[BBM23] Stefano Barbero, Emanuele Bellini, Rusydi H. Makarim. *Rotational Analysis of ChaCha Permutation*. Advances in Mathematics of Communications, Volume 17, Issue 6, 2023, pp. 1422-1439.

[BM22] Emanule Bellini, Rusydi H. Makarim.*Functional Cryptanalysis: Application to reduced-round Xoodoo*. In IACR Cryptology ePrint Archive (2022). IACR ePrint: 2022/134.

[MT14] Rusydi H. Makarim and Cihangir Tezcan. *Relating Undisturbed Bits to Other Properties of Substitution Boxes*. Lightweight Cryptography for Security and Privacy - Third International Workshop (LightSec), Istanbul, Türkiye, 2014, pp. 109-125.

Finally, the author has also contributed to the following publications related to the implementation of code-based post-quantum cryptography. Hence, these results are also excluded in this dissertation.

[BCM+19] Emanuele Bellini, Florian Caullery, Rusydi H. Makarim, Marc Manzano, Chiara Marcolla, Victor Mateu. *Advances and Challenges of Rank Metric Cryptography Implementations*. IEEE 37th International Conference on Computer Design (ICCD), Abu Dhabi, United Arab Emirates, 2019, pp. 325-328.

[MAB+21] Carlos Aguilar Melchor, Nicolas Aragon, Emanuele Bellini, Florian Caullery, Rusydi H. Makarim, Chiara Marcolla. *Constant Time Algorithms for ROLLO-I-128*. SN Computer Science, Volume 2, Number 5, September 2021, article number 382.