Algebraic Cryptanalysis of Cryptographic Schemes with Extension Field Structure

Morten Øygarden

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2021

UNIVERSITY OF BERGEN

Algebraic Cryptanalysis of Cryptographic Schemes with Extension Field Structure

Morten Øygarden

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Date of defense: 06.09.2021

Year: 2021

Title: Algebraic Cryptanalysis of Cryptographic Schemes with Extension Field Structure

Name: Morten Øygarden

Print: Skipnes Kommunikasjon / University of Bergen

Acknowledgements

I would first and foremost thank my supervisors Øyvind Ytrehus and Håvard Raddum, for all their help and guidance. I am also grateful to my coauthors, whom I have had the pleasure of collaborating with over these past three years. Special thanks are due to friends and colleagues at both Simula UiB and the Selmer Center; I am particularly grateful to Andrea and Isaac for all the interesting and helpful discussions.

Finally, my deepest gratitude goes to Mona for all her support and understanding.

My studies have been funded by The Research Council of Norway through the project “qsIoT: Quantum safe cryptography for the Internet of Things”.

Morten Øygarden
Bergen, May 2021

Abstract

Post–Quantum Cryptography studies cryptographic algorithms that quantum computers cannot break. Recent advances in quantum computing have made this kind of cryptography necessary, and research in the field has surged over the last years as a result. One of the main families of post–quantum cryptographic schemes is based on finding solutions of a polynomial system over finite fields. This family, known as *multivariate cryptography*, includes both public key encryption and signature schemes.

The majority of the research contribution of this thesis is devoted to understanding the security of multivariate cryptography. We mainly focus on *big field schemes*, i.e., constructions that utilize the structure of a large extension field. One essential contribution is an increased understanding of how Gröbner basis algorithms can exploit this structure. The increased knowledge furthermore allows us to design new attacks in this setting. In particular, the methods are applied to two encryption schemes suggested in the literature: EFLASH and Dob. We show that the recommended parameters for these schemes will not achieve the proposed 80–bit security. Moreover, it seems unlikely that there can be secure and efficient variants based on these ideas. Another contribution is the study of the effectiveness and limitations of a recently proposed rank attack. Finally, we analyze some of the algebraic properties of MiMC, a block cipher designed to minimize its multiplicative complexity.

Outline

This doctoral thesis comprises five chapters, and is based on four research papers. The first chapter presents a general introduction, and explains how the work of this thesis fits in the bigger picture of cryptography. A more specific overview of the research field is given in Chapter 2. Chapter 3 provides a brief summary of the papers, and Chapter 4 contains conclusions and final remarks. Finally, the four research papers are included in Chapter 5, listed as follows:

I Øygarden, M., Felke, P., Raddum, H., and Cid, C. Cryptanalysis of the multivariate encryption scheme EFLASH. In *Cryptographers Track at the RSA Conference*, pages 85-105. Springer, 2020.

II Øygarden, M., Felke, P., and Raddum, H. Analysis of Multivariate Encryption Schemes: Application to Dob. In *International Conference on Public-Key Cryptography (PKC)*, pages 155-183. Springer, 2021.
Invited to the Journal of Cryptology.

III Øygarden, M., Smith–Tone, D., and Verbel, J. On the Effect of Projection on Rank Attacks in Multivariate Cryptography. To appear in *PQCrypto: International Conference on Post-Quantum Cryptography*, 2021.

IV Eichlseder, M., Grassi, L., Lüftenegger, R., Øygarden, M., Rechberger, C., Schofnegger, M., and Wang, Q. An Algebraic Attack on Ciphers with Low–Degree Round Functions: Application to Full MiMC. In: *International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt)*, pages 477-506. Springer, 2020.

Note that the papers have been ordered thematically, not chronologically.

Notation and Conventions

The papers presented in Chapter 5 use different notation, which is introduced in the individual papers. The remaining chapters adhere to the following conventions. Unless otherwise stated, $\mathbb{F}$ denotes a finite field. If there is a need to specify the size of the field, we will also write $\mathbb{F}_q$ or $\mathbb{F}_{q^n}$, where q is the power of a prime number. We will also make use of the polynomial rings:

$$\begin{aligned} R_{q,n} &= \mathbb{F}_q[x_1,\dots,x_n]/\langle x_1^q - x_1,\dots,x_n^q - x_n\rangle, \qquad \text{and} \\ \overline{R}_{q,n} &= \mathbb{F}_q[x_1,\dots,x_n]/\langle x_1^q,\dots,x_n^q\rangle. \end{aligned}$$

All computations of polynomials in the univariate ring $\mathbb{F}_q[X]$ will implicitly be performed over the quotient ring $\mathbb{F}_q[X]/\langle X^q - X\rangle$. Homogeneous polynomials will be denoted by superscript h, e.g., g^h. Moreover, for a fixed polynomial f of degree l, we will use f^h to denote its leading form, i.e., the homogeneous degree l part of f. This extends to sequences of polynomials, where we write $\mathcal{F} = \{f_1,\dots,f_m\}$, and $\mathcal{F}^h = \{f_1^h,\dots,f_m^h\}$. Finally, while the thesis will predominantly be written in the "mathematical we", it will occasionally use the first person singular to emphasize when a statement is my own opinion.

Contents

Chapter 1

General Background

1.1 Introduction to Cryptography

Cryptography is the study of secure communications in the presence of adversaries. Historically, its use was mainly limited to message confidentiality for state and military leaders. Sensitive messages were turned into unintelligible ciphertexts that could ideally only be read by their intended recipient. This was traditionally achieved through steganography and linguistics, but the twentieth century saw a sharp shift in focus towards mathematics and computer science. Open, academic research started in earnest after the second world war. Claude Shannon published one of the foundational works of modern cryptography in 1949 [100]. The first open encryption standard, DES, was published in 1977 [85], which further accelerated the advancement of the field. Nowadays, cryptography is an active research area and covers more aspects of communication security, such as authentication, ensuring message integrity, and providing non–repudiation. It provides a crucial component of everyday life, including communication via mail and messaging apps, browsing the internet using HTTPS, and e–commerce.

Research in cryptography includes both the design of secure primitives and protocols, as well as cryptanalysis; the search for weaknesses in said designs. Proposed schemes are only secure up to a specified security level. This level is an estimated lower bound of the resources needed by a third party to bypass the effect of the scheme, such as retrieving secret information or forging signatures. It is typically measured in time, memory, or data (e.g., amount of known plaintext–ciphertext pairs). An attack that falls below this lower limit is regarded as a break, even if the procedure is computationally impractical. Thus, cryptanalysis is not only a nefarious activity, but considered a vital part of modern cryptography. For instance, design principles are often motivated by attacks found through such analysis. Moreover, trust in a cipher is typically only obtained after years of (unsuccessful) third–party cryptanalysis.

The cryptographic primitives, upon which more elaborate protocols are built, are traditionally divided into three categories: *hash functions*, *secret key (symmetric) primitives* and *public key (asymmetric) primitives*.

Hash Functions

A cryptographic hash function is an algorithm that maps input of arbitrary length to an output of fixed length. We furthermore require the mapping to be *pre–image resistant*, *second pre–image resistant*, and *collision resistant*. The most commonly used hash functions are the SHA2 and SHA3 families that have been standardized by the (US) National Institute for Standards and Technology (NIST) [87, 88].

Secret Key Primitives

In symmetric ciphers, the communicating parties share the same secret key. Examples include block ciphers, stream ciphers, authenticated encryption, and message authentication codes. Block ciphers are the common choice for bulk encryption, with the Advanced Encryption Standard (AES)[86] being the most popular variant.

Public Key Primitives

Public key, or asymmetric, cryptography includes a public key, in addition to the secret key. The secret key is known by only one of the communicating parties, whereas the public key is assumed to be known by everyone, including potential adversaries. In public key *encryption schemes*, a message is encrypted using the public key, and decryption is only feasible by use of the secret key. Asymmetric encryption is generally more cumbersome than using symmetric primitives, so this type of scheme is commonly used as a Key Encapsulation Mechanism (KEM), where two parties derive the secret key of, e.g., a block cipher.

The other main class of public key primitives is *signature schemes*, which are used to verify the authenticity of (the hash value of) messages. Only the owner of the secret key can efficiently compute signatures for a given message, which can then be verified by anyone with access to the public key.

The most common public key encryption and signature schemes in use today are based upon the *Integer Factorization Problem* (IFP) such as RSA [97], the *Discrete Logarithm Problem* (DLP) like Elgamal [53], or the *Elliptic Curve Discrete Logarithm Problem* (ECDLP) such as EdDSA [19].

1.2 Post–Quantum Cryptography

1.2.1 Quantum Computing and Algorithms

In 1994, Peter Shor showed that a large–scale quantum computer can solve IFP, DLP, and ECDLP in polynomial time [101]. Hence, if such a quantum computer were to be constructed, it would be able to break the public key cryptosystems currently in use today. In 1997, Lov Grover presented a quantum search algorithm with quadratic speedup [66]. Search problems can be encountered in a wider array of cryptographic applications, and this algorithm provides at least a theoretical speed–up for attacking symmetric ciphers and hash functions. On the other hand, there is a significant constant factor overhead that is associated with quantum error–correction, and it has been argued (see e.g., [11]) that Grover's algorithm might not be able to outperform classical approaches in practice. Either way, the speed–up of Grover's algorithm is less dramatic than the exponential speed–up of Shor's algorithm, and doubling the key length is, for instance, sufficient to secure symmetric ciphers.

Quantum computing were long seen as a purely theoretical field, but in the recent years much resources have been devoted to the research and development of this technology. Serious actors such as IBM, Microsoft and Google, as well as universities and agencies in various countries, all have their programs on quantum computing [84, Section 7.4.1]. In 2019, Google announced that their quantum computer was able to achieve so–called "quantum supremacy", i.e., solving a problem that is out of reach for modern day (classical) supercomputers [9]. In 2020, a quantum computer developed in Hefei, China, did also achieve quantum supremacy, for a different problem [114]. Nevertheless, implementing Shor's algorithm for large integers remains a tremendous challenge, and it is not likely to be feasible in the very near future. A 2019 report from the (US) National Academies of Sciences, Engineering, and Medicine (NASEM) writes ([84, p. 157]):

> "**Key Finding 1:** Given the current state of quantum computing and recent rates of progress, it is highly unexpected that a quantum computer that can compromise RSA 2048 or comparable discrete logarithm-based public key cryptosystems will be built within the next decade."

Defending against potential future quantum computers is no easy task. Three reasons are often pointed out for why this is a lengthy process.

- Firstly, developing and properly analyzing *post–quantum* cryptographic schemes, i.e., cryptosystems based on mathematical problems believed to be hard to solve by quantum computers, requires several years.

- Secondly, it is a slow process to completely replace old primitives in protocols that are in use. An illustrative example, pointed out in [84], is that while several works following a 2004 paper by Wang et al., [108] found severe weaknesses in the hash function MD5, Microsoft did not fully restrict its use in its systems until a 2014 update [81].

- Thirdly, a determined adversary can store encrypted messages today, with the intent of decrypting later. Thus, quantum safe cryptography should ideally be implemented well in advance full–scale quantum computers.

With all this in mind, it seems wise to start the shift to post–quantum cryptography sooner rather than later. Indeed, the 2019 NASEM report concludes on this issue as follows ([84] page 188):

> "**Key Finding 10:** Even if a quantum computer that can decrypt current cryptographic ciphers is more than a decade off, the hazard of such a machine is high enough–and the time frame for transitioning to a new security protocol is sufficiently long and uncertain–that prioritization of the development, standardization, and deployment of post–quantum cryptography is critical for minimizing the chance of a potential security and privacy disaster."

1.2.2 Post–Quantum Standardization

The potential threat quantum computers would pose against current public key cryptosystems prompted the National Institute of Science and Technology (NIST) to start the work towards new standards. In late 2016 the agency sent out a call for proposals of new, post–quantum key establishment and digital signature schemes [96]. By the end of 2017, NIST had received 69 acceptable submissions, and started the first round of their Post–Quantum Cryptography (PQC) standardization process. This public "competition–like" process relies on the cryptographic research community for security and efficiency analysis. In the summer of 2020 the third, and current, round was announced with seven finalists and eight alternate candidates [1]. NIST aims to select a small number of the finalists for new key establishment and digital signature standards by 2022; some of the alternate candidates may be standardized after an additional fourth round.

The post–quantum cryptographic schemes are typically divided into five families, depending on their underlying mathematical problem. These are: Lattice–based, Code–based, Hash–based, Isogeny–based, and Multivariate schemes. The families differ in underlying hardness assumptions and performance characteristics. There is also an intrinsic motivation to standardize schemes from different families, in order to achieve robustness against a sudden breakthrough in cryptanalysis (page 5, [1]). This thesis will mainly be concerned in the multivariate family, which we further introduce here.

1.3 Multivariate Cryptography

Let $\mathcal{F}$ be a fixed set of m quadratic polynomials $p_i \in \mathbb{F}[x_1,\dots,x_n]$, for $1 \leq i \leq m$. The Multivariate Quadratic (MQ)–problem asks to find (if possible) a solution $\mathbf{a} = (a_1,\dots,a_n) \in \mathbb{F}^n$ which satisfies $p_i(\mathbf{a}) = 0$ for all $1 \leq i \leq m$. The MQ–problem has been shown to be NP–complete for $\mathbb{F} = \mathbb{F}_2$ [61]. It is also believed to be hard on average, when $m \approx n$. The problem is furthermore expected to be difficult even for quantum computers, and the best known quantum algorithms are based on Grover's algorithm applied to either a direct search [98], or in conjunction with dedicated classical algorithms [20, 57].

The MQ–problem can be used to construct both encryption and signature schemes. In both cases the public key is a polynomial system $\mathcal{F}$, but its properties differ depending on the setting. For encryption, we want the map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ to be (almost) injective, and encrypting a plaintext $\mathbf{a} \in \mathbb{F}^n$ simply consists of evaluating $\mathcal{F}(\mathbf{a}) = \mathbf{c} \in \mathbb{F}^m$. Decryption is done by finding a solution to the system $\mathcal{F}(\mathbf{x}) - \mathbf{c} = \mathbf{0}$. In the signature setting, it is necessary for $\mathcal{F}$ to be (almost) surjective. The user signs a (hashed) document $\mathbf{h} \in \mathbb{F}^m$ by finding a valid solution to $\mathcal{F}(\mathbf{x}) - \mathbf{h} = 0$; verification is performed by evaluating $\mathcal{F}$ at the signature.

In order for decryption/signing to be efficient, it is necessary to generate $\mathcal{F}$ in a structured manner. Typically, the legitimate user chooses random invertible matrices $S \in \mathbb{F}^{n\times n}$ and $T \in \mathbb{F}^{m\times m}$, and constructs the public key as the composition

$$\mathcal{F} = T \circ \mathcal{F}' \circ S : \mathbb{F}^n \longrightarrow \mathbb{F}^m, \tag{1.1}$$

where the *central map*, $\mathcal{F}'$, is a simple (often publicly known) system of m polynomials over $\mathbb{F}[x_1,\dots,x_n]$, which is easy to solve. The role of the matrices S and T is to hide the structure of the central map from an attacker, but the effectiveness of this concealment is largely dependent on the chosen $\mathcal{F}'$. For instance, the first multivariate encryption scheme, the C^* scheme by Matsumoto and Imai in 1988 [80], was subject to an efficient attack by Patarin [90], due to the chosen central map. This spurred a long line of research into designing different central maps, and/or effective modifications to the basic composition in Equation (1.1).

Multivariate cryptography has been an active research area for the last three decades, with much research going into both the design and analysis of new schemes. The state of the art as of late 2017 at the start of the NIST PQC–competition, was the existence of several promising signature schemes, whereas good multivariate encryption schemes seemed harder to design. Furthermore, the more well–studied schemes can typically be divided into two subfamilies: *oil–and–vinegar* (OV) and *big field*. All this is reflected in the fact that the two multivariate constructions that made it to the third round of the PQC process was the signature finalist *Rainbow* [42](OV family), and the alternate signature candidate G*e*MSS [32](big field family). In section 2.1 we will dive deeper into the big field family, which is the setting for three of the articles comprising this thesis.

Recent Multivariate Analysis in the NIST PQC Process

In late 2020, attacks against both Rainbow and G*e*MSS were published [24, 105]. The team behind the Rainbow submission has released a note arguing that, while they acknowledge the new attack, they still regard the round 3 parameters to achieve their proposed security [34]. At the time of writing, no similar public statement has been issued by the team behind G*e*MSS.

Paper 3 in this thesis is concerned with further analyzing the techniques used in attack against G*e*MSS.

1.4 New Designs for Block Ciphers and Hash Functions

We already noted in Section 1.2.1 that quantum computers are not considered to pose a serious threat against AES or SHA2/SHA3. However, while these primitives are reasonably efficient for use in traditional cryptographic protocols, they are cumbersome to use in some modern constructions (see e.g., [17, Table 1], or [6]). This has motivated new designs for block ciphers and hash functions, which are more tailored to specific use–cases. Oftentimes, this includes a more succinct algebraic description than their traditional counterparts. A closer look at this trend will be provided in Section 2.4, and Paper 4 will be concerned with the analysis of one such construction, MiMC.

Chapter 2

Specific Background

2.1 Big Field Schemes

Recall that Equation (1.1) requires a central map $\mathcal{F}' : \mathbb{F}^n \to \mathbb{F}^m$ that is easy to solve, in the sense that for a given $\mathbf{y} \in \mathbb{F}^m$, it is easy to compute an element $\mathbf{x} \in \mathbb{F}^n$ satisfying $\mathcal{F}'(\mathbf{x}) = \mathbf{y}$. The big field family of multivariate schemes aims to construct this central map as having a simple description over an extension field. To be more specific, for a positive integer n, we fix an extension field isomorphism $\phi : \mathbb{F}_q^n \to \mathbb{F}_{q^n}$, and let $\mathcal{F}'$ be the composition $\mathcal{F}' = \phi^{-1} \circ F \circ \phi$, for a polynomial $F(X) \in \mathbb{F}_{q^n}[X]$. Recalling that the Frobenius morphism is linear over $\mathbb{F}_q$, it is common to let the polynomial $F(X)$ consist of monomials that are the multiplication of at most two Frobenius powers, i.e., $X^{q^{i_1}+q^{i_2}}$, for integers i_1, i_2. This ensures that the associated polynomials over the base field $\mathbb{F}_q$ are quadratic. There are two main strategies in the literature for constructing maps F that are easy to solve:

1. F consists of a single quadratic monomial, $F(X) = X^{1+q^\theta}$, for an integer $1 \leq \theta \leq n$ where $\gcd(q^n - 1, q^\theta + 1) = 1$. This choice of central map was used for C^* [80] and its variants, including PFLASH [35], whose security will be analyzed in Paper 3, and EFLASH [30], which will be attacked in Paper 1. C^* and PFLASH uses exponentiation to invert F, while EFLASH utilizes the bilinear relations discovered in [90].

2. The second type of central map is used in the *Hidden Field Equations* (HFE) scheme [92], and its variants. A degree bound D_{HFE} is chosen, and F is defined as

$$F(X) = \sum_{\substack{i,j \in \mathbb{N} \\ q^i+q^j \leq D_{HFE}}} \alpha_{i,j} X^{q^i+q^j} + \sum_{\substack{i \in \mathbb{N} \\ q^i \leq D_{HFE}}} \beta_i X^{q^i} + \gamma, \tag{2.1}$$

for constants $\alpha_{i,j}, \beta_i, \gamma \in \mathbb{F}_{q^n}$. A root of F can now be found using (variations of) the Berlekamp algorithm [18], whose efficiency mainly depends on D_{HFE}.

In [77], Macario–Rat and Patarin propose a third strategy for constructing a central map F, resulting in the *Two–Face* family. Consider $F(X) = Y$, for a fixed Y. While F itself has a high degree, and is not easy to invert directly, there is a corresponding bivariate polynomial equation $E_2(X,Y) = 0$, which has a low degree in the X–variable. Given Y, the legitimate user can easily recover X by finding a root of this latter equation. The security of the encryption variant of the Two–Face family, known as the *Dob encryption scheme*, will be analyzed in Paper 2.

2.1.1 Security of Big Field Schemes

Analysis of big field schemes can typically be divided into four categories: Gröbner basis attacks, rank attacks, differential attacks, and ad hoc attacks. The complexity of Gröbner basis attacks, also known as direct attacks, have been particularly difficult to determine for big field schemes, and Section 2.2 is devoted to these techniques. Rank attacks have also proved to be extremely efficient; we will return to them in Section 2.3.

Differential attacks constitute the third major class, and is perhaps best exemplified by the works by Dubois et al., ([49, 50]) which broke SFLASH [93], an early descendant of the C^* scheme. This class of attacks is now better understood than the other classes. Theorem 2 in [102] has shown that schemes based on the HFE central maps are resistant against these attacks; PFLASH and EFLASH, the most recent variants of the C^* scheme, are protected against differential attacks through the projection modifier [43, 29], which we will introduce in the next subsection.

2.1.2 Modifications

Directly using big field central maps in the composition of Equation (1.1), sometimes referred to as ‘unmodified’ or ‘nude’ versions, tend to lead to insecure constructions. For instance, we have already noted that C^* was broken by Patarin in [90]. Faugère and Joux solved the first HFE Challenge by computing a Gröbner basis in [58]. Macario–Rat and Patarin concludes from experiments in [77] that the nude version of the Dob encryption scheme seemed to be weak; this is indeed shown to be the case in Appendix D of Paper 2.

Certain modifications are then made in order to make these schemes secure against certain attacks. We list the most commonly used modifications in the following. $\mathcal{F}$:

$\mathbb{F}^n \to \mathbb{F}^m$ will refer to an 'unmodified' version of a scheme, as defined in Equation (1.1). We denote the public key, after applying specific modifier(s), as $\mathcal{P}$.

1. The *minus* (-) modifier removes a of the polynomials from the public key. If $\tau : \mathbb{F}^m \to \mathbb{F}^{m-a}, (y_1, \dots, y_m) \mapsto (y_1, \dots, y_{m-a})$ is the projection on the first $m-a$ coordinates, the public key with this modification is given as $\mathcal{P} = \tau \circ \mathcal{F} : \mathbb{F}^n \to \mathbb{F}^{m-a}$.

2. The *plus* (+) modifier appends l randomly chosen quadratic polynomials to the public key. If $H_l : \mathbb{F}^n \to \mathbb{F}^l$ is the map consisting of these polynomials, then $\mathcal{P} = \mathcal{F} || H_l : \mathbb{F}^n \to \mathbb{F}^{m+l}$.

3. The *projection* (p) modifier refers to the idea of projecting the plaintext space onto some subspace[1]. This is achieved by applying a linear embedding $\pi : \mathbb{F}^{n'} \to \mathbb{F}^n$, for some $n' < n$, resulting in $\mathcal{P} = \mathcal{F} \circ \pi : \mathbb{F}^{n'} \to \mathbb{F}^m$.

4. The *internal perturbation* (*ip*) modifier chooses k linear forms $(v_1, \dots, v_k) : \mathbb{F}^n \to \mathbb{F}^k$, and adds quadratic combinations of them to $\mathcal{F}$ to generate the public key. For a randomly chosen quadratic function, $H_{ip} : \mathbb{F}^k \to \mathbb{F}^m$, the public key is $\mathcal{P} = \mathcal{F} + (H_{ip} \circ (v_1, \dots, v_k)) : \mathbb{F}^n \to \mathbb{F}^m$.

5. The Q_+ modifier randomly chooses t quadratic polynomials $(q_1, \dots, q_t) : \mathbb{F}^n \to \mathbb{F}^t$ and adds linear combinations of them to form the public key. For a linear map $H_{Q_+} : \mathbb{F}^t \to \mathbb{F}^m$, we get $\mathcal{P} = \mathcal{F} + (H_{Q_+} \circ (q_1, \dots, q_t)) : \mathbb{F}^n \to \mathbb{F}^m$. Note that this is somewhat similar to the plus modifier, but the important difference is that the number of public polynomials does not increase.

6. The *vinegar* (v) modifier is applied to the univariate mapping $F(X)$, that is used to construct $\mathcal{F}$. It is most commonly used to modify HFE, so we describe it in this setting. Consider v 'vinegar variables' $\mathbf{x}_v = (x_1, \dots, x_v)$, and polynomials $\beta_i, \gamma \in \mathbb{F}_{q^n}[\mathbf{x}_v]$, where the β_i–polynomials are linear, and γ is quadratic. The central map for HFE with vinegar now uses the polynomial

$$F(X, \mathbf{x}_v) = \sum_{\substack{i,j \in \mathbb{N} \\ q^i + q^j \leq D_{HFE}}} \alpha_{i,j} X^{q^i + q^j} + \sum_{\substack{i \in \mathbb{N} \\ q^i \leq D_{HFE}}} \beta_i(\mathbf{x}_v) X^{q^i} + \gamma(\mathbf{x}_v). \tag{2.2}$$

   The legitimate user may now partially evaluate $F(X, \mathbf{x}_v)$ at a fixed value $\mathbf{x}_v \in \mathbb{F}_q^v$ to recover an HFE polynomial, as described in Equation (2.1). This is similar to the (*ip*) modifier, with the difference that v extra variables appear in the public polynomials.

[1] A more suitable name for this modifier might be "embedding", but we stick to "projection", as this is more commonly used in the literature.

The various modifications work against different attacks, and the drawbacks of using them may differ between encryption and signature schemes. They are often used in combinations. For instance, PFLASH and EFLASH are schemes built on C^* with the projection and minus modifiers. G*e*MSS is a variant of HFEv-, i.e., it uses the HFE central map with minus and vinegar. Finally, the Dob encryption scheme uses the Q_+ and (ip) modifiers.

Note that this exposition only scratches the surface of multivariate cryptography, and we end this section with a few suggestions for further reading. Section 4 of [110] provides an extended list of modifications that have been discussed in the literature. A recently written overview of multivariate schemes can be found in Section 3.1 of [33]. Finally, [31] studies the possible modifications to use with the C^* map, and their potential parameters.

2.2 Polynomial System Solving

Let $\mathcal{P} = (p_1, \ldots, p_m)$ be the public key of a multivariate scheme, where each p_i is a quadratic polynomial in $R_{q,n} = \mathbb{F}_q[x_1, \ldots, x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle$. Given a tuple $\mathbf{y} = (y_1, \ldots, y_m) \in \mathbb{F}_q^m$, we are interested in analyzing the complexity of finding an element $\mathbf{x} = (x_1 \ldots, x_n) \in \mathbb{F}_q^n$ satisfying $\mathcal{P}(\mathbf{x}) = \mathbf{y}$. Note that we want to find $\mathbf{x}$ with entries in $\mathbb{F}_q$, as opposed to any extension of $\mathbb{F}_q$, which is why we include the quotient relations in $R_{q,n}$. Solving this problem allows an attacker to decrypt an intercepted plaintext, or forge the signature of a document. A polynomial system with $m = n$ is known as a *determined* system, and the case $m > n$ and $m < n$ will be referred to as *overdetermined* and *underdetermined*, respectively. Encryption schemes tend to be (over)determined in order to be ("probabilistically") injective, whereas signature schemes are often (under)determined, which increases the likelihood that a valid signature exists for any document. Extremely overdetermined systems, where m exceeds the number of monomials, are easily solved by linearization. There also exists probabilistic algorithms to find a solution in polynomial time for very underdetermined systems ($n \geq m(m+3)/2$ for fields of even characteristic [82] and $n \geq m(m+1)/2$ for fields of odd characteristic [37]). Somewhat underdetermined systems, say $n = \alpha m$ for a rational number $1 < \alpha \lesssim m/2$, are often converted into determined systems with $m - \lfloor \alpha \rfloor + 1$ polynomials and variables, using the methods described in [106]. For this reason, all polynomial systems discussed in this section are assumed to be either determined or overdetermined.

The general mathematical tool for solving polynomial systems involves computing a Gröbner basis. The necessary theory, along with efficient algorithms for computing said basis, is introduced in the next subsections. We then discuss the class of semi–

regular sequences, and how they relate to ‘generic’[2] polynomial systems. It turns out that computing Gröbner bases for the public keys of big field multivariate schemes are easier than what one would expect for ‘generic’ systems. This is often heuristically explained through the notion of a first fall degree which will be the focus of the subsequent subsection. Finally, we discuss other algorithms for polynomial system solving.

2.2.1 Gröbner Bases

We start by recalling a few of the key concepts behind Gröbner bases; a more detailed overview can be found in [39]. For a fixed system of equations $\mathcal{P}(\mathbf{x}) = \mathbf{y}$, we define its associated ideal

$$I = \langle p_1(\mathbf{x}) - y_1, \ldots, p_m(\mathbf{x}) - y_m \rangle.$$

Before continuing, we need a way of ordering the monomials of $\mathbb{F}_q[x_1, \ldots, x_n]$. While there exist several important monomial orders, it is usually most efficient to compute a basis in the *graded reverse lexicograhpic (grevlex)* order. It is defined as follows.

Definition 1 (Grevlex Ordering) *The* grevlex ordering *is a total order relation* $\geq$ *on the set of monomials of* $\mathbb{F}_q[x_1, \ldots, x_n]$*, which satisfies the following property. For two monomials* $\mathbf{a} = x_1^{a_1} \ldots x_n^{a_n}$*, and* $\mathbf{b} = x_1^{b_1} \ldots x_n^{b_n}$ *we have* $\mathbf{a} > \boldsymbol{b}$ *if either*

- *deg(*$\mathbf{a}$*)* $>$ *deg(*$\mathbf{b}$*), or*
- *deg(*$\mathbf{a}$*)* $=$ *deg(*$\mathbf{b}$*), and* $a_{i_0} < b_{i_0}$ *for the largest integer* $1 \leq i_0 \leq n$ *such that* $a_{i_0} \neq b_{i_0}$.

For a fixed monomial ordering $\geq$, we define the *leading monomial* of any polynomial $f \in \mathbb{F}_q[x_1, \ldots, x_n]$, $LM(f)$, to be the largest monomial in f, according to $\geq$, and the *leading coefficient*, $LC(f)$, to be its associated coefficient. The *leading term* is then $LT(f) = LC(f) \cdot LM(f)$.

Definition 2 (Gröbner Basis) *Fix a monomial ordering* $\geq$*, and consider an ideal* $I \subset \mathbb{F}_q[x_1, \ldots, x_n]$*. A* Gröbner basis *associated with* I *and* $\geq$*, is a finite set of polynomials* $G = (g_1, \ldots, g_k) \subset I$*, such that for any* $f \in I$ *we have that* $LT(f)$ *is divisible by* $LT(g_i)$ *for some* $1 \leq i \leq k$.

A Gröbner basis G *is furthermore said to be* reduced *if for all* $g_i \in G$ *we have i)* $LC(g_i) = 1$*, and ii) no monomial of* g_i *lies in* $\langle LM(\{G \setminus g_i\}) \rangle$*. Here* $LM(\{G \setminus g_i\})$ *denotes the set consisting of the leading monomials of all polynomials in* $\{G \setminus g_i\}$.

[2] Note that our use of “generic polynomial systems” is different from the definition found in algebraic geometry. The usage of “generic” is more cavalier in the cryptographic literature. For instance, a generic quadratic polynomial system of m polynomials in $R_{q,n}$ is said to have property X, if the (vast) majority of systems of m quadratic polynomials in $R_{q,n}$ have property X.

Gröbner bases have several interesting properties that make them an important tool in the study of polynomial ideals. A reduced Gröbner basis for a fixed monomial order exists, and is unique for any nonzero polynomial ideal I, though it does in general depend on the chosen ordering. As the name suggests, it does indeed form a basis for I. The remainder after dividing by a Gröbner basis is unique, which leads to an efficient method for determining ideal membership, and allows for computations in $\mathbb{F}_q[x_1,\dots,x_n]/I$. More interesting to us will be the role Gröbner bases play in polynomial system solving. In general, computing a Gröbner basis in the grevlex order for I is only the first step; the Gröbner basis is typically transformed into a Gröbner basis in the lexicographical order. Under certain assumptions, this latter basis will contain a univariate polynomial where a root can be found. Back substitution with a fixed root yields another univariate polynomial, and repeating this process leads to a solution (more information can be found in Section 2 of [28]). There are examples of systems arising in cryptography where the complexity of finding a solution is dominated by the order changing or root finding step (see e.g., MiMC in the modelling described in [3]). The polynomial systems we are investigating in Papers 1 and 2 will, however, be much more well behaved. These papers study encryption systems, so we can reasonably expect a unique solution for a given $y_1,\dots,y_m$. Furthermore, the base field is $\mathbb{F}_2$, and augmenting I with the field equations $x_i^2+x_i$, $1 \le i \le n$ in I, ensures that the ideal is radical. Hence, by the Nullstellensatz, the Gröbner basis will be $(x_1+y_1,\dots,x_n+y_n)$, for all graded monomial orders.

2.2.2 Computing Gröbner Bases

Buchberger proposed an algorithm for computing Gröbner bases in his thesis [26], and several others have been developed since. Perhaps most notable is Faugère's F_4 algorithm [54], which often provides a significant speed–up when compared to Buchberger's algorithm. Faugère subsequently improved upon this, by reducing the number of unnecessary computations performed, in the F_5 algorithm [55]. Yet another noteworthy mention is the M4GB algorithm due to Makarim and Stevens, which in some cases seem to improve upon F_4 in terms of memory and run time [79]. This is just the tip of the iceberg when it comes to Gröbner basis algorithms; a deeper dive into this research area can be found in the survey presented in [52].

For the rest of this thesis, we will focus on the F_4 algorithm for computing Gröbner bases. This is the default algorithm used in the computer algebra system MAGMA, which serves as one of the main reference softwares for Gröbner basis computation in the cryptographic literature. It is also what we have used for the experiments presented in this work.

Complexity Estimates

Most modern Gröbner basis algorithms make use of linear algebra. The relation between polynomial systems and linear algebra is given by (variants of) Macaulay matrices, which we define for quadratic polynomial systems as follows:

Definition 3 *Let $\mathcal{P}$ be an (inhomogeneous) polynomial system in $R_{q,n}$, of degree two. An (inhomogeneous)* Macaulay matrix *of $\mathcal{P}$ at degree D, $M_D(\mathcal{P})$, is a matrix with entries in $\mathbb{F}_q$, such that:*

1. *The columns are indexed by the monomials of degree $\leq D$ in $R_{q,n}$, according to a fixed order.*
2. *The rows are indexed by the possible combinations $x^\alpha p_i$, where $1 \leq i \leq n$ and $x^\alpha \in R_{q,n}$ is a monomial of degree $\leq D-2$. The entries in one row corresponds to the coefficients of the associated polynomial.*

Similarly, we define the homogeneous Macaulay matrix *of $\mathcal{P}$ at degree D, $\overline{M}_D(\mathcal{P})$, by considering $\mathcal{P}^h \in \overline{R}_{q,n}$, only including monomials of degree D in the columns, and rows associated to combinations $x^\alpha p_i^h$, $deg(x^\alpha) = D-2$.*

Indeed, in the homogeneous setting a Gröbner basis can be computed through Gaussian elimination on Macaulay matrices up to some degree $\leq n$, as observed by Lazard in [72]. In the cryptographic literature, it has been common to estimate the complexity of Gröbner basis algorithms by the complexity of performing linear algebra on large "Macaulay–like" matrices up to some degree. Let $D_{solv} = D_{solv}(\mathcal{P})$ denote the *solving degree*, which is the degree associated with the largest matrix in a Gröbner basis computation of $\mathcal{P}$, using a fixed algorithm. Under the assumption that all monomials up to this degree appear in the computation, we estimate the complexity of computing a Gröbner basis of $\mathcal{P}$ to be:

$$\text{Complexity}_{\text{GB},\mathbb{F}_2} = \mathcal{O}\left(\left(\sum_{i=0}^{D_{solv}} \binom{n}{i} \right)^{\omega} \right), \tag{2.3}$$

for $\mathbb{F}_2$, and

$$\text{Complexity}_{\text{GB},\mathbb{F}_q} = \mathcal{O}\left(\binom{n+D_{solv}}{n}^{\omega} \right), \tag{2.4}$$

when $q > D_{solv}$. In both cases $2 \leq \omega \leq 3$ is the linear algebra constant. A similar statement for a fixed, intermediate field $2 \leq q \leq D_{solv}$ can easily be deduced by counting monomials in $R_{q.n}$, up to and including degree D_{solv}.

We see that determining the solving degree for a given polynomial system $\mathcal{P}$ is a crucial component for estimating the complexity of computing a Gröbner basis. This is a question that has attracted much research attention. Unfortunately, determining D_{solv} seems in general to be as difficult as computing the Gröbner basis itself, but we are able to say more under some assumptions on $\mathcal{P}$. In the following subsections we will present two of the ideas for estimating D_{solv} that are most commonly used in the setting of big field schemes: the degree of semi–regularity, and the first fall degree. Before that, we will briefly discuss linear algebra algorithms.

Linear Algebra Complexities

The value used for the linear algebra constant ω in Equations (2.3) and (2.4) often has a significant impact when it comes to choosing secure parameters for multivariate schemes. The naive Gaussian method, $\omega = 3$, is outperformed by Strassen's method, which has an asymptotic value of $\omega = 2.81$ [104]. Note that the latter method is also in use for matrices of practical size. Over $\mathbb{F}_2$, further speed–ups may be obtained by the 'four Russians method' [12], but it does not improve upon the asymptotic value of Strassen. The currently best known asymptotic value is $\omega = 2.37$ [73], but it is unlikely that this can be implemented in practice, making it an example of a so–called "galactic algorithm"[75].

We have so far only mentioned algorithms for dense linear algebra, but the matrices arising from polynomial systems tend to be sparse. For instance, for quadratic systems, each row in a Macaulay matrix at any degree D over $\mathbb{F}_2$ has at most only $\binom{n}{2} + n + 1$ non zero entries. In some cases this observation can be used directly with sparse matrix algorithms, as we will see later in Section 2.2.5. The F$_4$ family of algorithms can also take advantage of sparse linear algebra, as can for instance be seen in the linear algebra package used in the FGb library [59, 56].

2.2.3 Semi–Regularity

The solving degree is well understood for an important class of polynomial systems, the *semi–regular sequences*, which has been studied by Bardet, Faugère, Salvy and Yang [14, 16]. We briefly describe some of the key results from these works, following the two important cases; the boolean polynomial ring, and when the underlying field $\mathbb{F}$ is large (depending on m and n). We start with the definition of the degree of regularity for these cases.

Definition 4 a) *Let $I^h = \langle p_1^h, \dots, p_m^h \rangle$ be the ideal of a homogeneous, overdetermined polynomial system in $\mathbb{F}[x_1, \dots, x_n]$, for any field $\mathbb{F}$. Then the* degree of regularity*, d_{reg},* *of I^h is defined as*

$$d_{reg} = d_{reg}(I^h) = \min \left\{ d \geq 0 \,\middle|\, \dim_{\mathbb{F}}(\{p \in I^h \mid \deg(p) = d\}) = \binom{n+d-1}{d} \right\}.$$

b) *Let now $I^h = \langle p_1^h, \ldots, p_m^h \rangle$ be the ideal of a homogeneous polynomial system in $\overline{R}_{2,n}$. Then the* degree of regularity*, d_{reg}, of I^h is defined as*

$$d_{reg} = d_{reg}(I^h) = \min \left\{ d \geq 0 \,\middle|\, \dim_{\mathbb{F}_2}(\{p \in I^h \mid \deg(p) = d\} = \binom{n}{d} \right\}.$$

Version a) of the definition can be of interest, even though we are concerned with solving polynomial systems over the ring $R_{q,n}$. This is the case if q is so large that computations involving the field equations $x_i^q - x_i = 0$ are infeasible. Indeed, we may instead compute a Gröbner basis over $\mathbb{F}_q[x_1, \ldots, x_n]$, and then recover a solution in $\mathbb{F}_q$ by changing monomial order and finding a root, as briefly mentioned earlier.

Definition 4 ensures that the leading terms of the elements of I^h will cover all monomials of degree $d_{reg}(I^h)$, which makes the degree of regularity an upper bound for the degree of the polynomials of a Gröbner basis of the homogeneous ideal I^h. We extend the definition to an affine ideal $I = \langle p_1, \ldots, p_m \rangle$, to be $d_{reg}(I) = d_{reg}(I^h)$, where $I^h = \langle p_1^h, \ldots, p_m^h \rangle$, and p_i^h the upper homogeneous part of p_i. Note that if p_i belongs to the Boolean polynomial ring, we will consider p_i^h over $\overline{R}_{2,n} = \mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2, \ldots, x_n^2 \rangle$. Semi–regular sequences can now be defined as systems that exhibit only trivial syzygies (see Section 2.2.4) up to the degree of regularity.

Definition 5 a) *A set of homogeneous polynomials $(p_1^h, \ldots, p_m^h)$ in $\mathbb{F}[x_1, \ldots, x_n]$, for any field $\mathbb{F}$, is said to be semi–regular if for all $1 \leq i \leq m$ and any $g \in \mathbb{F}[x_1, \ldots, x_n]$ satisfying*

$$gp_i^h \in \langle p_1^h, \ldots, p_{i-1}^h \rangle \quad \text{and} \quad deg(gp_i^h) < d_{reg},$$

then g is in the ideal $\langle p_1^h, \ldots, p_{i-1}^h \rangle$.

b) *A set of homogeneous polynomials $(p_1^h, \ldots, p_m^h)$ in $\overline{R}_{2,n}$ is said to be semi–regular if for all $1 \leq i \leq m$ and any $g \in \overline{R}_{2,n}$ satisfying*

$$gp_i^h \in \langle p_1^h, \ldots, p_{i-1}^h \rangle \quad \text{and} \quad deg(gp_i^h) < d_{reg},$$

then g is in the ideal $\langle p_1^h, \ldots, p_i^h \rangle$.

Note that in b), we require g to be in $\langle p_1^h, \ldots, p_i^h \rangle$ (as opposed to $\langle p_1^h, \ldots, p_{i-1}^h \rangle$ in a)). This is to include the homogeneous field syzygies $(p_i^h)^2 = 0$. As before, we will say that an affine polynomial system $(p_1, \ldots, p_m)$ is semi–regular, if $(p_1^h, \ldots, p_m^h)$ is.

The class of semi–regular sequences are well understood. The complexity of Matrix–F$_5$ for homogeneous semi–regular systems are given by Equations (2.3) and (2.4), with d_{reg} in place of D_{solv}. In practice, this estimate is also used for inhomogeneous semi–regular systems. Moreover, the Hilbert series of these systems are explicitly given, and the degree of regularity can be computed by finding the first nonpositive coefficient in these series (see [16]). If d_i denotes the degree of p_i^h, then these series are:

$$S_{m,n}(z) = \prod_{i=1}^{m}(1-z^{d_i})/(1-z)^n, \text{ for } \mathbb{F}[x_1,\ldots,x_n] \text{ and} \tag{2.5}$$

$$T_{m,n}(z) = (1+z)^n \Big/ \prod_{i=1}^{m}(1+z^{d_i}), \text{ for } \overline{R}_{2,n}. \tag{2.6}$$

It is worth pointing out that there is no easy way to a priori determine whether a general polynomial system is semi–regular. Indeed, since the definition of semi–regularity relies on the degree of regularity of the associated ideal one would potentially have to compute the ideal up to the degree predicted by the series in Equations (2.5) and (2.6). Hence, in many cases the question of determining semi–regularity of a system is about the same order as that of computing its Gröbner basis. However, there is a widespread belief that random polynomial systems (i.e., polynomials where the coefficient of each monomial up to some fixed degree is chosen uniformly at random from the underlying field) are semi–regular with a high probability. This observation has been supported by numerous experiments (see for instance the end of Section 3 in [14]). Some progress in this direction can be found in [99], where Semaev and Tenti prove that a class of random, overdetermined boolean systems will have d_{reg} as predicted by the series in (2.6), with probability tending to 1 as $n \to \infty$. Unfortunately, the polynomial systems we will be concerned with in this work are not covered by this result, and so we will have to rely on heuristic assumptions on the semi–regularity of random systems.

2.2.4 First Fall Polynomials

In [58], Faugère and Joux solved an instance of the HFE cryptosystem, expected to achieve 80–bit security, by Gröbner basis computation. The computation only needed to work with polynomials of degree ≤ 4, which is notably smaller than the degree of regularity expected for random systems of the same size. In particular, the authors noted that this is caused by a significant number of *degree fall polynomials*, i.e., combinations of polynomials occurring in the computation, where the highest degree cancels out, and the resulting polynomial is not reduced to 0 by any combination of previously considered polynomials. These new polynomials can further be used for creating new pairs in an F$_4$ algorithm, potentially leading to even more degree fall polynomials. If a sufficient number of them can be found, the whole Gröbner basis can be computed by only manipulating polynomials of low degree, which turned out to be the case for the HFE system.

The question of why the HFE polynomial system behaves so differently from random polynomial systems has been studied in several works. Most relevant for us is that of Dubois and Gama [51] as well as Ding and Hodges [44]. These works focus on determining at which degree the initial degree fall polynomials appear, known as the *first fall degree*, D_{ff}. We will present the definition according to [44][3], but first we fix some notation.

Let $\mathcal{P}^h = (p_1^h,\dots,p_m^h) \in \overline{R}_{q,n}^m$ denote the homogeneous quadratic part of the public key $\mathcal{P}$, and let $\left[\overline{R}_{q,n}\right]_\nu$ denote the graded ν–th part of $\overline{R}_{q,n}$. Then $\mathcal{P}^h$ induces a map:

$$\begin{array}{cccc} \psi^{\mathcal{P}^h}: & \overline{R}_{q,n}^m & \longrightarrow & \overline{R}_{q,n} \\ & (r_1,\dots,r_m) & \longmapsto & \sum_{i=1}^m r_i p_i^h, \end{array} \tag{2.7}$$

which in turn splits into graded maps $\psi_{\nu-2}^{\mathcal{P}^h} : \left[\overline{R}_{q,n}\right]_{\nu-2}^m \longrightarrow \left[\overline{R}_{q,n}\right]_\nu$. The $\overline{R}_{q,n}$–module $\mathrm{Syz}(\mathcal{P}^h)_\nu = \mathrm{Ker}(\psi_{\nu-2}^{\mathcal{P}^h})$ is known as the *ν–th grade of the (first) syzygy module of* $\mathcal{P}^h$. When $\nu = 4$, $\mathrm{Syz}(\mathcal{P}^h)_4$ will contain the *Koszul Syzygies*, which are generated by $(0,\dots,0,p_j^h,0,\dots,0,p_i^h,0,\dots,0)$ (p_j^h is in position i and p_i^h is in position j), and the *field syzygies*, which are generated by $(0,\dots,0,p_i^h,0,\dots,0)$ (p_i^h in position i). These syzygies correspond to the cancellations $p_j^h p_i^h + p_i^h p_j^h = 0$ and $(p_i^h)^2 = 0$, which we will refer to as the *trivial syzygies*. Moreover, define the submodule $\mathcal{T}(\mathcal{P}^h)_\nu \subseteq \mathrm{Syz}(\mathcal{P}^h)_\nu$ to be the ν–th graded component of the module generated by the Koszul and field syzygies, and denote $\mathcal{S}(\mathcal{P})_\nu = \mathrm{Syz}(\mathcal{P}^h)_\nu / \mathcal{T}(\mathcal{P}^h)_\nu$.

Definition 6 *The first fall degree associated with the quadratic polynomial system $\mathcal{P}$ is the natural number*

$$D_{ff} = min\{\, D \geq 2 \mid \mathcal{S}(\mathcal{P})_D \neq \mathbf{0} \,\}.$$

The first fall degree is generally less than or equal to the degree of regularity[4], and can as such be seen as a more conservative choice for security estimates. For certain cases

[3] The definitions found in the papers [51] and [44] use different quotient rings. Note that both works use the name "degree of regularity", which is unfortunate, seeing that the definitions are in general different from that of the degree of regularity introduced earlier by Bardet et al. This is why some later works prefer to use the name "first fall degree".

[4] An exception would be semi–regular sequences where the number of monomials of degree d_{reg} is equal to the number of non–trivial polynomial combinations at this degree, but these would be rare cases.

it is also possible to upper bound the first fall degree, by finding specific syzygies that will occur in the system. Such an upper bound was proved in [44] for HFE, while HFE- and HFEv/HFEv- was upper bounded by Ding–Kleinjung and Ding–Yang in [45] and [47], respectively. There is also a bound suggested by Petzoldt [94], which was found by extrapolating experimental results.

The First Fall Assumption and its Limitations

The first fall degree is often taken to be a good approximation for the solving degree of the HFE family of systems (particularly over $\mathbb{F}_2$), and, more generally, to the family of big field schemes. This is sometimes known as the *first fall assumption*, or *first fall heuristic*. For instance, both G*e*MSS [32] and GUI [41], the two HFEv- entries in the NIST competition, used Petzoldt's bound (the smallest in the literature) when estimating the solving degree for direct attacks. Cartor and Smith–Tone adapted ideas from [45] when estimating the solving degree for EFLASH [30].

This assumption should, however, be used with great care. It is easy to construct polynomial systems with a low first fall degree, but high solving degree, as discussed in [46]. Systems with this property can also arise naturally in cryptography. For instance, due to their simple algebraic structure, some of the ciphers discussed in Section 2.4 have modellings that yield a low first fall degree. Yet, it seems unlikely that the first fall degree is a useful tool in this setting. Even in the big field multivariate setting, some care is needed. For example, [46] performs experiments on variations of HFE where the $\mathbb{F}_q$–linear part is allowed to exceed D_{hfe}; the difference between the solving and first fall degree seem to increase in some of these cases (a similar behaviour is also observed for the *Extended Dob System*, in Paper 2).

Nevertheless, the use of the first fall degree is not without merit for big field schemes, especially when considered over $\mathbb{F}_2$. The central maps used in variants of HFE, C^*, and Dob can all be considered a simple polynomial that will generate many degree fall polynomials at a small degree. The number of these degree fall polynomials are more closely studied in Papers 1 and 2 for the C^* map and the Dob map, respectively. Intuitively speaking, these degree fall polynommials cause something of a 'chain reaction' when computing a Gröbner basis for such systems (over the base field), as more and more degree fall polynomials are being found from the previous ones. In the F$_4$ algorithm we typically observe this in the form of several successive steps are being performed at the first fall degree, until a solution is found. The use of modifiers distorts this to some extent, particularly by increasing the first fall degree, but the overall picture remains. In the experiments performed in Papers 1 and 2, we only observe cases where the solving degree is equal to either D_{ff} or $D_{ff}+1$, though we cannot rule out the possibility that there could be larger discrepancies for bigger parameters. Perhaps more concerning is the possibility that first fall polynomials might leak information about the secret modifiers, which was demonstrated and exploited in Paper 2. For these reasons, I strongly believe that the first fall degree is a crucial property to study for big field schemes over $\mathbb{F}_2$. The picture is different for other polynomial systems arising in cryptography. While a low first fall degree could potentially be a weakness here (and analysts should strive to understand its cause), one should be careful about jumping to conclusions regarding the complexity of computing Gröbner bases based on this alone.

We conclude this subsection by noting that there are two systems which we can prove that we are able to solve by only manipulating polynomials up to degree D_{ff}, without having to rely on any assumptions. The first is C^* where the linear polynomials used in Patarin's attack [91] will be found at degree 3. For the majority of choices of n and θ, we can expect enough linear polynomials to solve the system; see [40] for more information. The second example is nude Dob, which is dealt with in Appendix D of Paper 2.

2.2.5 Other Algorithms for Polynomial Solving

XL and its Variants

The 'eXtended Linearization' (XL) algorithm was introduced by Courtois et al., in [38], and is a rediscovery of the ideas of Lazard [72]. Assuming that there is a unique solution to the polynomial system, the core idea is to generate a large Macaulay matrix at some degree called the *operating degree*, D_o, directly from the given system, and find the solution by reducing this matrix. For generic systems, we typically have $D_o = d_{reg}$, or $D_o = d_{reg} + 1$ [112], and [8] notes that the matrices appearing in XL are larger than the matrices used in F_5. The reason why XL may still be important in certain settings lies in its simplicity; the Macaulay matrix is extremely sparse, and can be constructed directly from the polynomial system. Thus making it ideal to use sparse black box matrix solvers such as (improvements of) the Wiedemann algorithm [109]. Assuming that the Macaulay matrix is close to being square of size N, and has at most α nonzero entries in each row, then this algorithm is expected to recover a kernel vector in about $3N^2\alpha$ field multiplications. Moreover, the Macaulay matrix need not be stored, so this procedure may require less memory when compared to other methods.

The XL–Wiedemann approach is particularly useful in settings where we can reasonably expect that D_{solv}, d_{reg} and D_o coincide. Larger experimentation with XL–Wiedemann in this setting has been performed by [36]. As for the setting of big field schemes, recall from Section 2.2.4 that a typical behaviour of F_4 for these systems is to run several successive steps of degree (close to) D_{ff}, generating a multitude of degree fall polynomials, until a solution is found. This property is not exploited by the XL algorithm (although some variants aim to capture it, see e.g., [83]). Hence, the XL–Wiedemann procedure is in general unlikely to outperform F_4 for direct attacks against big field schemes.

Lastly, we note that the XL algorithm has inspired several different variations, an overview of which can be found in [112].

Hybrid Methods and the Crossbred Algorithm

Hybrid methods in the setting of polynomial system solving typically refers to methods that combine exhaustive search with Gröbner basis techniques or sparse linear algebra [21, 22, 15]. Fixing the value of some variables will reduce both the time and memory required when working with the matrices associated with the reduced systems. A guess that leads to no solutions is easily spotted, as it yields a Gröbner basis $G = \{1\}$. The drawback is that this process needs to be repeated about $\mathbb{F}_q^a$ times, where a is the number of guessed variables. In practice this is a useful strategy when the underlying field is small, and a drop in the solving degree can be obtained after fixing a few variables. For semi–regular systems, we can use Equations (2.5) and (2.6) with $n-a$ in place of n, under the additional assumption that the systems remain semi–regular after fixing variables. The effect of fixing variables is much harder to quantify for big field systems. The equations provided in Papers 1 and 2 predict how the number of degree fall polynomials changes as variables are fixed, and there are experimental results where this affects both the first fall degree and solving degree.

While the hybrid method can mainly be thought of as first fixing variables, and then reducing a large matrix, the crossbred algorithm of Joux and Vitse [68] is something of the opposite. The idea is to reduce a large Macaulay matrix in a precomputation step, in order to derive a system that is easier to solve in a brute force step. More specifically, the variables are split into two sets; X_g and X_l. The algorithm is typically run such that the polynomials of the derived system only contains monomials with degree 1 in the X_l–variables (the X_g–variables may have a much higher degree). The brute force step is then performed by fixing values for the variables in X_g, and the derived system is partially evaluated at this guess. This results in a linear system in the X_l–variables, and the validity of the guess is checked by finding a solution to this latter, linear system. There are many factors that comes into play when setting the parameters for this algorithm, and its complexity is highly dependent on the input polynomial system. Further discussions on this can be found in [68] or [89].

Other Techniques

So far we have discussed the main algorithms that have been used for solving polynomial systems from multivariate schemes. For completeness, we conclude this section with a short overview of two more polynomial systems solving methods that have been relevant to other cryptographic constructions.

Variable elimination for Boolean polynomials have been studied by Greve et al., in [65]. The behaviour of these techniques when applied to toy versions of the block ciphers LowMC [6] and Prince [25], has been investigated in [64]. A new approach for polynomial system solving was proposed by Lokshtanov et al., in 2017 [76]. While this algorithm was not considered to outperform other methods for parameters relevant to cryptography, this may have changed due to very recent (May 2021) improvements by Dinur [48]. In particular, Dinur uses this variant to break several instances of the LowMC cipher that is used in Picnic [113], an alternate signature candidate for the NIST PQC–competition.

2.3 Rank Attacks

The central maps discussed in Section 2.1 will all have a low Q–rank when seen as quadratic polynomials over the basis $\underline{X} = \left(X, X^q, \ldots, X^{q^{n-1}}\right)$. The idea to exploit this property was first suggested by Kipnis and Shamir, who studied the (unmodified) HFE system [69]. The ideas have since developed to an entire class of key recovery attacks, known as rank attacks.

2.3.1 Fundamental Concepts

To simplify the exposition, we let q be an odd prime for the remainder of this section. Let $\mathcal{F}' \in \mathbb{F}_{q^n}[X]/\langle X^{q^n} - X\rangle$ be a (homogeneous) $\mathbb{F}_q$–quadratic polynomial, and k an integer $0 \leq k \leq n-1$. Then $\mathbf{F}^{*k}$ will denote the symmetric matrix satisfying $(\mathcal{F}')^{q^k} = \underline{X}\mathbf{F}^{*k}\underline{X}^\top$. For instance, if $\mathcal{F}'$ is the HFE central map and $d = \lceil \log_q(D_{hfe}) \rceil$, then $\mathbf{F}^{*0}$ can only take non zero values in its upper left $d \times d$ submatrix, so it will, in particular, have rank at most d. More generally, $\mathbf{F}^{*k}$, will be zero, except at a $d \times d$ submatrix that is shifted k places to the right, and k places down (indices taken mod n).

Let $M \in \mathbb{F}_{q^n}^{n\times n}$ be an invertible matrix that is associated with an $\mathbb{F}_{q^n}$–basis over $\mathbb{F}_q$ (as defined in Proposition 2 of [23]), and consider the public key of an unmodified big field scheme, $(p_0, \ldots, p_{n-1}) = \mathcal{F} = T \circ \mathcal{F}' \circ S$. If $\mathbf{P}_i$ denotes the symmetric matrix associated with the public polynomial p_i, and $\mathbf{x} = (x_0, \ldots, x_{n-1})$, then we can write the public key as:

$$(\mathbf{x}\mathbf{P}_0\mathbf{x}^\top,\ldots,\mathbf{x}\mathbf{P}_{n-1}\mathbf{x}^\top) = (\mathbf{x}W\mathbf{F}^{*0}W^\top\mathbf{x}^\top,\ldots,\mathbf{x}W\mathbf{F}^{*(n-1)}W^\top\mathbf{x}^\top)M^{-1}T, \tag{2.8}$$

where $W = SM$ (see [23]). In the case of unmodified HFE, we see that $W\mathbf{F}^{*k}W^\top$ will have rank at most d, for any $0 \le k \le n-1$. Since T is invertible, this means that there will be linear combinations of the public matrices $\mathbf{P}_0,\ldots,\mathbf{P}_{n-1}$ that has rank at most d. Finding such a linear combination requires solving an instance of the MinRank problem (Definition 7). Once a solution to this problem is found, an attacker can use it to construct invertible matrices T', S', as well as an HFE central map $\mathcal{F}''$ of degree D_{hfe}, such that we can write the public key as $\mathcal{F} = T' \circ \mathcal{F}'' \circ S'$ [23]. The recovered T', S' and $\mathcal{F}''$ are not necessarily the same maps as the private key, but an attacker can nevertheless use it to decrypt/sign as efficiently as the legitimate user. Indeed, the recovered triplet is considered to be one of the *equivalent keys*, as studied in [111].

Other works have studied this approach with different central maps, as well as how various modifiers affect it; HFE- [107], HFEv- [95] and PFLASH/EFLASH [29].

The most time consuming step of the attack described in the previous section is that of solving an instance of the MinRank problem. We give a description of the search version of the problem in the following.

Definition 7 *(MinRank Problem). Let a positive integer r, and n_x matrices $M_i \in \mathbb{F}_q^{m\times n}$, for $0 \le i \le n_x - 1$ be given. The search version of the MinRank problem is to find a nontrivial set of constants $(x_0\ldots,x_{n_x-1}) \in \mathbb{F}_q^{n_x}$ such that*

$$\mathit{Rank}\left(\sum_{i=0}^{n_x-1} x_i M_i\right) \le r.$$

The problem is, in general, NP–complete [27], and several methods for solving it have been suggested in the literature: linear algebra search [62], minors modelling [60], KS–modelling [69], and support minors modelling [13].

2.3.2 Recent Developments of Rank Attacks

In November 2020, Tao, Petzoldt and Ding proposed a new version of the rank attack that bypasses the effect of both the minus and vinegar modifiers [105]. Indeed, the attack breaks the parameters of G*e*MSS, the HFEv- variant that made it to the third round of the NIST PQC–process. From Equation (2.8), we recall that each public matrix $\mathbf{P}_k$, $0 \le k \le n-1$ can be written as a linear combination of the matrices $W\mathbf{F}^{*i}W^\top$, $0 \le i \le n-1$. Due to the translation of the nonzero submatrix along $\mathbf{F}^{*0},\ldots,\mathbf{F}^{*n-1}$, the authors of [105] note that if $u \in \mathbb{F}_{q^n}^n$ is a vector such that uW has weight 1, then each vector $u\mathbf{P}_k$, $0 \leq k \leq n-1$ can be written as a linear combination of only d vectors. In particular, if we define $u\mathbf{P}^*$ to be the matrix having rows $u\mathbf{P}_k$, for $0 \leq k \leq n-1$, it will have rank at most d. Hence, u can be found by solving an instance of the MinRank problem. Note that this approach does not require T to be invertible, and so the minus modifier will not make the problem harder (we refer to [105] for details on how to include the vinegar modifiers). On the other hand, the attack requires W to be invertible, in order to ensure the existence of a vector u such that uW has weight 1. In Paper 3, we investigate how this attack works when W is not invertible, which will be the case for schemes using the projection modifier.

2.4 Symmetric Ciphers of Low Multiplicative Complexity

Modern cryptographic techniques, including Fully Homomorphic Encryption (FHE), Multi-Party Computation (MPC), and various proof systems, provide new settings for block ciphers and hash functions. The new protocols depend on different metrics than what has traditionally been the case for these primitives, such as strongly favouring a low multiplicative complexity or depth, or operating over particular fields. As a result, the "traditional" block ciphers and hash functions tend to be inefficient in these settings (see e.g., [6] for a discussion on AES in MPC protocols, or Section 6.2 in [17] for SHA2/SHA3 in the STARK setting). This observation motivated the development of new cryptographic primitives, starting in 2015 when Albrecht et al. designed LowMC, a block cipher with a small number of AND gates [6]. In 2016 Albrecht et al. suggested MiMC, which has a particularly simple algebraic description [2]. The MiMC[5] block cipher treats the plaintext as a single element in a large finite field, and each round consists of addition with a known constant, the secret key K, and the cubing operation. That is, for a number of rounds r, the i–th round function is simply

$$\mathcal{R}_i(x) = (X + K + C_i)^3, \quad \text{for } 1 \leq i \leq r, \tag{2.9}$$

for $K, C_i \in \mathbb{F}_q$, where $C_1, \ldots, C_r$ are known round constants. MiMC has furthermore inspired the variants GMiMC [5] and HadesMiMC [63], and primitives in the MARVELlous–family [10, 7] have followed the philosophy of utilizing a succinct algebraic description. In the following, we will refer to these constructions as "algebraically simple ciphers".

[5] Here, and in Paper 4, we focus on version MiMC–n/n (see [2]).

2.4.1 Security Analysis of Algebraically Simple Ciphers

The security of this new class of symmetric ciphers is not well understood, but we may still make a few, broad comments. Techniques that are widely used in traditional block cipher analysis, such as differential and linear attacks, seem to be largely inefficient in this new setting (see e.g., Section 4.2 of [2]). On the other hand, the simple algebraic description makes the designs potentially vulnerable to algebraic attacks. Indeed, the parameters of Jarvis and Friday, two of the initial members of the MARVELlous–family, were broken using Gröbner basis methods [4]. Another potential concern is the small degree of the round function. If the polynomial representation of the block cipher does not reach maximal degree, it could open the door for interpolation attacks [67, 74], the GCD–attack [2], and higher–order differential techniques [71, 70].

In Paper 4, we analyze the degree growth of MiMC, both in encryption and decryption directions, and discuss how it can be exploited.

Chapter 3

Overview of Papers

Paper I: Cryptanalysis of the Multivariate Encryption Scheme EFLASH

Morten Øygarden, Patrick Felke, Håvard Raddum, and Carlos Cid. Cryptographers Track at the RSA Conference, pages 85-105. Springer, 2020.

EFLASH [30] is a multivariate encryption scheme that uses the C^* central mapping, as well as the projection and minus modifiers. In this work, we analyze the behaviour of Gröbner basis algorithms on polynomial systems from variations of the EFLASH scheme. Our work goes beyond earlier analysis, in the sense that we examine all the relations that corresponds to degree fall polynomials. This allows us to not only estimate the first fall degree, but also the exact number of degree fall polynomial we will find, for step degrees 3 and 4. These estimates are then confirmed by experiments.

The novel approach presented in the paper yields a smaller upper bound in the first fall degree, than what was anticipated by the authors of EFLASH. Since our experiments furthermore seem to indicate that the solving degree is close to the first fall degree for EFLASH polynomial systems, we conclude that the parameters suggested for this scheme are too optimistic. For instance, we estimate that parameters proposed for 80–bit security, will achieve at most 69 bits of security.

Paper II: Analysis of Multivariate Encryption Schemes: Application to Dob

Morten Øygarden, Patrick Felke, and Håvard Raddum. In International Conference on Public-Key Cryptography (PKC), pages 155-183. Springer, 2021. Invited to the Journal of Cryptology.

Paper 2 contains several contributions. The first part can be seen as a generalization of the ideas from Paper 1[1]. We continue to analyze the number of degree fall polynomials, not just the degree where they appear. The considered setting is now that of general big field schemes with modifications. The approach is to first estimate the dimension of the space of degree fall polynomials caused by the unmodified central map, and then account for the various modifiers. The motivating example is the Dob encryption scheme; the encryption variant of the larger Two–Face family suggested by Macario–Rat and Patarin [77]. Based on the Dobbertin permutation, it relies on the *ip* and Q_+ modifications to achieve security. The resulting estimate of the number of degree fall polynomials in variations of the Dob encryption scheme is seen to be precise in the experiments we have performed. This holds true, even if an attacker fixes the value for a number of variables.

The second part of Paper 2 describes a novel attack against the Dob construction. With the new understanding of how modifications affect degree fall polynomials, an attacker can recover certain polynomials that carry information about the secret modifiers. This allows the attacker to piece together the homogeneous quadratic part of the modifier polynomials, which can then be used to speed up decryption. Finally, we discuss how this approach makes the proposed parameters for the Dob encryption scheme fall short of 80–bit security.

Paper III: On the Effect of Projection on Rank Attacks in Multivariate Cryptography

Morten Øygarden, Daniel Smith–Tone, and Javier Verbel. To appear in PQCrypto: International Conference on Post-Quantum Cryptography, 2021

In [105], Tao, Petzoldt and Ding propose a new rank attack against the HFEv- signature scheme. While it is straightforward to see that the attack can be generalized to include other big field variations utilizing vinegar and minus, the idea relies on the fact that the input matrix, *S* is invertible. Hence, it is not clear how the projection modifier affects the attack. This is the main question that is explored in Paper 3. We focus on the signature schemes PFLASH, as well as HFEv- with projection (pHFEv-). We prove upper bounds on the rank of the MinRank part of the attack in both settings. The central maps are different enough to warrant distinct approaches, even though the end result is that the rank increases by p, where p is the projection modifier, for both schemes. In particular, this proves that the new rank attack breaks the suggested parameters for PFLASH.

The upper bounds are observed to be tight in the experiments we are able to run.

[1] This generalization is made explicit in Section 4.1 of this Thesis.

Finally, for pHFEv-, we compare the impact of projection, with that of increasing the HFE degree.

Paper IV: An Algebraic Attack on Ciphers with Low–Degree Round Functions: Application to Full MiMC

Maria Eichlseder, Lorenzo Grassi, Reinhard Lüftenegger, Morten Øygarden, Christian Rechberger, Markus Schofnegger, and Qingju Wang. International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt), pages 477–506. Springer, 2020.

MiMC is an algebraically simple permutation, proposed by Albrecht et al. in 2016 [2]. In this paper we analyze how the algebraic degree of binary MiMC–n/n grows with the number of rounds. This allows us to lower bound the number of rounds before the maximal algebraic degree can be reached – in both encryption and decryption direction. From these observations we discuss known and secret key distinguishers on binary MiMC.

The large number of rounds needed to reach maximal algebraic degree in the decryption direction, coupled with the simple algebraic description in the encryption direction, furthermore allows us to design a key recovery attack on block cipher versions of MiMC. While the attack is impractical, requiring half the code book, it is nevertheless the first attack described for full round MiMC.

Chapter 4

Conclusions

This chapter summarizes and concludes on the work that was performed for big field cryptosystems, namely Papers 1 – 3. From here on, it is assumed that the reader is familiar with these works, which can be found in Chapter 5. We start by making explicit the relation between Papers 1 and 2. In Section 4.2 I will discuss the recent trends in big field multivariate cryptography, and how the research presented in this thesis fits into this bigger picture. Finally, I list some of the open problems in this direction.

4.1 Connecting Papers 1 and 2

We will now see how the machinery of Paper 2 generalizes the results of Paper 1. Following the notation of Paper 2, let $\mathcal{P}$ be the public polynomials of a fixed instance of EFLASH with parameters d, n and a. Write $m = d - a$ for the number of public polynomials, and let M be the ideal generated by the a removed polynomials (considered over $\overline{B}(n)$). As in Equation (10) of Paper 2, we are interested in determining

$$N_\nu = \dim_\nu(\mathcal{S}(\mathcal{F})) + \dim_\nu(\mathcal{P}_M) - \dim_\nu(M), \tag{4.1}$$

for $\nu = 3, 4$. We expect $\dim_3(\mathcal{S}(\mathcal{F})) = 3d$, due to the polynomials associated with α, β_1 and β_2, as described in Paper 1. Furthermore, $\dim_3(\mathcal{P}_M) = 0$ and $\dim_3(M) = na$. Hence, Equation (4.1) predicts $N_3 = 3d - na$, which is the same as Equation (10) in Paper 1.

In Section 4.3 of Paper 1, we show that $\dim_4(\mathcal{S}(\mathcal{F})) = (3n-9)d$. Moreover, we expect $\dim_4(\mathcal{P}_M) = am$, by considering the possible products of public and removed polynomials. If a is small compared to d, we expect the generators of M to form a

4–semi–regular system, which yields $\dim_4(M) = a\binom{n}{2} - \binom{a+1}{2}$. Hence,

$$N_4 = (3n-9)d + am - a\binom{n}{2} + \binom{a+1}{2}.$$

Recalling that $m = d - a$, this is the same as Equation (12) of Paper 1.

The behaviour for EFLASH that was observed in Table 3 of Paper 1 seems analogous to the different variants of $N_v^{(\alpha,\beta)}$ observed for the Dob system, though I am still unsure about its exact cause.

4.2 Impact on Big Field Cryptography

It is well–known that the simple central maps used in big field cryptosystems make the constructions vulnerable to rank attacks and Gröbner basis methods. Certain modifiers have, up until now, been considered reasonable countermeasures; minus and *ip* have been popular for encryption schemes, whereas minus and vinegar are typically used for signatures.

Papers 1 and 2 cast doubt on the viability of this strategy for encryption schemes. One of the conclusions we can draw is that the modifiers used in this setting are not as effective as previously believed. The gluing technique discussed in Section 6 of Paper 2, is not restricted to the Dobbertin permutation, and I expect it to also be applicable for other big field constructions that exhibit similar behaviour in terms of degree fall polynomials. Signature schemes fare more favourably against this attack, since the number of modifiers has only a small impact on the signing time. A case in point would be the signature variant of Dob [77], where the proposed 128–bit parameters suggest using $d = 257$ for the extension field degree, and then remove 129 of the public polynomials. I doubt that the ideas from Paper 2 will be able to break these parameters. On the other hand, the new rank attack of Tao, Petzoldt and Ding [105] pose a serious threat to signature schemes. Indeed, neither minus, nor the vinegar modifier provide much security against this idea. However, Paper 3 shows that projection increases the rank of such an attack. This is a modification that does not increase the decryption time for encryption schemes, but it provides an exponential increase to the signing time. Moreover, it does not increase the security against methods related to Gröbner basis techniques. Hence, this is something of a converse situation to what we saw from Papers 1 and 2 with the minus modifier. If we wish to secure big field constructions with the "traditional" choices of modifiers discussed in this thesis, it would seem that we need to apply a significant amount of both projection *and* minus modifiers (or a different combination of modifiers with similar behaviour). It is an open question whether there will be any encryption or signature big field scheme that remain efficient under this restriction.

All of the “classical” big field constructions discussed in this thesis follow the same pattern. The central map is quadratic, and the modifiers can be applied by composing with specialized[1] affine maps over R_q. The recent advances in cryptanalysis against such schemes have motivated some authors to break this mold. Indeed, in January 2021, Macario–Rat and Patarin proposed using cubic polynomials, and discusses new modifiers in this setting [78]. Smith–Tone, in a work published in April 2021 [103], retains the quadratic central map, but suggests a new modifier that is fundamentally nonlinear. These novel designs for big field schemes are not covered by the above discussion, and it will be interesting to see if they will hold up to third party analysis.

4.3 Open Problems

Listed below are the main open problems in big field cryptography that I have identified during my research.

- A possible direction to advance the analysis presented in Paper 2 is to include different central maps and modifiers. A few natural examples include the HFE central map and the vinegar modifier. Expanding the discussion on the minus modifier that was briefly presented in Section 4.1 of this thesis, could also be included here.

- One drawback of Paper 1 (resp. Paper 2), is that we are only able to give formulas predicting the number of degree fall polynomials up to degree 4 (resp. 5). Finding exact formulas for larger degrees seems to be difficult, but determining reasonable upper bounds might be more feasible. Such bounds would provide insight into the complexity of various attacks, and would, in turn, be valuable for designing new schemes. This could, for instance, help determine how secure instances of PFLASH would look like, which was left as an open question in Paper 3.

- The complexity of the attack in Paper 2 is not dominated by the process of finding information about the secret modifiers. Rather, we are limited by the second step where we, somewhat naively, apply this extra information in a polynomial system solving step. The reason why this second step is still expensive, is that the linear parts of the Q_+ polynomials are still unknown. It would be interesting to see if this second step could be improved. Since an attacker now knows the homogeneous quadratic part of the secret structure, a promising approach might be to

[1] To be more specific, maps that can be described component wise as $\mathbf{x} \mapsto \sum c_i x_i + r$, for constants $c_i \in \mathbb{F}_q$, and r a specialized element in $R_{q,n}$ of degree at most 2.

proceed with rank techniques. This would yield "equivalent" matrices S' and T'. The corresponding univariate polynomial we obtain after composing with these matrices, will still have a high $\mathbb{F}_q$–linear degree, which is related to the missing linear components. This information might be used to recover the entirety of the Q_+ polynomials.

- For completeness, it would also be interesting to see how other modifiers affect the new rank attack of [105], similar to the analysis performed for the projection modifier in Paper 3. For instance, the Q_+ modifier adds t random polynomials to the system, and I would, a priori, expect it to increase the rank of the attack by t. At first glance, the *ip* modifier adds $\binom{k}{2}$ different polynomials. However, the Q–rank of each of these polynomials will be restricted to k, and since they are generated by the same linear forms, I expect the modifier to only increase the rank of the attack by k. This intuition should be paired with a more rigorous examination, as well as experiments, before drawing any final conclusions.
- Finally, it seems to me that the design of new modifiers, specifically aimed at thwarting the improved rank and Gröbner basis techniques, will play a crucial role if big field cryptosystems are to be successful in the future. The recent works [78, 103] mentioned at the end of the previous section can be seen as examples in this direction. I believe that further design of such modifiers, as well as analysis of them, is a promising research direction for the future.

Bibliography

[1] Alagic G, Alperin-Sheriff J, Apon D, Cooper D, Dang Q, Kelsey J, Liu Y-K, Miller C, Moody D, Peralta R, Perlner R, Robinson A, Smith-Tone D, (2019) Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8309. https://doi.org/10.6028/NIST.IR.8309. 1.2.2

[2] M. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 191–219. Springer, 2016. 2.4, 5, 2.4.1, 3

[3] M. R. Albrecht, C. Cid, L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, and M. Schofnegger. Algebraic cryptanalysis of STARK-friendly designs: application to MARVELlous and MiMC. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 371–397. Springer, 2019. 2.2.1

[4] M. R. Albrecht, C. Cid, L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, and M. Schofnegger. Algebraic cryptanalysis of STARK-friendly designs: application to MARVELlous and MiMC. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 371–397. Springer, 2019. 2.4.1

[5] M. R. Albrecht, L. Grassi, L. Perrin, S. Ramacher, C. Rechberger, D. Rotaru, A. Roy, and M. Schofnegger. Feistel structures for MPC, and more. In *European Symposium on Research in Computer Security*, pages 151–171. Springer, 2019. 2.4

[6] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. In *Annual International Conference on the Theory and*

Applications of Cryptographic Techniques, pages 430–454. Springer, 2015. 1.4, 2.2.5, 2.4

[7] A. Aly, T. Ashur, E. Ben-Sasson, S. Dhooghe, and A. Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Transactions on Symmetric Cryptology*, pages 1–45, 2020. 2.4

[8] G. Ars, J.-C. Faugère, H. Imai, M. Kawazoe, and M. Sugita. Comparison between XL and Gröbner basis algorithms. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 338–353. Springer, 2004. 2.2.5

[9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. 1.2.1

[10] T. Ashur and S. Dhooghe. MARVELlous: a STARK-Friendly Family of Cryptographic Primitives. Cryptology ePrint Archive, Report 2018/1098, 2018. https://eprint.iacr.org/2018/1098. 2.4

[11] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven. Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX Quantum*, 2(1), 2021. 1.2.1

[12] G. V. Bard. Accelerating cryptanalysis with the method of four russians. *Cryptology ePrint Archive: Report 2006/251*, 2006. 2.2.2

[13] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. Perlner, D. Smith-Tone, J.-P. Tillich, and J. Verbel. Improvements of Algebraic Attacks for solving the Rank Decoding and MinRank problems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 507–536. Springer, 2020. 2.3.1

[14] M. Bardet, J.-C. Faugère, and B. Salvy. Complexity of Gröbner basis computation for Semi-regular Overdetermined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$. 2003. [Research Report] RR-5049, INRIA, inria-00071534. 2.2.3, 2.2.3

[15] M. Bardet, J.-C. Faugère, B. Salvy, and P.-J. Spaenlehauer. On the complexity of solving quadratic boolean systems. *Journal of Complexity*, 29(1):53–75, 2013. 2.2.5

[16] M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In *Proc. of MEGA*, volume 5, 2005. 2.2.3, 2.2.3

[17] E. Ben-Sasson, L. Goldberg, and D. Levit. STARK Friendly Hash – Survey and Recommendation. Cryptology ePrint Archive, Report 2020/948, 2020. https://eprint.iacr.org/2020/948. 1.4, 2.4

[18] E. R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of computation*, 24(111):713–735, 1970. 2

[19] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. High-speed high-security signatures. *Journal of cryptographic engineering*, 2(2):77–89, 2012. 1.1

[20] D. J. Bernstein and B.-Y. Yang. Asymptotically faster quantum algorithms to solve multivariate quadratic equations. In *International Conference on Post-Quantum Cryptography*, pages 487–506. Springer, 2018. 1.3

[21] L. Bettale, J.-C. Faugère, and L. Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2009. 2.2.5

[22] L. Bettale, J.-C. Faugère, and L. Perret. Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, pages 67–74, 2012. 2.2.5

[23] L. Bettale, J.-C. Faugère, and L. Perret. Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. *Designs, Codes and Cryptography*, 69(1):1–52, 2013. 2.3.1, 2.3.1

[24] W. Beullens. Improved Cryptanalysis of UOV and Rainbow. Cryptology ePrint Archive, Report 2020/1343, 2020. https://eprint.iacr.org/2020/1343. 1.3

[25] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, et al. PRINCE - a low-latency block cipher for pervasive computing applications. In *International conference on the theory and application of cryptology and information security*, pages 208–225. Springer, 2012. 2.2.5

[26] B. Buchberger. Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. *PhD thesis, University of Innsbruck*, 1965. 2.2.2

[27] J. F. Buss, G. S. Frandsen, and J. O. Shallit. The computational complexity of some problems of linear algebra. *Journal of Computer and System Sciences*, 58(3):572–596, 1999. 2.3.1

[28] A. Caminata and E. Gorla. Solving multivariate polynomial systems and an invariant from commutative algebra. *arXiv preprint arXiv:1706.06319*, 2017. 2.2.1

[29] R. Cartor and D. Smith-Tone. An updated security analysis of PFLASH. In *International Workshop on Post-Quantum Cryptography*, pages 241–254. Springer, 2017. 2.1.1, 2.3.1

[30] R. Cartor and D. Smith-Tone. EFLASH: A New Multivariate Encryption Scheme. In C. Cid and M. Jacobson Jr., editors, *Selected Areas in Cryptography – SAC 2018*, volume 11349 of *Lecture Notes in Computer Science*, pages 281–299. Springer International Publishing, 2019. 1, 2.2.4, 3

[31] R. Cartor and D. Smith-Tone. All in the C^* family. *Designs, Codes and Cryptography*, 88(6):1023–1036, 2020. 2.1.2

[32] A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. GeMSS: A Great Multivariate Short Signature. Technical report, National Institute of Standards and Technology, 2019. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions. 1.3, 2.2.4

[33] O. Chakraborty. *Design And Cryptanalysis of Post Quantum Cryptosystems*. PhD thesis, Sorbonne Université, 2020. 2.1.2

[34] M.-S. Chen, J. Ding, M. Kannwischer, J. Patarin, A. Petzoldt, D. Schmidt, and B.-Y. Yang. Response to Recent Paper by Ward Beullens. Sent to the NIST Post-Quantum Cryptography Email list, 25th of December, 2020. http://precision.moscito.org/by-publ/recent/response-ward.pdf. 1.3

[35] M.-S. Chen, B.-Y. Yang, and D. Smith-Tone. PFLASH-secure asymmetric signatures on smart cards. In *Lightweight Cryptography Workshop*, 2015. 1

[36] C.-M. Cheng, T. Chou, R. Niederhagen, and B.-Y. Yang. Solving quadratic equations with XL on parallel architectures. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 356–373. Springer, 2012. 2.2.5

[37] C.-M. Cheng, Y. Hashimoto, H. Miura, and T. Takagi. A polynomial-time algorithm for solving a class of underdetermined multivariate quadratic equations over fields of odd characteristics. In *International Workshop on Post-Quantum Cryptography*, pages 40–58. Springer, 2014. 2.2

[38] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 392–407. Springer, 2000. 2.2.5

[39] D. A. Cox, J. Little, and D. O'shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006. 2.2.1

[40] A. Diene, J. Ding, J. E. Gower, T. J. Hodges, and Z. Yin. Dimension of the linearization equations of the Matsumoto-Imai cryptosystems. In *International Workshop on Coding and Cryptography*, pages 242–251. Springer, 2005. 2.2.4

[41] J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, and B.-Y. Yang. GUI. Technical report, National Institute of Standards and Technology, 2017. https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions. 2.2.4

[42] J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, B.-Y. Yang, M. Kannwischer, and J. Patarin. Rainbow (round 3 submission). Technical report, National Institute of Standards and Technology, 2020. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions. 1.3

[43] J. Ding, V. Dubois, B.-Y. Yang, O. C.-H. Chen, and C.-M. Cheng. Could SFLASH be repaired? In *International Colloquium on Automata, Languages, and Programming*, pages 691–701. Springer, 2008. 2.1.1

[44] J. Ding and T. J. Hodges. Inverting HFE systems is quasi-polynomial for all fields. In *Annual Cryptology Conference*, pages 724–742. Springer, 2011. 2.2.4, 3, 2.2.4

[45] J. Ding and T. Kleinjung. Degree of regularity for HFE-. *Cryptology ePrint Archive: Report 2011/570*, 2011. 2.2.4, 2.2.4

[46] J. Ding and D. Schmidt. Solving degree and degree of regularity for polynomial systems over a finite fields. In *Number Theory and Cryptography*, pages 34–49. Springer, 2013. 2.2.4

[47] J. Ding and B.-Y. Yang. Degree of regularity for HFEv and HFEv-. In *International Workshop on Post-Quantum Cryptography*, pages 52–66. Springer, 2013. 2.2.4

[48] I. Dinur. Cryptanalytic Applications of the Polynomial Method for Solving Multivariate Equation Systems over GF(2). Cryptology ePrint Archive, Report 2021/578, 2021. https://eprint.iacr.org/2021/578. 2.2.5

[49] V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern. Practical cryptanalysis of SFLASH. In *Annual International Cryptology Conference*, pages 1–12. Springer, 2007. 2.1.1

[50] V. Dubois, P.-A. Fouque, and J. Stern. Cryptanalysis of SFLASH with slightly modified parameters. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 264–275. Springer, 2007. 2.1.1

[51] V. Dubois and N. Gama. The degree of regularity of HFE systems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 557–576. Springer, 2010. 2.2.4, 3

[52] C. Eder and J.-C. Faugère. A survey on signature-based algorithms for computing Gröbner bases. *Journal of Symbolic Computation*, 80:719–784, 2017. 2.2.2

[53] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985. 1.1

[54] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999. 2.2.2

[55] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83, 2002. 2.2.2

[56] J.-C. Faugère. FGb: A Library for Computing Gröbner Bases. In K. Fukuda, J. Hoeven, M. Joswig, and N. Takayama, editors, *Mathematical Software - ICMS 2010*, volume 6327 of *Lecture Notes in Computer Science*, pages 84–87, Berlin, Heidelberg, September 2010. Springer Berlin / Heidelberg. 2.2.2

[57] J.-C. Faugère, K. Horan, D. Kahrobaei, M. Kaplan, E. Kashefi, and L. Perret. Fast quantum algorithm for solving multivariate quadratic equations. *arXiv preprint arXiv:1712.07211*, 2017. 1.3

[58] J.-C. Faugère and A. Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using gröbner bases. In *Annual International Cryptology Conference*, pages 44–60. Springer, 2003. 2.1.2, 2.2.4

[59] J.-C. Faugère and S. Lachartre. Parallel gaussian elimination for gröbner bases computations in finite fields. In *Proceedings of the 4th International Workshop on Parallel and Symbolic Computation*, pages 89–97, 2010. 2.2.2

[60] J.-C. Faugère, F. Levy-dit Vehel, and L. Perret. Cryptanalysis of MinRank. In *Annual International Cryptology Conference*, pages 280–296. Springer, 2008. 2.3.1

[61] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979. 1.3

[62] L. Goubin and N. T. Courtois. Cryptanalysis of the TTM cryptosystem. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 44–57. Springer, 2000. 2.3.1

[63] L. Grassi, R. Lüftenegger, C. Rechberger, D. Rotaru, and M. Schofnegger. On a generalization of substitution-permutation networks: The HADES design strategy. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 674–704. Springer, 2020. 2.4

[64] B. Greve, Ø. Ytrehus, and H. Raddum. Variable Elimination - a Tool for Algebraic Cryptanalysis. Cryptology ePrint Archive, Report 2019/112, 2019. https://eprint.iacr.org/2019/112. 2.2.5

[65] B. Greve, Ø. Ytrehus, H. Raddum, and G. Fløystad. Solving non-linear Boolean equation systems by variable elimination. *Applicable Algebra in Engineering, Communication and Computing*, pages 1–45, 2019. 2.2.5

[66] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical review letters*, 79(2):325, 1997. 1.2.1

[67] T. Jakobsen and L. R. Knudsen. The interpolation attack on block ciphers. In *International Workshop on Fast Software Encryption*, pages 28–40. Springer, 1997. 2.4.1

[68] A. Joux and V. Vitse. A crossbred algorithm for solving boolean polynomial systems. In *International Conference on Number-Theoretic Methods in Cryptology*, pages 3–21. Springer, 2017. 2.2.5

[69] A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Annual International Cryptology Conference*, pages 19–30. Springer, 1999. 2.3, 2.3.1

[70] L. R. Knudsen. Truncated and higher order differentials. In *International Workshop on Fast Software Encryption*, pages 196–211. Springer, 1994. 2.4.1

[71] X. Lai. Higher order derivatives and differential cryptanalysis. In *Communications and cryptography*, pages 227–233. Springer, 1994. 2.4.1

[72] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *European Conference on Computer Algebra*, pages 146–156. Springer, 1983. 2.2.2, 2.2.5

[73] F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303, 2014. 2.2.2

[74] C. Li and B. Preneel. Improved interpolation attacks on cryptographic primitives of low algebraic degree. In *International Conference on Selected Areas in Cryptography*, pages 171–193. Springer, 2019. 2.4.1

[75] R. J. Lipton and K. W. Regan. David johnson: Galactic algorithms. In *People, Problems, and Proofs*, pages 109–112. Springer, 2013. 2.2.2

[76] D. Lokshtanov, R. Paturi, S. Tamaki, R. Williams, and H. Yu. Beating brute force for systems of polynomial equations over finite fields. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2190–2202. SIAM, 2017. 2.2.5

[77] G. Macario-Rat and J. Patarin. Two-face: New public key multivariate schemes. In *International Conference on Cryptology in Africa*, pages 252–265. Springer, 2018. 2.1, 2.1.2, 3, 4.2

[78] G. Macario-Rat and J. Patarin. Ariadne Thread and Salt: New Multivariate Cryptographic Schemes with Public Keys in Degree 3. Cryptology ePrint Archive, Report 2021/084, 2021. https://eprint.iacr.org/2021/084. 4.2, 4.3

[79] R. H. Makarim and M. Stevens. M4GB: An efficient Gröbner-basis algorithm. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, pages 293–300, 2017. 2.2.2

[80] T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and C. G. Günther, editors, *Advances in Cryptology — EUROCRYPT '88*, pages 419–453, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg. 1.3, 1

[81] Microsoft Security Advisory 2862973. Update for Deprecation of MD5 Hashing Algorithm for Microsoft Root Certificate Program. Version 3.0. (Retrieved 04.03.2021), 2014. https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2014/2862973. 1.2.1

[82] H. Miura, Y. Hashimoto, and T. Takagi. Extended algorithm for solving underdefined multivariate quadratic equations. In *International Workshop on Post-Quantum Cryptography*, pages 118–135. Springer, 2013. 2.2

[83] M. S. E. Mohamed, W. S. A. E. Mohamed, J. Ding, and J. Buchmann. MXL2: Solving polynomial equations over GF(2) using an improved mutant strategy. In *International Workshop on Post-Quantum Cryptography*, pages 203–215. Springer, 2008. 2.2.5

[84] National Academies of Sciences, Engineering, and Medicine. *Quantum Computing: Progress and Prospects*. The National Academies Press, Washington, DC, 2019. 1.2.1

[85] National Bureau of Standards. Data Encryption Standard. FIPS 46. January 15, 1977. https://csrc.nist.gov/publications/detail/fips/46/archive/1977-01-15. 1.1

[86] National Institute of Standards and Technology. Advanced Encryption Standard (AES). FIPS 197. November 2001. https://csrc.nist.gov/publications/detail/fips/197/final. 1.1

[87] National Institute of Standards and Technology. Secure Hash Standard (SHS). FIPS PUB 180-4. August 2015. https://csrc.nist.gov/publications/detail/fips/180/4/final. 1.1

[88] National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. FIPS 202. August 2015. https://csrc.nist.gov/publications/detail/fips/202/final. 1.1

[89] R. Niederhagen, K.-C. Ning, and B.-Y. Yang. Implementing Joux-Vitses Crossbred Algorithm for Solving $\mathcal{MQ}$ Systems over $\mathbb{F}_2$ on GPUs. In *International Conference on Post-Quantum Cryptography*, pages 121–141. Springer, 2018. 2.2.5

[90] J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt88. In *Annual International Cryptology Conference*, pages 248–261. Springer, 1995. 1.3, 1, 2.1.2

[91] J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt88. In *Annual International Cryptology Conference*, pages 248–261. Springer, 1995. 2.2.4

[92] J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 33–48. Springer, 1996. 2

[93] J. Patarin, N. Courtois, and L. Goubin. Flash, a fast multivariate signature algorithm. In *Cryptographers Track at the RSA Conference*, pages 298–307. Springer, 2001. 2.1.1

[94] A. Petzoldt. On the Complexity of the Hybrid Approach on HFEv-. Cryptology ePrint Archive, Report 2017/1135, 2017. https://eprint.iacr.org/2017/1135. 2.2.4

[95] A. Petzoldt, M.-S. Chen, B.-Y. Yang, C. Tao, and J. Ding. Design principles for HFEv-based multivariate signature schemes. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 311–334. Springer, 2015. 2.3.1

[96] "Post-Quantum Cryptography: Proposed Requirements and Evaluation Criteria, 81 Federal Register 50686 (August 2, 2016), pp. 50686-50687. https://federalregister.gov/a/2016-18150. 1.2.2

[97] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. 1.1

[98] P. Schwabe and B. Westerbaan. Solving binary $\mathcal{MQ}$ with Grovers algorithm. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 303–322. Springer, 2016. 1.3

[99] I. Semaev and A. Tenti. Probabilistic analysis on Macaulay matrices over finite fields and complexity of constructing Gröbner bases. *Journal of Algebra*, 565:651–674, 2021. 2.2.3

[100] C. E. Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, 1949. 1.1

[101] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. IEEE, 1994. 1.2.1

[102] D. Smith-Tone. Properties of the discrete differential with cryptographic applications. In *International Workshop on Post-Quantum Cryptography*, pages 1–12. Springer, 2010. 2.1.1

[103] D. Smith-Tone. New Practical Multivariate Signatures from a Nonlinear Modifier. Cryptology ePrint Archive, Report 2021/429, 2021. https://eprint.iacr.org/2021/429. 4.2, 4.3

[104] V. Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969. 2.2.2

[105] C. Tao, A. Petzoldt, and J. Ding. Improved Key Recovery of the HFEv- Signature Scheme. Cryptology ePrint Archive, Report 2020/1424, 2020. https://eprint.iacr.org/2020/1424. 1.3, 2.3.2, 3, 4.2, 4.3

[106] E. Thomae and C. Wolf. Solving underdetermined systems of multivariate quadratic equations revisited. In *International Workshop on Public Key Cryptography*, pages 156–171. Springer, 2012. 2.2

[107] J. Vates and D. Smith-Tone. Key recovery attack for all parameters of HFE- . In *International Workshop on Post-Quantum Cryptography*, pages 272–288. Springer, 2017. 2.3.1

[108] X. Wang, D. Feng, X. Lai, and H. Yu. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. Cryptology ePrint Archive, Report 2004/199, 2004. https://eprint.iacr.org/2004/199. 1.2.1

[109] D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE transactions on information theory*, 32(1):54–62, 1986. 2.2.5

[110] C. Wolf and B. Preneel. Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations. Cryptology ePrint Archive, Report 2005/077, 2005. https://eprint.iacr.org/2005/077. 2.1.2

[111] C. Wolf and B. Preneel. Equivalent keys in $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. *Journal of Mathematical Cryptology*, 4(4):375–415, 2011. 2.3.1

[112] B.-Y. Yang and J.-M. Chen. All in the XL family: Theory and practice. In *International Conference on Information Security and Cryptology*, pages 67–86. Springer, 2004. 2.2.5

[113] G. Zaverucha, M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, J. Katz, X. Wang, V. Kolesnikov, and D. Kales. Picnic (round 3 submission). Technical report, National Institute of Standards and Technology, 2020. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions. 2.2.5

[114] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020. 1.2.1

Chapter 5

Scientific Papers

Paper I

Cryptanalysis of the Multivariate Encryption Scheme EFLASH

Morten Øygarden, Patrick Felke, Håvard Raddum, and Carlos Cid.
In *Cryptographers Track at the RSA Conference*, pages 85-105. Springer, 2020.

Cryptanalysis of the Multivariate Encryption Scheme EFLASH

Morten Øygarden[1], Patrick Felke[2], Håvard Raddum[1], and Carlos Cid[1,3]

[1] Simula UiB
[2] University of Applied Sciences Emden-Leer
[3] Royal Holloway University of London
{morten.oygarden,haavardr}@simula.no,
patrick.felke@hs-emden-leer.de,
carlos.cid@rhul.ac.uk

Abstract. EFLASH is a multivariate public-key encryption scheme proposed by Cartor and Smith-Tone at SAC 2018. In this paper we investigate the hardness of solving the particular equation systems arising from EFLASH, and show that the solving degree for these types of systems is much lower than estimated by the authors. We show that a Gröbner basis algorithm will produce *degree fall polynomials* at a low degree for EFLASH systems. In particular we are able to accurately predict the number of these polynomials occurring at step degrees 3 and 4 in our attacks. We performed several experiments using the computer algebra system MAGMA, which indicate that the solving degree is at most one higher than the one where degree fall polynomials occur; moreover, our experiments show that whenever the predicted number of degree fall polynomials is positive, it is exact. Our conclusion is that EFLASH does not offer the level of security claimed by the designers. In particular, we estimate that the EFLASH version with 80-bit security parameters offers at most 69 bits of security.

1 Introduction

Public-key cryptosystems whose security is based on the hardness of solving multivariate polynomial systems over finite fields have been studied for several decades. This problem is believed to be hard to solve even for full–scale quantum computers, and so multivariate cryptography has received increasing attention the past years as post–quantum cryptography has become ever more important. A noteworthy initiative in this area is the ongoing post–quantum standardization process by the National Institute of Standards and Technology (NIST).

One of the earliest and most notable examples of multivariate cryptosystems is the encryption scheme C^* proposed by Matsumoto and Imai in 1988 [22]. Their idea was to let the public polynomial system defined over a small base field have a secret, but simple description over a larger extension field, where decryption can be done efficiently. While C^* was broken by Patarin in 1995 [23], several schemes were later proposed based on the same underlying idea; these are often referred to as *big field schemes*. One generalisation is to make the central map over the extension field more complex. Examples include HFE and its variants [24], as well as k–ary C^* [18]. Another idea is to keep the simple description over the extension field, but alter the resulting public key with modifiers that enhance the security against known attacks, as for example done in SFLASH [25] and PFLASH [7].

While there are presently several multivariate *signature* schemes that have resisted years of cryptanalysis, designing multivariate *encryption* schemes seems to be much more challenging. Examples of multivariate encryption schemes that have been successfully cryptanalysed include not only the original C^* [22][23], but also HFE [24][3], ABC [28][21], ZFHE [27][5] and SRP [29][26]. This observation is further echoed by the fact that all four multivariate cryptosystems that have made it to the second round of the NIST standardization process are signature schemes. EFLASH [6], proposed by Cartor and Smith-Tone at SAC 2018, is yet another attempt to design a secure and efficient multivariate encryption scheme. At its core, EFLASH is a modified C^* scheme with a new decryption strategy to maintain effectiveness.

1.1 Our Contribution

We present a direct algebraic cryptanalysis of EFLASH, based on the notion of *first fall degree*. We do so by developing a method to estimate this degree for the equation systems arising from EFLASH – an original approach which is different from the rank–based analysis that has been used against somewhat similar HFE variants. We are not only able to predict the first fall degree itself, but also the exact number of first fall polynomials occurring at step degrees 3 and 4. Our analysis indicates that EFLASH does not offer the level of security claimed by the designers; in particular, we are able to successfully cryptanalyse the EFLASH version with 80-bit security parameters. Ultimately, we hope that our approach can lead to a deeper understanding of the impact similar modifiers have on big field schemes.

1.2 Organisation

The paper is organised as follows. In Section 2 we go through the required preliminaries for our analysis. This includes a description of EFLASH, a brief discussion on the complexity of Gröbner basis algorithms, along with the notions of first fall and solving degrees, as well as some results on univariate and multivariate representation of polynomials. In Section 3 we present and discuss the previously suggested bound on the first fall degree of EFLASH. In Section 4 we develop the theory behind our new approach for estimating this degree for EFLASH, and put it to the test by experiments in Section 5. We discuss the implications that our analysis and experiments have on the security of EFLASH in Section 6. Potential follow-up work is discussed in Section 7, with our conclusions in Section 8.

2 Preliminaries

2.1 Description of EFLASH

EFLASH is a public-key encryption scheme proposed at SAC 2018 [6]. The system is built around the C^* encryption scheme by Matsumoto and Imai [22], using both the minus-modifier that removes some polynomials from the public key, and the embedding of the plaintext space $\mathbb{F}_q^n$ into a larger space $\mathbb{F}_q^d$. The signature scheme PFLASH [10, 7] is built in the same way, and EFLASH can be seen as the encryption variant of PFLASH.

The C^* scheme has operations taking place in $\mathbb{F}_q^d$ and $\mathbb{F}_{q^d}$. The encryption for C^* can be explained as follows: the plaintext and ciphertext spaces are both $\mathbb{F}_q^d$. Let S and T be two invertible $d \times d$-matrices over $\mathbb{F}_q$, defining linear transformations of $\mathbb{F}_q^d$. Fix an isomorphism between $\mathbb{F}_q^d$ and $\mathbb{F}_{q^d}$, denoted by ϕ, where $\phi : \mathbb{F}_q^d \longrightarrow \mathbb{F}_{q^d}$. Finally, we have the central mapping $X \mapsto X^{1+q^{\mathit{\Theta}}}$ over $\mathbb{F}_{q^d}$.

These mappings are combined together into P' as follows

$$P' = T \circ \phi^{-1} \circ X^{1+q^{\mathit{\Theta}}} \circ \phi \circ S. \tag{1}$$

Since the exponent of X has q-weight 2 and all other operations are linear, P' can be expressed as d quadratic polynomials in d variables over $\mathbb{F}_q$. The secret key of the C^* scheme are the two matrices S, T, and the public key consists of the polynomials P'. Encryption of a plaintext x into the ciphertext y is done by computing $y = P'(x)$. Decryption by someone knowing S and T can be done efficiently by inverting all operations in (1).

In [23] the basic C^* scheme was broken, by finding bilinear polynomials $f_i(x, y) = 0$ that relate the plaintext x with the ciphertext y. Computing the polynomials f_i's turns out to be easy, more so when knowing S and T. In fact, the most efficient decryption is actually done by inserting the values of y in the f_i's, and solving the resulting linear system of equations to recover the plaintext.

EFLASH expands the C^* scheme by adding an embedding π at the beginning and a projection τ in the end. More specifically, for $n < m < d$, the operations π and τ are defined as

$$\begin{aligned} \pi : \quad \mathbb{F}_q^n \quad &\longrightarrow \mathbb{F}_q^d \\ (x_1, \ldots, x_n) &\longmapsto (x_1, \ldots, x_n, 0, \ldots, 0) \end{aligned}$$

and

$$\begin{aligned} \tau : \quad \mathbb{F}_q^d \quad &\longrightarrow \mathbb{F}_q^m \\ (y_1, \ldots, y_d) &\longmapsto (y_1, \ldots, y_m) \end{aligned}$$

The plaintext space of EFLASH is then $\mathbb{F}_q^n$ and the ciphertext space is $\mathbb{F}_q^m$. The mappings π and τ are added as wrappers around the C^* scheme, so the complete EFLASH mapping P becomes

$$P = \tau \circ P' \circ \pi.$$

The complete diagram of mappings is shown in Figure 1.

$$\begin{tikzcd}
& & \mathbb{F}_{q^d} \arrow[r, "X^{1+q^{\Theta}}"] & \mathbb{F}_{q^d} \arrow[d, "\phi^{-1}"] & & \\
& \mathbb{F}_q^d \arrow[r, "S"] & \mathbb{F}_q^d \arrow[u, "\phi"'] & \mathbb{F}_q^d \arrow[r, "T"] & \mathbb{F}_q^d \arrow[dr, "\tau"] & \\
\mathbb{F}_q^n \arrow[ur, "\pi"] & & & & & \mathbb{F}_q^m
\end{tikzcd}$$

Fig. 1: Diagram of EFLASH mappings.

The extra mappings π and τ just add and remove some coordinates, so P can still be expressed as m quadratic polynomials over $\mathbb{F}_q$ in n variables. The size of the projection τ is an important parameter, so for convenience we define $a = d - m$ to be the number of polynomials removed from P'. The public key of EFLASH consists of the m polynomials in P, and the secret key is still the two matrices S, T (we assume the exponent Θ is publicly known).

Encryption in EFLASH is done the same way as for C^*: the plaintext x is transformed into ciphertext y by computing $y = P(x)$. On the other hand decryption is not as completely straightforward as for C^*. For a given ciphertext $y = (y_1, \ldots, y_m)$, the decryptor will exhaustively try all possible values for the missing coordinates $y_{m+1}, \ldots, y_d$, and decrypt every choice using the bilinear polynomials $f_i(x, y)$ from the C^* scheme. This results in up to q^a possible plaintexts embedded in $\mathbb{F}_q^d$, and the one whose last $d - n$ coordinates are all zero is chosen as the correct one. As $n < m$ we can expect there will be only one possible plaintext fulfilling the restriction given by π. In [6] the authors analyse the probability of there being two or more possible plaintexts matching a given ciphertext, which would lead to a decryption failure. For the suggested choices of n, m, d the probability is approximately 2^{-17}, which is still non-negligible.

Table 1 shows the parameters suggested in [6] for 80- and 128-bit security levels against an attacker with either a classical or quantum computer available.

In the remainder of the paper we will fix $q = 2$. Although most of the theory presented in later sections can be generalised to other fields, this is what is often used in practice and in particular what is suggested in EFLASH (Table 1).

Table 1: Suggested parameters (q, n, m, d) for EFLASH.

	80-bit security	128-bit security
classical adversary	$(2, 80, 96, 101)$	$(2, 134, 150, 159)$
quantum adversary	$(2, 160, 176, 181)$	$(2, 256, 272, 279)$

2.2 Gröbner Basis Algorithms

As is the case for all multivariate encryption schemes, the plaintext $(a_1, ..., a_n)$ associated to the ciphertext $(y_1, ..., y_m)$ can be found through direct attacks, that is, by solving the polynomial system

$$p_1(x_1, ..., x_n) + y_1 = ... = p_m(x_1, ..., x_n) + y_m = 0,$$

where $p_i(x_1, ..., x_n)$, $1 \leq i \leq m$, are the quadratic polynomials that make up the public key P. The usual strategy for solving such a system is to compute a Gröbner basis (see [8] for further details) for the ideal $\langle p_i + y_i \rangle_{1 \leq i \leq m}$ in the grevlex monomial order, using a state–of–the–art algorithm such as F_4 [14] or F_5 [15]. Since we implicitly include the field equations, the system generates a radical ideal. The solution of this system can by design be assumed to be unique and thus we are able to solve it directly from the Gröbner basis, which is by the above remark $x_1 + a_1, \ldots, x_n + a_n$ for any term ordering.

In our setting the F_4 algorithm will proceed step–wise, and to each step there is an associated *step degree* D, which is the maximal degree of the polynomials involved in this step. The complexity of each step is dominated by reduction of a Macaulay matrix associated with these polynomials. If we define the *solving degree*, D_{solv}, to be the step degree associated with the largest such matrix (this notation was introduced in [13]), then the complexity of the algorithm (in the Boolean case) can be estimated by:

$$\text{Complexity}_{\text{GB}} = \mathcal{O}\Bigg(\bigg(\sum_{i=0}^{D_{solv}} \binom{n}{i} \bigg)^{\omega} \Bigg), \tag{2}$$

where n is the number of variables and $2 \leq \omega \leq 3$ is the linear algebra constant. This makes D_{solv} crucial for estimating the complexity of a direct attack, but in general this value is difficult to determine. It is also worth noting that D_{solv} is not necessarily the highest degree encountered in the algorithm; indeed [13] shows examples of this for HFE–systems, while we will also see examples where this is the case for EFLASH in Section 5.

An important class of polynomial systems where D_{solv} can be determined is the class of *semi–regular sequences* [2]. In this case D_{solv} will coincide with the degree of regularity D_{reg}, which for quadratic polynomial systems over $\mathbb{F}_2$ can be calculated as the degree of the first non–positive term in the series [1]:

$$T_{m,n}(z) = \frac{(1+z)^n}{(1+z^2)^m}. \tag{3}$$

From experiments it seems to be the case that randomly generated polynomial systems will behave as semi–regular sequences [2], and the degree of regularity is in many instances sensible to use for complexity estimation. However, it is well known that polynomial systems associated with big field multivariate cryptography tend to have a lower solving degree than what is predicted by the degree of regularity; see for example [16]. For these schemes the notion of *first fall degree* (Definition 1), which in general provides a lower bound for the solving degree, has been often used to estimate the complexity of solving such systems [11, 12]. The authors of EFLASH have also chosen this path, and in [6] a bound for the first fall degree was derived and used to estimate the resistance of this scheme against algebraic attacks. We will later argue that this derived bound for the first fall degree is not tight, but the idea of using this invariant as an approximation for the solving degree seems justified for EFLASH. Indeed, in all our experiments we find the solving degree to be either the same or one greater than the first fall degree (see Section 5). We end this subsection by recalling the definition of first fall degree.

Consider the graded quotient ring $B = \mathbb{F}_2[x_1, ..., x_n]/\langle x_1^2, ..., x_n^2\rangle$, where $B_\nu \subset B$ is the set of homogeneous polynomials of degree ν in B. Let $p_1^h, ..., p_m^h \in B_2$ be the homogeneous quadratic part of the polynomials in the public-key P, and $p_i^l, 1 \leq i \leq m$ be the corresponding linear, or lower-degree, terms, so that $p_i = p_i^h + p_i^l$. We can then define the map

$$\begin{array}{rcl} \psi_{\nu-2} : & B_{\nu-2}^m & \longrightarrow B_\nu \\ & (f_1, ..., f_m) & \longmapsto \sum_{i=1}^m f_i p_i^h \end{array}$$

Any element of $ker(\psi_{\nu-2})$ is called a *syzygy*. Now let $\nu = 4$. Then particular syzygies are the *Kozul syzygies*, generated by $(0, ..., 0, p_j^h, 0, ..., 0, p_i^h, 0, ..., 0)$ where p_j^h is in position i and p_i^h is in position j, and the *field syzygies* generated by $(0, ..., 0, p_i^h, 0, ..., 0)$ (p_i^h in position i). These syzygies will boil down to the relations $p_j^h p_i^h + p_i^h p_j^h = 0$ and $(p_i^h)^2 = 0$. Since they are always present, and not depending on the polynomials p_i^h themselves, these syzygies generate the *trivial syzygies*, $T(\psi_{\nu-2}) \subseteq ker(\psi_{\nu-2})$.

Definition 1. *The* first fall degree *associated with the quadratic polynomial system $p_1, ..., p_m$ is the natural number*

$$D_{ff} = min\{\ d \geq 2 \mid ker(\psi_{d-2})/T(\psi_{d-2}) \neq 0\ \}.$$

Remark 1. The elements $(0, ..., 0, p_j^h, 0, ..., 0, p_i^h, 0, ..., 0)$ and $(0, ..., 0, p_i^h, 0, ..., 0)$ will, strictly speaking, not be syzygies themselves when solving for $p_1, ..., p_m$ in $\mathbb{F}_2[x_1, ..., x_n]$. For example, $p_j^h p_i + p_i^h p_j \neq 0$ will in general be of degree 3. We still call these degree falls *trivial*, as they do not give any new or useful information in an actual attack. This fact can be seen as follows.

When trying to solve a system by multiplying equations with all monomials up to some degree, the multiplications are done by increasing degrees. That is, all monomials of degree $\leq D - 1$ are used before multiplying with monomials of degree D. The Kozul syzygies will give the degree fall polynomial

$$p_j^h p_i + p_i^h p_j = p_j^h(p_i^h + p_i^l) + p_i^h(p_j^h + p_j^l) = p_j^h p_i^l + p_i^h p_j^l.$$

However, the very same polynomial can be expressed using only multiplication with the lower-degree monomials in p_j^l and p_i^l:

$$p_i^l p_j + p_j^l p_i = p_i^l(p_j^h + p_j^l) + p_j^l(p_i^h + p_i^l) = p_i^l p_j^h + p_j^l p_i^h.$$

Hence the degree fall generated by p_i^h and p_j^h does not give us anything new when we already have multiplied with all lower-degree terms. Moreover it is a priori clear that these polynomials reduce to zero modulo p_j, p_i and therefore give no new information when computing a Gröbner basis, except slowing the computation down.

The same holds for the field syzygies, where it is easy to see that the polynomial $p_i p_i = p_i$ can be "generated" by the (lower-degree) constant 1 as $1 \cdot p_i$.

2.3 Univariate and Multivariate Representation of Polynomials

Our analysis will rely heavily on the easy description the central map of EFLASH has as univariate polynomial over the extension field. The idea of exploiting this simple description in cryptanalysis was also used in the Kipnis–Shamir attack on HFE in [20], and we refer to their work for further details on the following result. We will write $w(t)$ to denote the *binary weight* of an integer t. Recall that this is defined as $\sum z_i$, where $t = \sum z_i 2^i$ is the 2–adic representation of t.

Theorem 1. *Let $P(X) \in \mathbb{F}_{2^d}[X]/\langle X^{2^d} + X\rangle$ and fix an isomorphism ϕ between $\mathbb{F}_{2^d}$ and $(\mathbb{F}_2)^d$. With this isomorphism, $P(X)$ admits d unique polynomials $p_1, ..., p_d \in \mathbb{F}_2[x_1, ..., x_d]/\langle x_1^2 + x_1, ..., x_d^2 + x_d\rangle$. Furthermore, the degree of the polynomials $p_1, ..., p_d$ is given by max$\{w(t) \mid X^t \in \mathcal{M}_P\}$, where $\mathcal{M}_P$ is the set of monomials in $P(X)$ with non-zero coefficients.*

Based on this result we will define the *2–weight* associated with a polynomial $P(X) \in \mathbb{F}_{2^d}[X]/\langle X^{2^d} + X\rangle$ to be $w(P)$ =max$\{w(t) \mid X^t \in \mathcal{M}_P\}$. There are two particular actions over the extension field, and their corresponding actions over the base field, that are worth pointing out. First, we note that raising $P(X)$ to a power of 2, i.e. $(P(X))^{2^i}$, will correspond to applying an invertible linear transformation on the associated multivariate polynomials $p_1, ..., p_d$.

The second action is that the multivariate polynomials associated with the product $H(X)P(X)$ will be d sums of the form $\sum h_j p_i$, where h_i is a multivariate polynomial of maximum degree equal to $w(H)$. These actions (on the multivariate polynomials) are exactly the ones performed by Gröbner basis algorithms. Linear maps do not affect the degree of the polynomials, so if $T \circ \phi^{-1} \circ P(X) \circ \phi \circ S$ is the central map of an unmodified big field scheme (e.g. original C^* or HFE), then the degree fall polynomials encountered when computing a Gröbner basis can be described by the two aforementioned actions on the univariate polynomial $P(X)$. More specifically, we will call any combination

$$F(X) = \sum_{i,j} [C_{i,j} H_i(X) P(X)]^{2^j} \in \mathbb{F}_{2^d}[X]/\langle X^{2^d} + X \rangle,$$

where

$$w(F) < w(P) + \max\{w(H_i)\},$$

a *2–weight fall polynomial.* This will in turn admit d multivariate degree fall polynomials.

We note that in the Faugère–Joux attack on HFE [16] these 2–weight fall polynomials are the reason for the effectiveness of algebraic attacks on this cryptosystem. Likewise, in [18] specific q–weight fall polynomials (i.e. the natural generalisation to other fields of size q) were constructed in order to show the first fall degree of k–ary C^*, another generalisation of C^*. Things get more complicated as modifiers are added to the public key, particularly in the case for the minus modifier. However we will describe how to deal with this in Section 4.

3 Suggested First Fall Degree Bound

In this section we discuss an upper bound for the first fall degree that was suggested for EFLASH in [6][4]. Since EFLASH can be seen as a special case of HFE-, the bound is derived following a similar line of reasoning as was used for this latter scheme in [12]. The idea is to first examine how the minus modifier affects the Q–rank of the quadratic form associated with the central map, and then apply this to the upper bound derived in Theorem 4.1 of [11]. The arguments made in Section 5.1 of [6] is that the minus modifier is even more effective at increasing the Q–rank when applied to EFLASH than it is for HFE-, due to the extreme sparseness of the central map of the former. This led to the following upper bound for EFLASH [6]:

$$D_{ff,EFLASH} \leq a + 3. \tag{4}$$

However we argue that focusing on Q–rank alone does not reveal the entire picture when the (unmodified) central map is as simple as it is in EFLASH. To this end we introduce the following notation, which will also be important for our own estimates of first fall degree:

Definition 2. *Consider the quotient ring $\mathbb{F}_{2^d}[X]/\langle X^{2^d} + X \rangle$, and an instance of C^*. Let $y \in \mathbb{F}_2^d$ represent a given ciphertext, and $V = \phi \circ T^{-1}(y)$. We then define*

$$Q = X^{1+2^{\Theta}} + V \tag{5}$$

[4] The authors call this the degree of regularity, but are in fact describing the first fall degree.

to represent the central map associated to C^* over $\mathbb{F}_{2^d}[X]/\langle X^{2^d} + X\rangle$. We also define the following 2–weight fall equations:

$$\alpha = X^{2^{d-\Theta}} Q + X^{2^\Theta} Q^{2^{d-\Theta}} = X^{2^{d-\Theta}} V + X^{2^\Theta} V^{2^{d-\Theta}}, \tag{6}$$

$$\beta_1 = XQ = X^{2+2^\Theta} + XV \text{ and} \tag{7}$$

$$\beta_2 = X^{2^\Theta} Q = X^{1+2^{\Theta+1}} + X^{2^\Theta} V. \tag{8}$$

Since we are not removing any polynomials (i.e. a = 0), Equation (4) predicts that the polynomial Q defined above has first fall degree 3 (this is also pointed out in Example 4.3 in [11]). Here Q is treated as any polynomial with Q–rank 2, and following the proof of Theorem 4.1 in [11], we find that the predicted first fall degree is due to the existence of the univariate polynomials β_1 and β_2, which would correspond to quadratic multivariate polynomials. However, in the definition above there is also a third 2–weight fall polynomial, α, which will correspond to linear multivariate polynomials (these are the same that Patarin found in his original attack on C^* [23]). Thus there seems to be more information in the system than what is captured by methods based on the Q–rank alone. It is indeed the case that removing public polynomials makes it more difficult for an attacker, but we will see in the next section that there may still be combinations of multivariate degree fall polynomials, generated by the relations α, β_1 and β_2 present in the polynomial system. Again, methods based on the Q–rank alone do not seem to fully capture this.

Another notable difference between EFLASH and HFE- is the large dimension of the embedding ($n < d$) present in the former. We will see that this modifier also plays a role in determining the number of degree fall polynomials in a system. While it does not have the same impact as the minus modifier, there are parameters for which this affects the first fall degree of a system; see Section 5 for examples.

4 The First Fall Degree of EFLASH

This section starts off with a brief discussion on the impact the choice of Θ may have on the security of EFLASH. The condition that $\gcd(2^d - 1, 2^\Theta + 1) = 1$ is needed for the map X^{1+2^Θ} to be a bijection, and has been a requirement for this family of cryptosystems ever since the original paper of Matsumoto and Imai [22]. While not explicitly stated in [6], it seems reasonable to assume that this is also the case for EFLASH. We will later see that the total number of degree fall polynomials in the original C^*–scheme will have a big impact on the complexity of algebraic attacks towards EFLASH.

The question of how different choices of Θ affect the number of degree fall polynomials has partly been studied in [9]. In that work the authors consider the effect Θ has on the number of linearisation equations, which can be seen as a special subset of degree fall polynomials of degree 1. Examples of special values for Θ from this work are $\Theta = d/3$ and $\Theta = 2d/3$. In these cases it is shown that there are only $2d/3$ linearisation equations, and so it is unlikely that these choices for Θ can be used in an efficient instantiation of EFLASH (as d linear equations are used for decryption). On the other hand, there are also cases found in [9] that renders more than d linear equations, which could benefit an attacker. What would amount to special cases in our analysis will ultimately go beyond linear equations: for $D = 3$, degree falls polynomials will also include quadratic polynomials, and cubic polynomials when $D = 4$. It is beyond the scope of this paper to identify every such special case. Therefore for the rest of this paper, unless otherwise stated, all equations and formulas are assumed to hold for *general* choices of Θ. *General* is here used in a non–technical sense by which we mean that we expect the result in question to hold for all values $\Theta = 0, 1, \dots, d-1$, save for a few exceptions.

4.1 The Effect of Removing Polynomials

We wish to obtain a representation of the central map of EFLASH that in some sense not only preserves the easy description given over the univariate polynomial ring, but also keeps track of what is lost due to the minus modifier, τ. Consider the cryptosystem in a state before τ has been applied (but after the linear transformation T, see Figure 1). Finding a plaintext associated with a fixed ciphertext would amount to solving the system of quadratic polynomials $p_i(x_1, ..., x_n) = 0$, for $1 \leq i \leq d$ (for ease of notation we are assuming the fixed ciphertext to be part of the p_i–polynomials). Let

$$\begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_d \end{bmatrix} = T^{-1} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_d \end{bmatrix}, \tag{9}$$

in other words, each q_i is a linear combination of the polynomials $p_1, ..., p_d$.

Even though the polynomials p_j are depending on the x–variables, we will at an intermediate step want to consider them as formal variables. In an effort to keep the notation precise, we will write $\hat{p}_1, ..., \hat{p}_a$ to denote the polynomials as formal variables that will be removed by τ. On the other hand, $\bar{p}_{a+1}, ..., \bar{p}_d$ will denote the formal variables associated with the polynomials unaffected by τ (i.e. the public polynomials). We will also write q_i^* to denote the linear combinations defined in Equation (9), but now depending on the formal variables $\hat{p}_j$ and $\bar{p}_k$.

In the previous section we have considered sums of the form $\sum X^{2^{i_1}+\ldots+2^{i_k}} Q^{2^j}$ in the univariate polynomial ring $\mathbb{F}_{2^d}[X]/\langle X^{2^d} + X\rangle$. We will now inspect the same sums, but treat Q as a formal variable in the bivariate polynomial ring $\mathcal{A}_{XQ} := \mathbb{F}_{2^d}[X, Q]/\langle X^{2^d} + X, Q^{2^d} + Q\rangle$. We will furthermore write Q as $Q = (q_1^* + q_2^*\gamma + ... + q_d^*\gamma^{d-1})$, where γ is a primitive element associated with the isomorphism ϕ. We then consider the following composition of maps:

$$\mathcal{A}_{XQ} \xrightarrow{\phi^{-1}} (\mathbb{F}_2[x_1, ..., x_n, \hat{p}_1, ..., \hat{p}_a, \bar{p}_{a+1}, ..., \bar{p}_d])^d \xrightarrow{ev_{P,a}} (\mathbb{F}_2[x_1, ..., x_n])^d$$

where $ev_{P,a}$ acts entry–wise in the d–vector space by "evaluating" the formal variables $\hat{p}$ to 0, and regarding $\bar{p}$ as polynomials in x–variables. To be more precise, $ev_{P,a} : (z_1, ..., z_d) \mapsto (ev^*_{P,a}(z_1), ..., ev^*_{P,a}(z_d))$, where:

$$\begin{aligned} ev^*_{P,a} : \mathbb{F}_2[x_1, ..., x_n, \hat{p}_1, ..., \bar{p}_d] &\longrightarrow \mathbb{F}_2[x_1, ..., x_n] \\ x_i &\longmapsto x_i \text{ for } 1 \le i \le n \\ \hat{p}_j &\longmapsto 0 \text{ for } 1 \le j \le a \\ \bar{p}_k &\longmapsto p_k(x_1, ..., x_n) \text{ for } a+1 \le k \le d. \end{aligned}$$

It is straightforward to check that if t is an integer with 2–weight $D-2$, then $ev_{P,a} \circ \phi^{-1}(X^t Q)$ will result in d polynomials of degree at most D, which are generated by the public polynomials $p_{a+1}, ..., p_d$. We will use this new notation to show the following lemma, which will be key in our ensuing analysis. An interpretation is that the minus modifier τ only obscures the degree fall polynomials by adding polynomials generated from a small set, namely the removed polynomials $p_1, ..., p_a$.

Lemma 1. *Let $ev_{P,0} \circ \phi^{-1}(\sum X^{k_1} Q^{k_2})$ give d polynomials over $\mathbb{F}_2[x_1, ..., x_n]$ that are degree fall polynomials of degree $< D = w(k_1) + 2w(k_2)$. Then, for $a > 0$ the degree D–parts of the d polynomials $ev_{P,a} \circ \phi^{-1}(\sum X^{k_1} Q^{k_2})$ are generated by $p_1, ..., p_a$.*

Proof. Let g be any of the d polynomials in $\mathbb{F}_2[x_1, ..., x_n, \hat{p}_1, ..., \bar{p}_d]$, that are in the image of $\phi^{-1}(\sum X^{k_1} Q^{k_2})$. Fix polynomials $h_1, h_2, ..., h_{a+1}$ such that we can write g on the triangular form:

$$\begin{aligned} g = &\, h_1(x_1, ..., x_n, \hat{p}_2, ..., \hat{p}_a, \bar{p}_{a+1}, ..., \bar{p}_d)\hat{p}_1 \\ &+ h_2(x_1, ..., x_n, \hat{p}_3, ..., \hat{p}_a, \bar{p}_{a+1}, ..., \bar{p}_d)\hat{p}_2 \\ &\vdots \\ &+ h_a(x_1, ..., x_n, \bar{p}_{a+1}, ..., \bar{p}_d)\hat{p}_a \\ &+ h_{a+1}(x_1, ..., x_n, \bar{p}_{a+1}, ..., \bar{p}_d) \end{aligned}$$

Recall that when $a > 0$ then $ev^*_{P,a}(\hat{p}_j) = 0$ for $1 \le j \le a$. Since we are working over a field of characteristic 2, we can equivalently think of this as addition with all terms containing the $\hat{p}_j$–variables and then evaluating everything using $ev^*_{P,0}$. Note that all $\hat{p}_i$ change to $\bar{p}_i$ when evaluated with $ev^*_{P,0}$ instead of $ev^*_{P,a}$. This can then be written out as follows:

$$\begin{aligned} ev^*_{P,a}(g) &= ev^*_{P,0}(g + \sum_{1 \le i \le a} h_i \bar{p}_i) \\ &= ev^*_{P,0}(g) + ev^*_{P,0}(\sum_{1 \le i \le a} h_i \bar{p}_i) \\ &= ev^*_{P,0}(g) + \sum_{1 \le i \le a} h_i p_i. \end{aligned}$$

By assumption $ev^*_{P,0}(g)$ has degree $< D$ so any term of degree D must come from $\sum_{1 \leq i \leq a} h_i p_i$, which proves the statement. □

One observation that can be drawn from this lemma is that if the number of degree fall polynomials that would be generated by a similar polynomial system with $a = 0$ exceed the number of highest degree combinations generated by the removed polynomials (i.e. the possible combinations of $x_{i_1} ... x_{i_{D-2}} \hat{p}_j$), then there will be linear combinations of the degree fall polynomials that can be written without the use of $\hat{p}_j$–elements. These can in turn be found by an attacker through the use of Gröbner basis algorithms. This is the intuition that will be further explored in the following subsections, but first we illustrate the point for the bilinear equations in the following example:

Example 1. *Consider an EFLASH instance with $a = 1$. Recall from Equation (6) in Definition 2 that the bilinear relations come from $\alpha = X^{2^{d-\Theta}} Q + X^{2^{\Theta}} Q^{2^{d-\Theta}}$. By Lemma 1 we can write $ev_{P,1} \circ \phi^{-1}(\alpha)$ as d polynomials in the ring $\mathbb{F}_2[x_1, ..., x_n]$, whose degree 3–part are linear combinations of $x_i \hat{p}_1$ for $1 \leq i \leq n$. This means that the homogeneous degree 3–part has at most dimension n, whereas the image of $ev_{P,1} \circ \phi^{-1}(\alpha)$ has dimension d (under the assumption that the resulting d polynomials are linearly independent). Since $d > n$ for EFLASH, this means that there will be $d - n$ different independent linear combinations of these polynomials that can be written without using $\hat{p}_1$. As a result a Gröbner basis algorithm will find $d - n$ linear relations at $D = 3$.*

It is worth pointing out that the embedding modifier π, while needed to protect against differential attacks and more sophisticated attacks, as e.g. in [4], actually weakens the effect of the minus modifier τ. Indeed, had there been no embedding, i.e. $d = n$, we would not expect to find any linear relations at $D = 3$ in the example above. Thus in this special case we see there is a trade-off between π and τ. Without the embedding one would have to deal with the above mentioned attacks while the classic attack by Patarin would be prevented. On the other hand, by applying the embedding you would get back parts of the linear relations from Patarin's classical attacks while preventing the above attacks. This shows that more research is required to better understand how to securely combine the two kinds of modifiers.

In the next two subsections we will focus on how things evolve when increasing the step degree D. We start by generalising Example 1 to include more degree falls at $D = 3$.

4.2 First Fall Polynomials at D = 3

In Definition 2 we saw that with $a = 0$, we will in addition to the linear polynomials given by α (Equation (6)) also have two more quadratic degree falls given by β_1 and β_2 (Equations (7) and (8)). The $3d$ multivariate polynomials associated to these will in general account for all the degree fall polynomials that show up at step degree $D = 3$. Lemma 1 implies that when $a > 0$ these polynomials will generally be of degree 3, where the degree 3–part is further generated by the polynomials $x_i p_j$, for $1 \leq i \leq n$ and $1 \leq j \leq a$. Hence there are $3d$ resulting polynomials where the top degree is generated by na elements, and so an estimate of the number of degree fall polynomials at $D = 3$ can be found by merely subtracting the two. To be more precise, recall from Section 2.2 that $ker(\psi_{D-2})/T(\psi_{D-2})$ denotes the vector space of non–trivial degree fall polynomials at degree D. We write $\{\#\mathrm{P}_{\mathrm{df}}\}_D = dim\big(ker(\psi_{D-2})/T(\psi_{D-2})\big)$ for its dimension, and derive the following estimate for $\{\#\mathrm{P}_{\mathrm{df}}\}_3$:

$$N_3(n, d, a) = 3d - na. \tag{10}$$

When N_3 is negative, we do not expect to find any degree fall polynomials. In this case we take $\max\{N_3, 0\}$ as the estimate for $\{\#\mathrm{P}_{\mathrm{df}}\}_3$. The accuracy of this estimate will be tested in Section 5

4.3 First Fall Polynomials at $D = 4$

The analysis gets more complicated at step degree 4, mainly due to the syzygies appearing in the polynomial system at this degree. More specifically we wish to find out what polynomials in $\mathcal{A}_{XQ}$ that will correspond to multivariate degree falls that are considered trivial, in the sense of Remark 1, by Gröbner basis algorithms. The following lemma classifies these polynomials.

Lemma 2. *The polynomials associated with*

$$ev_{P,a} \circ \phi^{-1}\big[(X^{1+2^{\Theta}})^{2^{k_1}} Q^{2^{k_2}}\big], \ \textit{for } 0 \leq k_1, k_2 \leq d-1.$$

can be written on the form:

$$\sum_{\substack{1 \leq i \leq d \\ a+1 \leq j_1 \leq d \\ i \neq j_1}} b_{i,j_1} p_i p_{j_1} + \sum_{a+1 \leq j_2 \leq d} c_{j_2} p_{j_2}, \ \textit{for } b_{i,j_1}, c_{j_2} \in \mathbb{F}_2. \tag{11}$$

Proof. We prove the statement for the case $k_2 = 0$ (other values of k_2 can be written as a power of 2 of this case). For the ciphertext $(y_1, ..., y_d)$, write:

$$\begin{bmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_d \end{bmatrix} = T^{-1} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_d \end{bmatrix}.$$

Recall that we included the ciphertext in the definition of the p_i–polynomials, so this must be accounted for when considering $X^{1+2^{\Theta}}$ (which will contain no constant terms). We then have:

$$(X^{1+2^{\Theta}})^{2^{k_1}} Q = \left[\sum_{i=1}^{d} (q_i + y'_i)\gamma^{(i-1)2^{k_1}}\right] \cdot \left[\sum_{j=1}^{d} q^*_j \gamma^{j-1}\right],$$

and so if g is any of the d polynomials in $\phi^{-1}\big((X^{1+2^\Theta})^{2^{k_1}}Q\big)$, we can write:

$$g = q_1^*\left[\sum_{i=1}^{d} g_{1i}(q_i + y_i')\right] + ... + q_d^*\left[\sum_{i=1}^{d} g_{di}(q_i + y_i')\right]$$

for some $g_{ji} \in \mathbb{F}_2$. Recall that the q_i's are linear combinations of $p_1, ...p_d$ (written out in $\mathbb{F}_2[x_1, ..., x_n]$) and will be unaffected by $ev_{p,a}^*$. The q_i^*'s are linear combinations of the formal variables $\hat{p}_1, ..., \bar{p}_d$. Since the evaluation map sends all the variables $\hat{p}_1, ...\hat{p}_a$ to zero, the statement (11) in the lemma now follows from $ev_{p,a}^*(g)$. □

We note that a system of quadratic polynomials $p_1, ..., p_d$ with the property that a sum of the form $\sum_{i \neq j} b_{i,j} p_i p_j$, with $b_{i,j} \in \mathbb{F}_2$, results in a non–trivial degree fall (i.e. one not generated by Kozul Syzygies) would be a very degenerate system, not suitable for multivariate cryptography. We may assume therefore that a polynomial system associated with C^* is very unlikely to have this property. Thus, under the assumption that no such non–trivial relation exists, Lemma 2 implies that any degree fall polynomial that originates from a sum of the form $\sum_{k_1,k_2} c_{k_1,k_2}(X^{1+2^\Theta})^{2^{k_1}} Q^{2^{k_2}}$ is simply a linear combination of the public polynomials $p_{a+1}, ..., p_d$. As this gives no new information to an attacker, it should be regarded as trivial (similar to what was discussed in Remark 1).

We may now return to the question of what degree fall combinations that should be counted. The polynomials α, β_1 and β_2 discussed earlier, when multiplied with X^{2^i} will also generate degree fall polynomials for $D = 4$. Indeed, our experiments suggest that all of degree fall polynomials at this step degree are generated by these elements.

At first glance there will be $3dn$ multivariate polynomials associated with the elements $X^{2^i}\alpha$, $X^{2^i}\beta_1$ and $X^{2^i}\beta_2$ for $1 \leq i \leq d$. Note that here we are using the fact that the variable X may be written using linear combinations of the n variables $x_1, ..., x_n$. Hence, multiplying by all $X, X^2, ..., X^{2^{d-1}}$ will effectively only give n different combinations, as opposed to d. However, not all of these should be counted, for various reasons. We list the exceptions below:

- $X\beta_1 = X^2Q$ and $X^{2^\Theta}\beta_2 = X^{2^{\Theta+1}}Q$ are both elements belonging to $D = 3$.
- $X^{2^\Theta}\beta_1 = X^{1+2^\Theta}Q = X\beta_2$, will be cases of the trivial degree falls discussed in Lemma 2. The same is true for $X^{2^{d-\Theta}}\beta_1 = (X^{1+2^\Theta})^{2^{d-\Theta}}Q$ and $X^{2^{2\Theta}}\beta_2 = (X^{1+2^\Theta})^{2^\Theta}Q$. Lastly, the following is a sum of two trivial degree falls: $X\alpha = (X^{1+2^\Theta})^{2^{d-\Theta}}Q + X^{1+2^\Theta}Q^{2^{d-\Theta}}$.
- From $X^{2^{d-\Theta}}\alpha = X^{2^{d-\Theta+1}}Q + X^{2^{d-\Theta}+2^\Theta}Q^{2^{d-\Theta}} = X^{2^{d-\Theta+1}}Q + \big(X^{2^{2\Theta}}\beta_1\big)^{2^{d-\Theta}}$ we see that $X^{2^{d-\Theta}}\alpha$ can be written out as a polynomial generated by β_1, and one regular polynomial of degree 3. For this reason, the degree fall polynomials generated by either $X^{2^{d-\Theta}}\alpha$ or $X^{2^{2\Theta}}\beta_1$ do not bring anything new to the system once the other has been created, and so only one should be counted. The same is true for $X^{2^\Theta}\alpha = X^{2^{d-\Theta}}\beta_2 + X^{2^{\Theta+1}}Q^{2^{d-\Theta}}$.

There are two, five and two relations from the first to last bullet point, respectively, which do not count towards generating new degree fall polynomials made from $X^{2^i}\alpha$, $X^{2^i}\beta_1$ and $X^{2^i}\beta_2$. Summing these up we find that the adjusted number of degree fall polynomials at $a = 0$ should be $(3n-9)d$.

At first it may seem that the degree 4 part will be generated by $a\binom{n}{2}$ elements, namely all combinations $x_i x_j \hat{p}_k$, but this does not take into account the trivial syzygies arising from the fact that the $\hat{p}_k$'s are ultimately polynomials in the x_i–variables. Thus one should retract all combinations of trivial syzygies involving the $\hat{p}_k$–elements, namely the field syzygies; $\hat{p}_k^2 + \hat{p}_k = 0$ and Kozul syzygies of the types $\hat{p}_i\hat{p}_k + \hat{p}_k\hat{p}_i = 0$, for $i, k \in \{1, \ldots, a\}$, and $\hat{p}_k\bar{p}_j + \bar{p}_j\hat{p}_k = 0$, for $k \in \{1, \ldots, a\}$ and $j \in \{a+1, \ldots, d\}$. There are a such field equations, $\binom{a}{2}$ of the Kozul syzygies of the first type and $a(d-a)$ Kozul syzygies of the second type. This sums up to

$$a + \binom{a}{2} + a(d-a) = ad + \frac{a - a^2}{2},$$

which should be subtracted from $a\binom{n}{2}$ to give the precise number of degree fall polynomials lost due to τ. Similar to the case $D = 3$, we can now add together everything discussed so far to obtain an estimate of the number of linearly independent degree fall polynomials at $D = 4$:

$$N_4(n, d, a) = (3n-9)d - a\binom{n}{2} + ad + \frac{a - a^2}{2}. \tag{12}$$

Again, N_4 may become negative, so we take $\max\{N_4, 0\}$ to be our estimate for $\{\#\mathrm{P}_{\mathrm{df}}\}_4$.

5 Experimental Results

We now present experimental results to test the validity of the formulas from the previous section predicting the number of first fall polynomials. In the first set of experiments (Table 2) we vary the choices of parameters d, n, a and Θ. The numbers N_3 and N_4 have been calculated according to equations (10) and (12), and the predicted first fall degree is the first degree where we expect a positive value. We then give the first fall degree and the number of first fall polynomials obtained at this step from the Gröbner basis routine in the MAGMA computer algebra system. In all our experiments the degree of the first fall polynomials were maximal, i.e. one less than the first fall degree. The solving degree is measured as the degree associated with the step having the largest matrix in the algorithm. In Section 5.1 of [6] the authors note that smaller EFLASH–systems could be solved at degree equal to or one lower than for random systems of the same parameters (D_{reg} in our notation). As the systems (and hence also D_{reg}) grow in size, it was suggested to use the bound in Equation (4), namely $a + 3$. We have included both D_{reg} and this bound in the last two columns of the table for comparison. One can notice that these values do not seem to be an adequate measure of the solving degree in our experiments.

Table 2: Experimental Results for EFLASH with varying parameters.

d	n	a	θ	N_3/N_4	D_{ff}	D_{ff}	$\{\#\mathrm{P}_{\mathrm{df}}\}_{D_{ff}}$	D_{solv}	$a+3$	D_{reg}
					(predicted)	(Magma)	(Magma)			
51	49	5	13	-92/1403	4	4	1403	4	8	9
51	49	3	13	6/3660	3	3	6	4	6	9
53	39	7	13	-114/887	4	4	887	5	10	7
56	40	9	8	-192/-336	≥ 5	4	20	5	12	7
56	40	4	8	8/3314	3	3	8	4	7	7
60	50	4	8	-20/3794	4	4	3794	4	7	8
63	50	3	7	39/5394	3	3	39	4^*	6	8
63	50	3	5	39/5394	3	3	39	4^*	6	8

$*$ The highest degree reached in MAGMA was 5, but this step occurred after 50 linear relations were found, and consequently had little impact on the running time.

Note that the first two entries satisfy the condition $n > d - a = m$. This is to emphasise that the validity of our theory is not only restricted to EFLASH (e.g. the parameters in the PFLASH signature scheme are taken to be $n > d - a$). There are several observations from Table 2 that we would like to point out. The first is that when at least one of the predictions N_3 and N_4 is positive, then our theory accurately predicts both the first fall degree and the number of polynomials obtained. An odd case in this regard happens in the fourth row, where we do not expect any degree fall polynomials at $D = 4$, but the GB algorithm is still able to find a small number of them. Secondly, we note that the recorded first fall degree and solving degrees are either the same or one apart in all the experiments. It is possible that this relation may be understood through the number of first fall polynomials. For example, a low $\{\#\mathrm{P}_{\mathrm{df}}\}_{D_{ff}}$ could imply $D_{solv} = D_{ff} + 1$, whereas a large $\{\#\mathrm{P}_{\mathrm{df}}\}_{D_{ff}}$ implies $D_{solv} = D_{ff}$, but any further exploration into this is beyond the scope of this paper.

The third point we wish to elaborate on from Table 2 is that the last two experiments differs only in $\Theta = 7$ and 5. Here 7 is a divisor of $d = 63$, while 5 is not. We obtain the same number of degree fall polynomials, indicating that for direct methods it does not seem to make a difference whether Θ divides d, as opposed to other attacks (see e.g. [17]).

In the next set of experiments we have fixed the value of the parameters $d = 56$, $n = 40$ and $\Theta = 8$, while only varying the number a of removed public polynomials. Note that when $a = 9$ this is the same case as presented in row 2 of Table 2. In these experiments we only present N_4 from equation (12) and the first fall degree and number of first fall polynomials measured by MAGMA.

For $6 \leq a \leq 8$ in Table 3 we find a positive value for N_4 and in these cases the theory exactly matches the experimental results. For $9 \leq a \leq 11$ the theory predicts no degree fall polynomials at $D = 4$, but MAGMA is still able to find a small number of degree fall polynomials here. We see that this number decreases by 9 as a is increased. When $a = 12$ public polynomials have been removed, no degree fall polynomials are detected at $D = 4$, but a substantial amount is found at $D = 5$.

Table 3: Effects of increasing a for $d = 56$, $n = 40$, $\Theta = 8$. The entry marked with * has been measured at $D = 5$.

a	**Measured** D_{ff}	N_4	$\{\#\mathrm{P}_{\mathrm{df}}\}_{D_{ff}}$
6	4	1857	1857
7	4	1127	1127
8	4	396	396
9	4	−336	20
10	4	−1069	11
11	4	−1803	2
12	5	−2538	8552*

This type of behaviour observed for $9 \leq a \leq 11$, with a small set of degree fall polynomials not predicted by Equation (12) has also been observed for other sets of parameters, so we do not believe that the parameters considered in Table 3 form a special case with regards to this. At this point we are not able to explain what causes these degree fall polynomials.

6 Security Estimation for EFLASH

Based on our results from previous sections, we now examine the suggested 80–bit security parameters for EFLASH versus classical and quantum adversaries (Table 1), using our formula for $N_4(n, d, a)$ in Equation (12). We find

$$N_4(80, 101, 5) = 8026 \qquad \text{and} \qquad N_4(160, 181, 5) = 22546,$$

which means that we expect that these sets of parameters will both admit a first fall degree of 4. From the experiments in the previous section we observed that when N_4 gives a positive number, it predicts the number of degree fall polynomials precisely. Furthermore, in all our experiments we find that the solving degree is at most one greater than the first fall degree. In Table 4 we have computed the complexity of solving the EFLASH equation system on these parameter sets using Equation (2) when D_{solv} is 4 and 5. We have chosen to include two values that are typically used for ω: 2.4 corresponding to the smallest known value (here up to 1 decimal precision), and 2.8 which is the value from Strassen's algorithm. From Table 4 we find that both sets of parameters fail to achieve 80–bit security in all scenarios, with the exception of the parameters versus quantum adversaries under the most pessimistic (for an attacker) assumptions ($\omega = 2.8$ and $D_{solv} = 5$).

For the suggested 128–bit security parameters in Table 1 we get a negative number for N_4 and so we are not able to predict the first fall degree for these cases. We have however seen that the minus modifier does not work as effectively for EFLASH as initially believed, and so it is very likely that these parameters will also fail to achieve their proposed security level.

Table 4: The complexity of solving the 80–bit security parameters suggested with respect to a classical adversary (left table) and a quantum adversary (right table).

D_{solv} ω	4	5
2.4	2^{50}	2^{59}
2.8	2^{58}	2^{69}

D_{solv} ω	4	5
2.4	2^{59}	2^{71}
2.8	2^{69}	2^{83}

7 Further Work

Following the attack described in this paper, one may wonder whether it is possible to fix the EFLASH scheme. We have seen that the relations β_1 and β_2 play a crucial role in the low first fall degree for this system. They are a direct consequence of the small base field, so it seems natural to try and choose a larger base field to mitigate this. The problem with this approach is that the condition for the central map to be injective, $\gcd(q^d - 1, q^\Theta + 1) = 1$, can only be satisfied when q is even. Furthermore, if $\mathbb{F}_q$ is chosen to be a small extension field of $\mathbb{F}_2$, then the system can always be solved as a system over $\mathbb{F}_2$, and so the existence of β_1, β_2 ultimately seems unavoidable. The minus modifier does help, but as we have seen it also strongly affects the efficiency of decryption in EFLASH. Since q^a needs to be low in order for decryption to be efficient, the designer is limited in the use of this modifier. For these reasons we cannot think of parameters that would result in instances of EFLASH that seem both efficient and secure.

A related question is whether the analysis presented here would have an impact on the security of the signature scheme PFLASH. As mentioned earlier, EFLASH and PFLASH share the same central map, and so the latter will also suffer from the same degree fall generators α, β_1 and β_2. The main difference is that signature schemes can allow a significant number of public polynomials to be removed without becoming inefficient. This can be seen from the suggested parameters for PFLASH in [7], where roughly one third of the public polynomials are removed. We are at this point not able to conclude either way on the security of the current PFLASH parameters, but our work shows the need for an updated security analysis against direct attacks for this scheme.

It will also be interesting to see if the ideas presented in this work may have an impact on other multivariate big field schemes that also benefit from the minus modifier. We point out that our methods not only predict the first fall degree, but also the number of degree fall polynomials obtained at this degree. It remains to be seen if this information can be used in other ways by an attacker.

One idea is to use this information in conjunction with the Joux–Vitse algorithm [19]. For example, if we predict k degree fall polynomials at degree D, then it may be the case that combining Mac_{D-1} and the k degree fall polynomials of degree $\leq D - 1$ leads to optimal parameter choices for this algorithm (see [19] for notation and more details on this). This could be paricularly interesting in cases where the first fall degree and solving degree may be far apart.

8 Conclusions

With the prospect of quantum computers becoming a reality, cryptographers have looked for quantum-safe public-key encryption algorithms that can replace RSA. The C^* scheme was proposed more than 30 years ago and is based on the MQ problem which is considered quantum-safe. However, the basic C^* scheme was quickly broken and cryptographers have since tried to find variants that may lead to secure quantum-safe public-key schemes. Some signature schemes built around the C^* construction have indeed withstood cryptanalysis; however it has proven to be much harder to come up with secure and efficient encryption algorithms based on it. EFLASH is one recent attempt.

However we have shown in this work that non-trivial degree fall polynomials arise rather early in a Gröbner basis attack when the central mapping is just a power-function and q is even (in particular when $q = 2$, as suggested for EFLASH). Two techniques that have been proposed for overcoming the deficiencies of the basic C^* system are to embed the plaintext space in a larger field, and to remove some of the polynomials in the public key before it is published. In this work we have seen that these two techniques to some extent work against each other, and we have shed some light on how much security is actually gained by the removal of some of the public polynomials.

During this work we were able to explain and give formulas for how many degree fall polynomials will appear at step degrees 3 and 4 in a solving algorithm. Experiments of fairly large instances show that our formulas give the exact number of degree fall polynomials when the predicted number is positive, giving confidence that we have captured the whole picture in our analysis. However, in some cases we get a few non-trivial degree fall polynomials when our formulas predict none, so more research is needed to explain these.

Based on our analysis we are very confident that we will indeed see a large number of non-trivial degree fall polynomials at step degree 4 for the suggested 80-bit security parameter sets for EFLASH. In all likelihood the solving degree for an actual EFLASH system will then be at most 5, giving solving complexities significantly lower than the claimed security. This means that EFLASH does not withstand direct Gröbner basis attacks, and should therefore be considered insecure.

References

1. M. Bardet, J.-C. Faugere, B. Salvy, and B.-Y. Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In *Proc. of MEGA*, volume 5, 2005.
2. M. Bardet, J.-C. Faugère, and B. Salvy. Complexity of Gröbner basis computation for Semi-regular Overdetermined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$. 2003. [Research Report] RR-5049, INRIA, inria-00071534.
3. L. Bettale, J. Faugère, and L. Perret. Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. *Des. Codes Cryptogr.*, 69(1):1–52, 2013.

4. F.-P.-A. Bouillaguet, Charles and G. Macario-Rat. Practical Key-recovery For All Possible Parameters of SFLASH.
5. D. Cabarcas, D. Smith-Tone, and J. A. Verbel. Key Recovery Attack for ZHFE. In *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, volume 10346 of *Lecture Notes in Computer Science*, pages 289–308. Springer, 2017.
6. R. Cartor and D. Smith-Tone. EFLASH: A New Multivariate Encryption Scheme. In C. Cid and M. Jacobson Jr., editors, *Selected Areas in Cryptography – SAC 2018*, volume 11349 of *Lecture Notes in Computer Science*, pages 281–299. Springer International Publishing, 2019.
7. M.-S. Chen, B.-Y. Yang, and D. Smith-Tone. PFLASH - secure asymmetric signatures on smart cards. Lightweight Cryptography Workshop 2015, 2015. `https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=926103`.
8. D. A. Cox, J. Little, and D. O'shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.
9. A. Diene, J. Ding, J. E. Gower, T. J. Hodges, and Z. Yin. Dimension of the linearization equations of the Matsumoto-Imai cryptosystems. In *International Workshop on Coding and Cryptography*, pages 242–251. Springer, 2005.
10. J. Ding, V. Dubois, B.-Y. Yang, O. C.-H. Chen, and C.-M. Cheng. Could SFLASH be Repaired? In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *Automata, Languages and Programming*, pages 691–701, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
11. J. Ding and T. J. Hodges. Inverting HFE systems is quasi-polynomial for all fields. In *Annual Cryptology Conference*, pages 724–742. Springer, 2011.
12. J. Ding and T. Kleinjung. Degree of regularity for HFE-. *IACR Cryptology ePrint Archive*, 2011:570, 2011.
13. J. Ding and D. Schmidt. Solving degree and degree of regularity for polynomial systems over a finite fields. In *Number Theory and Cryptography*, pages 34–49. Springer, 2013.
14. J. C. Faugere. A new efficient algorithm for computing Gröbner bases (F4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.
15. J. C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F 5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83. ACM, 2002.
16. J.-C. Faugere and A. Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In *Annual International Cryptology Conference*, pages 44–60. Springer, 2003.
17. P. Felke. On the Affine Transformations of HFE-Cryptosystems and Systems with Branches. In *Coding and Cryptography, International Workshop, WCC 2005, Bergen, Norway, March 14-18, 2005. Revised Selected Papers*, pages 229–241, 2005.
18. P. Felke. On the security of biquadratic C* public-key cryptosystems and its generalizations. *Cryptography and Communications*, pages 1–16, 2018.
19. A. Joux and V. Vitse. A crossbred algorithm for solving Boolean polynomial systems. In *International Conference on Number-Theoretic Methods in Cryptology*, pages 3–21. Springer, 2017.
20. A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Annual International Cryptology Conference*, pages 19–30. Springer, 1999.
21. J. Liu, Y. Yu, B. Yang, J. Jia, S. Wang, and H. Wang. Structural Key Recovery of Simple Matrix Encryption Scheme Family. *The Computer Journal*, 61, 10 2018.

22. T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and C. G. Günther, editors, *Advances in Cryptology — EUROCRYPT '88*, pages 419–453, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.
23. J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In *Annual International Cryptology Conference*, pages 248–261. Springer, 1995.
24. J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 33–48. Springer, 1996.
25. J. Patarin, N. Courtois, and L. Goubin. FLASH, a fast multivariate signature algorithm. In D. Naccache, editor, *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, volume 2020 of *Lecture Notes in Computer Science*, pages 298–307. Springer, 2001.
26. R. A. Perlner, A. Petzoldt, and D. Smith-Tone. Total Break of the SRP Encryption Scheme. In *Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers*, volume 10719 of *Lecture Notes in Computer Science*, pages 355–373. Springer, 2018.
27. J. Porras, J. Baena, and J. Ding. ZHFE, a New Multivariate Public Key Encryption Scheme. In *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, volume 8772 of *Lecture Notes in Computer Science*, pages 229–245. Springer, 2014.
28. C. Tao, A. Diene, S. Tang, and J. Ding. Simple Matrix Scheme for Encryption. In *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings*, volume 7932 of *Lecture Notes in Computer Science*, pages 231–242. Springer, 2013.
29. T. Yasuda and K. Sakurai. A Multivariate Encryption Scheme with Rainbow. In *Information and Communications Security - 17th International Conference, ICICS 2015, Beijing, China, December 9-11, 2015, Revised Selected Papers*, volume 9543 of *Lecture Notes in Computer Science*, pages 236–251. Springer, 2016.

Paper II

Analysis of Multivariate Encryption Schemes: Application to Dob

Morten Øygarden, Patrick Felke, and Håvard Raddum.

In *International Conference on Public-Key Cryptography (PKC)*, pages 155-183. Springer, 2021.
Invited to the Journal of Cryptology.

Analysis of Multivariate Encryption Schemes: Application to Dob

Morten Øygarden[1], Patrick Felke[2], and Håvard Raddum[1]

[1] Simula UiB
[2] University of Applied Sciences Emden/Leer
{morten.oygarden,haavardr}@simula.no,
patrick.felke@hs-emden-leer.de,

Abstract. In this paper, we study the effect of two modifications to multivariate public key encryption schemes: internal perturbation (*ip*), and Q_+. Focusing on the *Dob encryption scheme*, a construction utilising these modifications, we accurately predict the number of degree fall polynomials produced in a Gröbner basis attack, up to and including degree five. The predictions remain accurate even when fixing variables. Based on this new theory we design a novel attack on the Dob encryption scheme, which breaks Dob using the parameters suggested by its designers.

While our work primarily focuses on the Dob encryption scheme, we also believe that the presented techniques will be of particular interest to the analysis of other big–field schemes.

1 Introduction

Public key cryptography has played a vital role in securing services on the internet that we take for granted today. The security of schemes based on integer factorization and the discrete logarithm problem (DLP) is now well understood, and the related encryption algorithms have served us well over several decades.

In [25] it was shown that quantum computers can solve both integer factorization and DLP in polynomial time. While large scale quantum computers that break the actual implementations of secure internet communication have yet to be built, progress is being made in constructing them. This has led the community for cryptographic research to look for new public key primitives that are based on mathematical problems believed to be hard even for quantum computers, so called *post–quantum cryptography.*

In 2016 NIST launched a project aimed at standardizing post–quantum public key primitives. A call for proposals was made and many candidate schemes were proposed. The candidates are based on a variety of problems, including the shortest vector problem for lattices, the problem of decoding a random linear code, or the problem of solving a system of multivariate quadratic equations over a finite field (the MQ problem).

The first encryption scheme based on the MQ problem, named C^*, was proposed in [21] and was broken by Patarin in [23]. Since then, much work has gone into designing new central maps, as well as modifications that can enhance the security of existing ones. Several multivariate schemes have been proposed following C^*, for instance [24, 5, 27, 28]. While some of the schemes for digital signatures based on the MQ problem seem to be secure, it has been much harder to construct encryption schemes that are both efficient and secure. The papers [16, 22, 29, 26, 1], all present attacks on MQ-based public key encryption schemes, and as of now we are only aware of a few (e.g., [9, 32]) that remain unbroken.

In [20] a new kind of central mapping is introduced, which can be used to construct both encryption and signature schemes. The novel feature of the central mapping is that it has a high degree over an extension field, while still being easy to invert. The encryption variant proposed in [20] is called Dob and uses two types of modifications to its basic construction.

Our Contribution

The initial part of our work provides a theoretical analysis of (combinations of) two modifications for multivariate cryptosystems. The Q_+–modification was (to the best of our knowledge) first proposed in [20], while the second, internal perturbation (*ip*), has been in use for earlier schemes [12, 8, 9]. More specifically, we develop the tools for computing the dimension of the ideal associated with these modifications, at different degrees. This in turn provides key insights into the complexity of algebraic attacks based on Gröbner basis techniques.

As an application, we focus on the Dob encryption scheme proposed in [20]. We are able to deduce formulas that predict the exact number of first fall polynomials for degrees 3,4 and 5. These formulas furthermore capture how the number of degree fall polynomials changes as an attacker fixes variables, which also allows for the analysis of hybrid methods (see e.g., [3]).

Finally, the newfound understanding allow us to develop a novel attack on the Dob encryption scheme. Through analyzing and manipulating smaller, projected polynomial systems, we are able to extract and isolate a basis of the secret modifiers, breaking the scheme. While the details of the attack have been worked out for the Dob encryption scheme, we believe the techniques themselves could be further generalised to include different central maps and modifications.

Organisation

The paper is organized as follows. In Section 2 we recall the relation between $\mathbb{F}_2^d$ and $\mathbb{F}_{2^d}$, as well as the necessary background for solving multivariate systems over $\mathbb{F}_2$. In Section 3 we develop the general theory that explores the effectiveness of the modifications Q_+ and *ip* . Section 4 introduces the Dob scheme, and we deduce formulas that predict the number of degree fall polynomials for this construction. Experimental data verifying the accuracy of these formulas is presented in Section 5. In Section 6 we develop the novel attack on the Dob encryption scheme, using the information learned from the previous sections. Finally, sections 7 and 8 discuss and conclude the work.

Table of definitions

Throughout the paper we will use the notation in Table 1. We list it here for easy reference.

Term	Meaning
$B(n)$	$B(n) = \mathbb{F}_2[x_1, \dots, x_n]/\langle x_1^2 + x_1, \dots, x_n^2 + x_n \rangle$
$\overline{B}(n)$	$\overline{B}(n) = \mathbb{F}_2[x_1, \dots, x_n]/\langle x_1^2, \dots, x_n^2 \rangle$
$\overline{B}(n)_\nu$	The set of homogeneous polynomials of degree ν in n variables.
$\langle \mathcal{R} \rangle$	The ideal associated with the set of polynomials $\mathcal{R}$.
$\langle \mathcal{R} \rangle_\nu$	The ν–th degree part of a graded ideal $\langle \mathcal{R} \rangle$.
$\dim_\nu(\langle \mathcal{R} \rangle)$	The dimension of $\langle \mathcal{R} \rangle_\nu$ as an $\mathbb{F}_2$–vector space.
$\mathcal{P}^h$	A set of homogeneous quadratic polynomials over $\overline{B}(n)_2$
$\mathrm{Syz}(\mathcal{P}^h)_\nu$	The grade ν part of the (first) syzygy module of $\mathcal{P}^h$. (See section 2.1)
$\mathcal{T}(\mathcal{P}^h)_\nu$	The grade ν part of the trivial syzygy module of $\mathcal{P}^h$. (See section 2.1)
$\mathcal{S}(\mathcal{P}^h)_\nu$	$\mathcal{S}(\mathcal{P})_\nu = \mathrm{Syz}(\mathcal{P})_\nu / \mathcal{T}(\mathcal{P}^h)_\nu$.
Q_+, q_i, t	The Q_+ modifier, with $q_1, \dots, q_t$ added quadratic polynomials.
(ip), v_i, k	The *internal perturbation* modifier with $v_1, \dots, v_k$ linear forms.
$N_\nu^{(\alpha,\beta)}$	Estimate of the number of degree fall polynomials at degree ν.

Table 1: Notation used in the paper

2 Preliminaries

Multivariate big–field encryption schemes are defined using the field $\mathbb{F}_{q^d}$ and the d-dimensional vector space over the base field, $\mathbb{F}_q^d$. In practical implementations, $q = 2$ is very often used, and we restrict ourselves to only consider this case in the paper.

2.1 Polynomial System Solving

A standard technique used in the cryptanalysis of multivariate schemes, is to compute a Gröbner basis associated with the ideal $\langle p_i + y_i \rangle_{1 \leq i \leq m}$, for a fixed ciphertext $y_1, \dots, y_m$ (see for example [7] for more information on Gröbner bases). As we are interested in an encryption system, we can reasonably expect a unique solution in the boolean polynomial ring $B(n)$. In this setting the solution can be read directly from a Gröbner basis of any order.

One of the most efficient algorithms for computing Gröbner bases is F$_4$ [15]. In the usual setting, the algorithm proceeds in a step–wise manner; each step has an associated degree, D, where all the polynomial pairs of of degree D are reduced simultaneously using linear algebra. The degree associated with the most time consuming step is known as the *solving degree*, D_{solv}, and time complexity of F_4 can be estimated to be:

$$\text{Complexity}_{\text{GB}} = \mathcal{O}\Bigg(\bigg(\sum_{i=0}^{D_{solv}} \binom{d}{i} \bigg)^{\omega} \Bigg), \tag{1}$$

where $2 \leq \omega \leq 3$ denotes the linear algebra constant. Determining D_{solv} is in general difficult, but there is an important class of polynomial systems that is well understood. Recall that a homogeneous polynomial system, $\mathcal{F}^h = (f_1^h, \ldots, f_m^h) \in \overline{B}(n)^m$, is said to be *semi–regular* if the following holds; for all $1 \leq i \leq m$ and any $g \in \overline{B}(n)$ satisfying

$$gf_i^h \in \langle f_1^h, \ldots, f_{i-1}^h \rangle \text{ and } \deg(gf_i) < D_{reg}, \tag{2}$$

then $g \in \langle f_1^h, \ldots, f_i^h \rangle$ (note that f_i^h is included since we are over $\mathbb{F}_2$). Here D_{reg} is the *degree of regularity* as defined in [2], (for $i = 1$ the ideal generated by $\emptyset$ is the 0–ideal). We will also need a weaker version of this definition, where we say that $\mathcal{F}^h$ is D_0–semi–regular, if the same condition holds, but for $D_0 < D_{reg}$ in place of D_{reg} in eq. (2). An inhomogeneous system $\mathcal{F}$ is said to be (D_0–)semi–regular if its upper homogeneous part is. For a quadratic, semi–regular system $\mathcal{F}$ over $\overline{B}(n)$, the Hilbert series of $\overline{B}(n)/\mathcal{F}$ is written as (Corollary 7 in [2]):

$$T_{m,n}(z) = \frac{(1+z)^n}{(1+z^2)^m}, \tag{3}$$

and the degree of regularity can be computed explicitly as the degree of the first non–positive term in this series. Determining whether a given polynomial system is semi–regular may, in general, be as hard as computing a Gröbner basis for it. Nevertheless, experiments seem to suggest that randomly generated polynomial systems behave as semi–regular sequences with a high probability [2], and the degree of regularity can in practice be used as the solving degree in eq. (1). We will denote the degree of regularity for a semi–regular sequence of m polynomials in n variables as $D_{reg}(m, n)$. On the other hand, it is well known that many big–field multivariate schemes are not semi–regular (e.g., [16][5]). In these cases the *first fall degree* is often used to estimate the solving degree ([10][22]). The first fall degree, according to [10], will be defined in definition 2, but before that we recall the definition of a *Macaulay matrix* associated to a polynomial system.

Definition 1. *Let $\mathcal{P}$ be an (inhomogeneous) polynomial system in $B(n)$, of degree two. An (inhomogeneous)* Macaulay matrix *of $\mathcal{P}$ at degree D, $M_D(\mathcal{P})$, is a matrix with entries in $\mathbb{F}_2$, such that:*

1. *The columns are indexed by the monomials of degree $\leq D$ in $B(n)$.*
2. *The rows are indexed by the possible combinations $x^{\alpha}p_i$, where $1 \leq i \leq n$ and $x^{\alpha} \in B(n)$ is a monomial of degree $\leq D-2$. The entries in one row corresponds to the coefficients of the associated polynomial.*

Similarly, we define the homogeneous Macaulay matrix *of $\mathcal{P}$ at degree D, $\overline{M}_D(\mathcal{P})$, by considering $\mathcal{P}^h \in \overline{B}(n)$, only including monomials of degree D in the columns, and rows associated to combinations $x^{\alpha}p_i^h$, deg$(x^{\alpha}) = D-2$.*

Syzygies and Degree Fall Polynomials. Let $\mathcal{P}^h = (p_1^h, \dots, p_m^h) \in \overline{B}(n)_2^m$ denote a homogeneous quadratic polynomial system. The set $\mathcal{P}^h$ induces a map:

$$\begin{array}{rrl} \psi^{\mathcal{P}^h} : & \overline{B}(n)^m & \longrightarrow \overline{B}(n) \\ & (b_1, \dots, b_m) & \longmapsto \sum_{i=1}^m b_i p_i^h, \end{array} \tag{4}$$

which in turn splits into graded maps $\psi_{\nu-2}^{\mathcal{P}^h} : \overline{B}(n)_{\nu-2}^m \longrightarrow \overline{B}(n)_\nu$. The $\overline{B}(n)$–module $\mathrm{Syz}(\mathcal{P}^h)_\nu = \mathrm{Ker}(\psi_{\nu-2}^{\mathcal{P}^h})$ is known as the *ν–th grade of the (first) syzygy module of $\mathcal{P}^h$*. When $\nu = 4$, $\mathrm{Syz}(\mathcal{P}^h)_4$ will contain the *Koszul Syzygies*[3], which are generated by $(0, ..., 0, p_j^h, 0, ..., 0, p_i^h, 0, ..., 0)$ (p_j^h is in position i and p_i^h is in position j), and the *field syzygies*, which are generated by $(0, ..., 0, p_i^h, 0, ..., 0)$ (p_i^h in position i). These syzygies correspond to the cancellations $p_j^h p_i^h + p_i^h p_j^h = 0$ and $(p_i^h)^2 = 0$. As they are always present, and not dependent of the structure of $\mathcal{P}^h$, they are sometimes referred to as the *trivial syzygies.* More generally, we will define the submodule $\mathcal{T}(\mathcal{P}^h)_\nu \subseteq \mathrm{Syz}(\mathcal{P}^h)_\nu$ to be the ν–th graded component of the module generated by the Koszul and field syzygies, and denote $\mathcal{S}(\mathcal{P})_\nu = \mathrm{Syz}(\mathcal{P}^h)_\nu / \mathcal{T}(\mathcal{P}^h)_\nu$.

Definition 2. *The* first fall degree *associated with the quadratic polynomial system $\mathcal{P}$ is the natural number*

$$D_{ff} = min\{\ D \geq 2 \mid \mathcal{S}(\mathcal{P})_D \neq 0\ \}.$$

Representations over base and extension fields For any fixed isomorphism $\mathbb{F}_2^d \simeq \mathbb{F}_{2^d}$, there is a one–to–one correspondence between d polynomials in $B(d)$ and a univariate polynomial in $\mathbb{F}_{2^d}[X]/\langle X^{2^d} + X\rangle$ (see 9.2.2.2 in [4] for more details). For an integer j, let $w_2(j)$ denote the number of nonzero coefficients in the binary expansion of j. For a univariate polynomial $H(X)$, we define $\max_{w_2}(H)$ as the maximal $w_2(j)$ where j is the degree of a term occurring in H. Let $P(X)$ be the univariate representation of the public key of a multivariate scheme, and suppose there exists a polynomial $H(X)$ such that

$$\max_{w_2}(H(X)P(X)) < \max_{w_2}(H(X)) + \max_{w_2}(P(X)). \tag{5}$$

Then the multivariate polynomials corresponding to the product $H(X)P(X)$ will yield degree fall polynomials from (multivariate) degree $\max_{w_2}(H) + \max_{w_2}(P)$ down to degree $\max_{w_2}(HP)$.

It was mentioned in [16] that the presence of polynomials satisfying eq. (5) was the reason for Gröbner basis algorithms to perform exceptionally well on HFE–systems. Constructing particular polynomials that satisfy eq. (5) has also been a central component in the security analyzes found in [10] and [22].

[3] Here we follow the nomenclature used, for instance, in [18].

3 Estimating the Number of Degree Fall Polynomials

We start by introducing a general setting, motivated by the Dob encryption scheme which we will focus on later. Let $\mathcal{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a system of m quadratic polynomials over $B(n)$. Furthermore, consider the following two modifiers[4]:

1. The *internal perturbation* (*ip*) modification chooses k linear combinations $v_1, \dots, v_k$, and adds a random quadratic polynomial in the v_i's to each polynomial in $\mathcal{F}$.
2. The Q_+ modifier selects t quadratic polynomials $q_1, \dots, q_t$, and adds a random linear combination of them to each polynomial in $\mathcal{F}$.

Let H_{ip} be the random quadratic polynomials in $v_1, \dots, v_k$ and H_{Q_+} the random linear combinations of $q_1, \dots, q_t$. A modification of the system $\mathcal{F}$ can then be written as

$$\begin{aligned} \mathcal{P} : \mathbb{F}_2^n &\longrightarrow \mathbb{F}_2^m \\ x &\longmapsto \mathcal{F}(x) + H_{ip}(x) + H_{Q_+}(x). \end{aligned} \tag{6}$$

The problem we will be concerned with in this section is the following: given full knowledge of the degree fall polynomials of the system $\mathcal{F}$, what can we say about the degree fall polynomials of the system $\mathcal{P}$?

3.1 The Big Picture

Let $\mathcal{F}^h$ and $\mathcal{P}^h$ denote the homogeneous parts of the systems $\mathcal{F}$ and $\mathcal{P}$ respectively, and consider them over $\overline{B}(n)$. For a positive integer $\alpha \leq k$, we define V^α to be the homogeneous ideal in $\overline{B}(n)$ that is generated by all possible combinations of α linear forms from the *ip* modification, i.e.:

$$V^\alpha = \langle (v_{i_1} v_{i_2} \cdots v_{i_\alpha})^h \mid 1 \leq i_1 < i_2 < \ldots < i_\alpha \leq k \rangle. \tag{7}$$

In other words, V^α is the product ideal $\overbrace{V^1 \cdot V^1 \cdot \ldots \cdot V^1}^{\alpha}$. Similarly, for the quadratic polynomials associated with the Q_+ modifier we define Q^β for a positive integer $\beta \leq t$ to be the product ideal:

$$Q^\beta = \langle (q_{i_1} q_{i_2} \cdots q_{i_\beta})^h \mid 1 \leq i_1 < i_2 < \ldots < i_\beta \leq t \rangle. \tag{8}$$

Finally, for $0 \leq \alpha \leq k$ and $0 \leq \beta \leq t$, we define the ideal of different combinations of the modifiers, $M^{(\alpha,\beta)} = \langle V^\alpha, Q^\beta \rangle$, along with the boundary cases $M^{(\alpha,0)} = V^\alpha$, $M^{(0,\beta)} = Q^\beta$ and $M^{(0,0)} = \langle 1 \rangle$.

The following result is an important first step to understand how the degree fall polynomials in $\mathcal{F}$ behave when modifiers are introduced to the scheme.

[4] The authors of [20] named these two modifiers $\oplus$ and " + ". Note that in earlier literature (c.f. [31]), the " + " modification refers to a different modification than what is described in [20], and the $\oplus$ modification has been called *internal perturbation* (*ip*). (To the best of our knowledge, the " + " modification from [20] has not been used in earlier work). To avoid any confusion, we have chosen to stick with the name (*ip*) and use Q_+ for [20]'s "+"

Lemma 1. *Let $\mathcal{P}^h$, $\mathcal{F}^h$, $M^{(2,1)}$ be defined as above, and $\psi^{\mathcal{P}^h}$ be as defined in eq.* (4)*. Then $\langle \psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))\rangle$ and $\langle \psi^{\mathcal{P}^h}(Syz(\mathcal{F}^h))\rangle$ are homogeneous subideals of $\langle \mathcal{P}^h\rangle \cap M^{(2,1)}$.*

Proof. We show the statement for $\langle \psi^{\mathcal{P}^h}(\mathrm{Syz}(\mathcal{F}^h))\rangle$; the case of $\langle \psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))\rangle$ is similar. First note that $\psi^{\mathcal{P}^h}(\mathrm{Syz}(\mathcal{F}^h))$ is a group, as it is the image of a group under a group homomorphism. Secondly, for any element $\mathbf{a} = (a_1, \ldots, a_m) \in \mathrm{Syz}(\mathcal{F}^h)$, and any $r \in \overline{B}(n)$, we have $r\psi^{\mathcal{P}^h}(\mathbf{a}) = \psi^{\mathcal{P}^h}((ra_1, \ldots, ra_m))$, where also $(ra_1, \ldots, ra_m) \in \mathrm{Syz}(\mathcal{F}^h)$. It follows that $\psi^{\mathcal{P}^h}(\mathrm{Syz}(\mathcal{F}^h))$ is indeed an ideal.

The inclusion $\langle \psi^{\mathcal{P}^h}(\mathrm{Syz}(\mathcal{F}^h))\rangle \subseteq \langle \mathcal{P}^h\rangle$ follows directly from the definition of $\psi^{\mathcal{P}^h}$. For the other inclusion we note that, by construction, we can write $p_i^h = f_i^h + \sum_{j=1}^{t} b_{i,j} q_j^h + \sum_{j,l=0}^{k} c_{i,j,l}(v_j v_l)^h$, for all $1 \le i \le m$ and for suitable constants $b_{i,j}, c_{i,j,l} \in \mathbb{F}_2$, where f_i^h, p_i^h are the polynomials of $\mathcal{F}^h$ and $\mathcal{P}^h$ respectively. When $\mathbf{a} \in \mathrm{Syz}(\mathcal{F}^h)$, the f_i^h–parts in $\psi^{\mathcal{P}^h}(\mathbf{a})$ will vanish, and we are left with a polynomial that can be generated from the elements of V^2 and Q^1. Hence we also have $\langle \psi^{\mathcal{P}^h}(\mathrm{Syz}(\mathcal{F}^h))\rangle \subseteq M^{(2,1)}$.

In particular, there is the following chain of ideals

$$\langle \psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))\rangle \subseteq \langle \psi^{\mathcal{P}^h}(\mathrm{Syz}(\mathcal{F}^h))\rangle \subseteq \langle \mathcal{P}^h\rangle \cap M^{(2,1)} \subseteq M^{(2,1)}. \tag{9}$$

We now allow ourselves to be slightly informal, in order to see how this all relates in practice to the cases we are interested in. At each degree ν, the dimension $\dim_\nu(M^{(2,1)})$ of $M_\nu^{(2,1)}$ as a vector space over $\mathbb{F}_2$ can be seen as a measure of how much information the modifiers can hide. An interesting case from an attacker's point of view is when $\langle \psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))\rangle_{\nu_0}$ has the maximal dimension $\dim_{\nu_0}(\langle \psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))\rangle) = \dim_{\nu_0}(M^{(2,1)})$, for a relatively small ν_0. While 'excess' polynomials in $\langle \psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))\rangle_{\nu_0}$ will sum to 0 in $\overline{B}(n)$, there is a chance that the corresponding inhomogeneous polynomials will result in degree fall polynomials when treated over $B(n)$. In particular, this yields an upper bound $D_{ff} \le \nu_0$ on the first fall degree. We can do even better in practice.

Note that $(M^{(2,1)}\langle \mathcal{P}^h\rangle)_\nu$ will be a subspace of (the row space of) the Macaulay matrix $\overline{M}_\nu(\mathcal{P})$. As this matrix can be constructed by an attacker, we should count the possible combinations of polynomials from both $(M^{(2,1)}\langle \mathcal{P}^h\rangle)$ and the image of $\psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))$. Some caution is warranted when counting these combinations. For instance, $\psi^{\mathcal{P}^h}(ms) \in M^{(2,1)}\langle \mathcal{P}^h\rangle$ for any $m \in M^{(2,1)}$ and $s \in \mathcal{S}(\mathcal{F})$, so we need to be careful in order to not count the same elements twice. For now we will keep up with our informal theme and denote '$M^{(2,1)}\langle \mathcal{P}^h\rangle$ modulo these collisions' by $\mathcal{P}_{M^{(2,1)}}$. We will deal with it more properly when computing its dimension in section 3.3. We also show later, in appendix A, that $\langle \psi^{\mathcal{P}^h}(\mathcal{T}(\mathcal{F}^h))\rangle \subseteq M^{(2,1)}\langle \mathcal{P}^h\rangle$, which is why we will focus on $\langle \psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))\rangle$ (as opposed to $\langle \psi^{\mathcal{P}^h}(\mathrm{Syz}(\mathcal{F}^h))\rangle$).

We now have everything needed to discuss estimates of the number of degree fall polynomials at different degrees. We start by assuming that none of the degree fall polynomials we get from $\mathcal{S}(\mathcal{F})$ (under $\psi^{\mathcal{P}^h}$) can be reduced by lower–degree Macaulay matrices of $\mathcal{P}$. This allows us to directly use $\dim_\nu(\mathcal{S}(\mathcal{F}))$. We furthermore add $\dim_\nu(\mathcal{P}_{M^{(2,1)}})$, and subtract by $\dim_\nu(M^{(2,1)})$. This yields the expression for our first estimate of degree fall polynomials, $N_\nu^{(0,0)}$, at degree ν:

$$N_\nu^{(0,0)} = \dim_\nu(\mathcal{S}(\mathcal{F})) + \dim_\nu(\mathcal{P}_{M^{(2,1)}}) - \dim_\nu(M^{(2,1)}). \tag{10}$$

In a sense, $N_\nu^{(0,0)}$ can be thought of as estimating the number of degree fall polynomials, as an effect of 'over saturating' $M_\nu^{(2,1)}$. When $N_\nu^{(0,0)}$ is a positive number, this is the number of degree fall polynomials we expect to find (based on this effect); if $N_\nu^{(0,0)}$ is negative, there is no such over saturation, and we do not expect any degree fall polynomials at degree ν. The benefits of having the expression in eq. (10) is that the study of the relatively complex polynomial system $\mathcal{P}^h$ can be broken down to studying three simpler systems. The dimensions of $M^{(2,1)}$ and $\mathcal{P}_{M^{(2,1)}}$ can, in particular, be further studied under the assumptions that the modifiers form a semi–regular system. In addition to being a reasonable assumption as the modifiers are randomly chosen, this is also the ideal situation for the legitimate user, as this maximizes the dimension of $M^{(2,1)}$. Indeed, the study of $M^{(2,1)}$ and $\mathcal{P}_{M^{(2,1)}}$ will be continued in the following subsections. Before that, we will generalize the ideas presented so far, arriving at several expressions that can be used to estimate the number of degree fall polynomials.

Generalised Estimates of Degree Fall Polynomials. Let $M^{(\alpha,\beta)}\mathrm{Syz}(\mathcal{F})$ denote the module $\{ms \mid m \in M^{(\alpha,\beta)}, s \in \mathrm{Syz}(\mathcal{F})\}$ (which is well–defined since $\mathrm{Syz}(\mathcal{F})$ is a $\overline{B}(n)$–module), and define

$$\mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}} := [M^{(\alpha,\beta)}\mathrm{Syz}(\mathcal{F})]/\mathcal{T}(\mathcal{F}).$$

Instead of considering *all* the syzygies $\mathcal{S}(\mathcal{F})$, we can start with submodules of the form $\mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}}$. The benefit is that the ideal we need to 'over saturate' will now be $M^{(\alpha,\beta)}M^{(2,1)}$. In section 5 we will see several examples where this yields a better estimate than $N_\nu^{(0,0)}$. Following through with this idea, along with the same considerations discussed prior to eq. (10), we arrive at the following estimate for $\alpha, \beta \geq 0$:

$$\begin{aligned} N_\nu^{(\alpha,\beta)} = {} & \dim_\nu(\mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}}) - \dim_\nu(M^{(\alpha,\beta)}M^{(2,1)}) \\ & + \dim_\nu(\mathcal{P}^h_{M^{(\alpha,\beta)}M^{(2,1)}}). \end{aligned} \tag{11}$$

Recalling the convention that $M^{(0,0)} = \langle 1 \rangle$, this is indeed a generalisation of eq. (10).

We now have several different estimates for degree fall polynomials, varying with the choice of α, β. Any of these may be dominating, depending on the parameters of the scheme. The general estimate at degree ν is then taken to be their maximum:

$$N_\nu = \max\{0, N_\nu^{(\alpha,\beta)} \mid 0 \leq \alpha \leq k \text{ and } 0 \leq \beta \leq t\}. \tag{12}$$

Note in particular that if $N_\nu = 0$, then all our estimates are non–positive, and we do not expect any degree fall polynomials at this degree.

Consider now the main assumptions underlying these estimates. Firstly, recall that we assumed that none of the degree fall polynomials that can be made from $\psi^{\mathcal{P}}(\mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}})$ will be reduced to 0 when solving the system $\mathcal{P}$. Secondly, the formulas implicitly assume that all the polynomials in $M^{(\alpha,\beta)}M^{(2,1)}$ need to be reduced before we can observe degree fall polynomials. The third assumption, concerning $\mathcal{P}^h_{M^{(\alpha,\beta)}M^{(2,1)}}$, will be specified in section 3.3.

Finally, we stress that the aim of this section has been to investigate one of the aspects that can lead to a system exhibiting degree fall polynomials. The estimates presented should not be used without care to derive arguments about lower bounds on the first fall degree. Nevertheless, we find that in practice these estimates and their assumptions seem to be reasonable. With the exception of a slight deviation in only two cases (see Section 4.3), the estimates lead to formulas that are able to describe all our experiments for the Dob encryption scheme that will be investigated in Section 4.

3.2 Dimension of the Modifiers

The estimate given in eq. (11) requires knowledge of the dimension of (products of) the ideals $M^{(\alpha,\beta)}$. These will in turn depend on the chosen modifications V^α and Q^β. In this section we collect various results that will be needed to determine these dimensions. We start with the following elementary properties.

Lemma 2. *Consider $M^{(\alpha,\beta)} = (V^\alpha + Q^\beta)$, and positive integers $\alpha_0, \alpha, \beta_0, \beta, \nu$. Then the following holds:*

(i) $V^{\alpha_0}V^\alpha = V^{\alpha_0+\alpha}$ *and* $Q^{\beta_0}Q^\beta = Q^{\beta_0+\beta}$.

(ii) $V^{\alpha_0}Q^{\beta_0} \subseteq V^\alpha Q^\beta$ *if* $\alpha \leq \alpha_0$ *and* $\beta \leq \beta_0$.

(iii) $M^{(\alpha_0,\beta_0)}M^{(\alpha,\beta)} = M^{(\alpha_0+\alpha,\beta_0+\beta)} + V^{\alpha_0}Q^\beta + V^\alpha Q^{\beta_0}$.

(iv) $dim_\nu(M^{(\alpha,\beta)}) = dim_\nu(Q^\beta) + dim_\nu(V^\alpha) - dim_\nu(Q^\beta \cap V^\alpha)$.

(v)
$$dim_\nu(M^{(\alpha_0,\beta_0)}M^{(\alpha,\beta)}) = dim_\nu(M^{(\alpha_0+\alpha,\beta_0+\beta)}) + dim_\nu(V^{\alpha_0}Q^\beta) + dim_\nu(V^\alpha Q^{\beta_0}) - dim_\nu(M^{(\alpha_0+\alpha,\beta_0+\beta)} \cap V^{\alpha_0}Q^\beta) - dim_\nu(M^{(\alpha_0+\alpha,\beta_0+\beta)} \cap V^\alpha Q^{\beta_0}) - dim_\nu(V^{\alpha_0}Q^\beta \cap V^\alpha Q^{\beta_0}) + dim_\nu(M^{(\alpha_0+\alpha,\beta_0+\beta)} \cap V^{\alpha_0}Q^\beta \cap V^\alpha Q^{\beta_0}).$$

Proof. Properties (i) – (iv) follow from the appropriate definitions in a straightforward manner; we give a brief sketch of property (v) here. From property (iii) we know that $M^{(\alpha_0,\beta_0)}M^{(\alpha,\beta)}$ can be written as the sum of the three ideals $M^{(\alpha_0+\alpha,\beta_0+\beta)}$, $V^{\alpha_0}Q^\beta$ and $V^\alpha Q^{\beta_0}$. We start by summing the dimension of each of these three ideals individually. Any polynomial belonging to exactly two of these subideals is now counted twice, which is why we subtract by the combinations intersecting two of these ideals. Lastly, a polynomial belonging to all three of the subideals will, at this point, have been counted thrice, and then subtracted thrice. Hence, we add the dimension of intersecting all three subideals.

The dimension $\dim_\nu(V^\alpha)$ can be further inspected using the following result.

Lemma 3. *Suppose that $v_1, \dots, v_k$ are k linearly independent linear forms in $\overline{B}(n)$. Then*

$$dim_\nu(V^\alpha) = \sum_{\substack{i\geq\alpha, j\geq 0\\ i+j=\nu}} \binom{k}{i}\binom{n-k}{j} \tag{13}$$

holds under the conventions that $\binom{a}{b} = 0$ if $b > a$, and $\binom{a}{0} = 1$.

Proof. As $v_1, \dots, v_k$ are linearly independent, we can choose $n-k$ linear forms of $\overline{B}(n)$, $w_{k+1}, \dots, w_n$, that constitute a change of variables

$$\overline{B}(n) \simeq \overline{B}' = \mathbb{F}_2[v_1, \dots, v_k, w_{k+1}, \dots w_n]/\langle v_1^2, \dots, w_n^2\rangle.$$

For any monomial $\gamma \in \overline{B}'$, we will define $\deg_v(\gamma)$ as its degree in the $v_1, \dots, v_k$-variables, and $\deg_w(\gamma)$ as its degree in the variables $w_{k+1}, \dots, w_n$. The elements of V^α of (total) degree ν, is now generated (in $\overline{B}'$ as an $\mathbb{F}_2$–vector space) by all monomials γ such that $\deg_v(\gamma) \geq \alpha$ and $\deg_v(\gamma) + \deg_w(\gamma) = \nu$. The number of all such monomials are counted in eq. (13).

Lemma 4. *Let $q_1^h, \dots, q_t^h$ be a D_0–semi–regular system of homogeneous quadratic polynomials over $\overline{B}(n)$. Then, for any $2 \leq \nu < D_0$, we have*

$$dim_\nu(Q^1) = \binom{n}{\nu} - [z^\nu]T_{t,n}(z),$$

where $[z^\nu]T_{t,n}(z)$ denotes the coefficient of the monomial z^ν in the expansion of the series $T_{t,n}(z)$, as given in eq. (3)*.*

Proof. By assumption, the series $T_{t,n}(z)$ coincides with the Hilbert series of $\overline{B}(n)/Q^1$, for the terms with degree $2 \leq \nu < D_0$. From the additive property of the Hilbert function, we have that $\dim_\nu(Q^1) = \dim_\nu(\overline{B}(n)) - [z^\nu]T_{t,n}(z)$, and it is well–known that $\dim_\nu(\overline{B}(n)) = \binom{n}{\nu}$.

Lemma 5. *Suppose that $(v_1, \dots, v_k, q_1, \dots, q_t)$ is D_0–semi–regular, and consider $1 \leq \alpha \leq k$ and $1 \leq \beta \leq t$. Then*

$$(V^\alpha \cap Q^\beta)_\nu = (V^\alpha Q^\beta)_\nu,$$

holds for all $\nu < D_0$.

Proof. (Sketch) The product of any pair of ideals is contained in their intersection. For the other direction, consider a non–trivial element $e \in (V^\alpha \cap Q^\beta)_\nu$. Then, for some polynomials f_i, g_j, we can write $e = \sum f_i q_{i_1}^h \cdots q_{i_\beta}^h \in Q_\nu^\beta$, and $e = \sum g_j v_{j_1} \cdots v_{j_\alpha} \in V_\nu^\alpha$, which yields the syzygy

$$\sum f_i(q_{i_1}^h \cdots q_{i_\beta}^h) + \sum g_j(v_{j_1} \cdots v_{j_\alpha})^h = 0.$$

By assumption, all syzygies of degree $< D_0$ of $(v_1, \dots, v_k, q_1^h, \dots, q_t^h)$ will be generated by the field and Koszul syzygies of the v_i– and q_j^h–polynomials. It follows that (after possibly reducing by syzygies generated by only $q_1^h, \dots, q_t^h$) we have $f_i \in V^\alpha$. Similarly, we have $g_j \in Q^\beta$. In particular, $e \in V^\alpha Q^\beta$.

A general characterisation of the ideal $V^\alpha Q^\beta$ is trickier. We are content with discussing some special cases of its dimension, which will be of interest to us.

Example 1 *Suppose that $(v_1, \dots, v_k, q_1, \dots, q_t)$ is D_0–semi–regular, and let $1 \leq \alpha \leq k$ and $1 \leq \beta \leq t$.*

(a) The generators of $V^\alpha Q^\beta$ are of degree $\alpha + 2\beta$, hence $dim_\nu(V^\alpha Q^\beta) = 0$ for all $\nu < \alpha + 2\beta$. (This also holds without the D_0–semi–regularity assumption).

(b) Suppose furthermore that $D_0 > \alpha + 2\beta + 1$. Then $dim_{(\alpha+2\beta+1)}(V^\alpha Q^\beta) = \binom{t}{\beta} dim_{\alpha+1}(V^\alpha)$. To see this, note that $\langle V^\alpha Q^\beta \rangle_{\alpha+2\beta+1}$ is generated by elements of the form $v_{l_1} \dots v_{l_\alpha} q_{c_1} \dots q_{c_\beta} x_r$, where $1 \leq l_1 < \dots < l_\alpha \leq k$, $1 \leq c_1 < \dots < c_\beta \leq t$ and $1 \leq r \leq n$. The semi–regularity assumption assures that there will be no cancellations (save for the ones already accounted for in $dim_{\alpha+1}(V^\alpha)$).

(c) Suppose furthermore that $D_0 > \alpha + 2\beta + 2$, then $dim_{(\alpha+2\beta+2)}(V^\alpha Q^\beta) = \binom{t}{\beta} dim_{\alpha+2}(V^\alpha) - \binom{k}{\alpha}\left[\binom{t}{\beta}t - \binom{t}{\beta+1}\right]$. The reasoning is similar to (b), with the difference that $dim_{\alpha+2}(V^\alpha)$ will now include the polynomials of the form $q_c^h(v_{l_1} \dots v_{l_\alpha})^h$. There are $\binom{k}{\alpha}\left[\binom{t}{\beta}t - \binom{t}{\beta+1}\right]$ combinations of these that will reduce to 0 over $\overline{B}(n)$ (when multiplied with the combinations $q_{c_1}^h \dots q_{c_\beta}^h$).

3.3 Dimension of $\mathcal{P}_{M^{(\alpha,\beta)}M^{(2,1)}}$

As noted in section 3.1, we want $\mathcal{P}_{M^{(\alpha,\beta)}M^{(2,1)}}$ to be $M^{(\alpha,\beta)}M^{(2,1)}\langle \mathcal{P}^h \rangle$, modulo the polynomials of the form $\psi^{\mathcal{P}^h}(ms)$, for $ms \in \mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}M^{(2,1)}}$. Computing the dimension of $(M^{(\alpha,\beta)}M^{(2,1)}\langle \mathcal{P}^h \rangle)_\nu$ directly might be difficult, seeing that $\mathcal{P}^h$ depends on $M^{(2,1)}$. To tackle this, we start with the assumption that the cancellations in $M^{(\alpha,\beta)}M^{(2,1)}\langle \mathcal{P}^h \rangle$ are only generated by the 'generic' cancellations, and cancellations coming from the underlying structure, depending on $\mathcal{F}$. By 'generic' cancellations we mean those generated by the Koszul– or field syzygies in either the p_i^h– or m_j–polynomials. The assumption furthermore implies that the second type of cancellations will lie in the image of $\psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}M^{(2,1)}})$. Let $\mathcal{G}_{SR}$ be a system of homogeneous quadratic polynomials, of the same size and number of variables as $\mathcal{P}^h$, such that $\{V^1, Q^1, \mathcal{G}_{SR}\}$ forms a semi–regular system. With the assumption outlined above, we have

$$\dim_\nu(\mathcal{P}_{M^{(\alpha,\beta)}M^{(2,1)}}) = \dim_\nu(M^{(\alpha,\beta)}M^{(2,1)}\mathcal{G}_{SR}) - \dim_\nu(\mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}M^{(2,1)}}). \quad (14)$$

Indeed, any would–be cancellations that are over–counted in the term $\dim_\nu(M^{(\alpha,\beta)}M^{(2,1)}\mathcal{G}_{SR})$ would be subtracted in $-\dim_\nu(\mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}M^{(2,1)}})$.

$\mathcal{S}(\mathcal{F})_{M^{(\alpha,\beta)}M^{(2,1)}}$ requires knowledge of the underlying central map, $\mathcal{F}$, and will be dealt with in the next section. Computing the dimensions of the product ideal $M^{(\alpha,\beta)}M^{(2,1)}\mathcal{G}_{SR}$ has many similarities with the work that was done in the previous subsection. In particular, the dimension at degree ν is zero if the degrees of all of its generators are $> \nu$. We conclude with the following short example, which covers the other cases that will be the most relevant to us.

Example 2 *Let $\mathcal{G}_{SR}$ be a system of d homogeneous quadratic polynomials over $\overline{B}(n)$, such that $\{V^1, Q^1, \mathcal{G}_{SR}\}$ forms a semi–regular system. Then*

$$dim_\nu(M^{(2,1)}\mathcal{G}_{SR}) = n\big[dim_{\nu-2}(Q^1) + dim_{\nu-2}(V^2)\big],$$

holds for $\nu = 4, 5$.

4 Number of Degree Fall Polynomials in the Dob Encryption scheme

There are several ways to construct a central map $\mathcal{F} : \mathbb{F}_2^d \to \mathbb{F}_2^d$. For big–field schemes, the idea is to fix an isomorphism $\phi : \mathbb{F}_2^d \to \mathbb{F}_{2^d}$ between the vector space over the base field and the extension field, and choose two random invertible $d \times d$-matrices over $\mathbb{F}_2$, called S and T. $\mathcal{F}$ is then constructed as the composition $\mathcal{F} = S \circ \phi^{-1} \circ F \circ \phi \circ T$, where $F(X) \in \mathbb{F}_{2^d}[X]$, $\max_{w_2}(F) = 2$, and such that $F(X) = Y$ is easy to solve for any given Y. In particular, this ensures that $\mathcal{F}$ is a system of d quadratic polynomials, and ciphertexts can easily be decrypted with the knowledge of the secret S, T and F. There are two main ways in the literature to construct F with these properties:

1. $F(X) = X^e$, where $w_2(e) = 2$. This is the case for C^* [21].
2. $F(X) = \sum_{i=0}^{t} c_i X^{e_i}$, where we have $w_2(e_i) \leq 2$ for all i, and each e_i is bounded by a relatively small constant b. This is used in HFE [24].

Indeed, both C^* and HFE have been suggested with the *ip*–modification, known as PMI an ipHFE, respectively [8, 12]. These schemes were broken in [17, 14], by specialised attacks recovering the kernel of the linear forms of the *ip*–modification. Nevertheless, a later version of the C^* variant, PMI+ [9], also added the "$+$" modification in order to thwart this attack, and remains unbroken. We note that ipHFE, PMI and PMI+ all fits into the framework presented in section 3, and the techniques presented here can be used to understand their resistance against algebraic attacks (recall that the "+" modification does not increase the security versus algebraic attacks). A comprehensive study of these schemes are beyond the scope of this work, as we focus on a newer construction that utilizes both the *ip*– and Q_+–modification.

4.1 The Dob Encryption Scheme

The *Two–Face* family, introduced in [20], presents a third way to construct a function $F(X)$. Writing $Y = F(X)$, we get the polynomial equation

$$E_1(X, Y) = Y + F(X) = 0.$$

When F has the Two–Face property, it can be transformed into a different polynomial $E_2(X,Y) = 0$, which has low degree in X and have 2–weight at most 2 for all exponents in X. The degree of E_2 in Y can be arbitrary. Given Y, it is then easy to compute an X that satisfies $E_2(X,Y) = 0$, or equivalently, $Y = F(X)$.

For a concrete instantiation, the authors of [20] suggest the polynomial

$$F(X) = X^{2^m+1} + X^3 + X, \tag{15}$$

where $d = 2m-1$. Dobbertin showed in [13] that F is a permutation polynomial. In [20], based on the results of [13], it is further pointed out that

$$E_2(X,Y) = X^9 + X^6Y + X^5 + X^4Y + X^3(Y^{2^m} + Y^2) + XY^2 + Y^3 = 0$$

holds for any pair $Y = F(X)$. Note that F itself has high degree in X, but the highest exponent of X found in E_2 is 9 and all exponents have 2–weight at most 2.

The public key $\mathcal{F}$ associated with eq. (15) under the composition described at the beginning of section 4 is called *nude Dob*, and was observed in [20] to be weak. More precisely, experiments show that the associated multivariate system has solving degree three. Indeed, in appendix D we will show that this is the case for any d.

The (full) Dob encryption scheme is made by extending nude Dob with the two modifications, Q_+ and *ip*, as described at the beginning of section 3. The public key is the d quadratic polynomials $\mathcal{P}$, constructed according to eq. (6). The secret key consists of S, T, H_{ip} and H_{Q_+}. The plaintext space of the scheme is $\mathbb{F}_2^d$ and encryption is done by evaluating $y = \mathcal{P}(x)$, producing the ciphertext y.

To decrypt, the receiver of a ciphertext y guesses on the values of $v_i(x)$ and $q_j(x)$ for all $1 \leq i \leq k$ and $1 \leq j \leq t$, and computes the corresponding values of the polynomials in H_{ip} and H_{Q_+}. These values are added to y, removing the effect of the modifiers when the guess is correct. The resulting value y' is then the ciphertext of the nude Dob. This can be decrypted by first multiplying y' with S^{-1}, resulting in Y from the central mapping, which is then inverted using E_2 and multiplied with T^{-1} to recover the candidate plaintext x_0. The initial guess is then verified by checking if all $v_i(x_0)$ and $q_j(x_0)$ indeed evaluate to the guessed values.

In order for decryption to have an acceptable time complexity, the size of the modifications, k and t, can not be too large. To decrypt a ciphertext one must on the average do 2^{k+t-1} inversions of $\mathcal{P}$ before the correct plaintext is found. In [20] it is suggested to use $k = t = 6$ for 80–bit security.

For the remainder of this work, we let $\mathcal{F}$ and $\mathcal{P}$ denote the public keys of nude Dob and the (full) Dob encryption scheme, respectively.

4.2 Syzygies of the Unmodified Dob Scheme

The goal of this subsection is to estimate the dimension of $\mathcal{S}(\mathcal{F})_\nu$, for $\nu = 3, 4, 5$. We start by inspecting F (eq. (15)) over the extension field $\mathbb{F}_{2^d}[X]/\langle X^{2^d} + X\rangle$.

Note that $\max_{w_2}(F) = 2$, and consider the following polynomials:

$$G_1 = XF \qquad \text{and} \qquad G_2 = (X^{2^m} + X^2)F. \tag{16}$$

One finds that G_1 and G_2 are both products of F and a polynomial of 2–weight one, but the resulting polynomials still have $\max_{w_2}(G_i) = 2$. They are then examples of polynomials satisfying eq. (5) from section 2.1, and will correspond to $2d$ degree fall polynomials at degree three, down to quadratic polynomials. They form all the syzygies we expect at degree three, hence we set

$$\dim_3(\mathcal{S}(\mathcal{F})) = 2d. \tag{17}$$

Recall that it was noted in [20] that experiments of nude Dob had a solving degree of three, though the authors did not provide a proof that this is always the case. The presence of G_1 and G_2 ensures that the first fall degree of nude Dob is three. A complete proof that the solution of nude Dob can be found by only considering polynomials of degree three is a little more involved, and is included in appendix D.

Things get more complicated for dimensions $\nu > 3$. While we expect the two polynomials G_1 and G_2 to generate a significant part of the syzygies, we also expect there to be other generators, as well as cancellations to keep track of. Due to the complexity of fully characterizing the higher degree parts of $\mathcal{S}(\mathcal{F})$, we instead found an expression for its dimension at degrees $\nu = 4, 5$ experimentally. The experimental setup is further described at the end of this subsection. Note that the formulas we present in this subsection will be a multiple of d. This strongly suggests that all the syzygies of the system come from its extension field structure. These relations could then, in principle, be written out analytically as was the case for $\nu = 3$. In particular, this makes it reasonable to expect the formulas to continue to hold for larger values of d (i.e., beyond our experimental capabilities).

In the subsequent formulas we introduce the following notation, which will be useful to us later. Whenever counting the syzygies that can be generated from syzygies of lower degree, we will multiply by n (the number of variables in an associated multivariate system), as opposed to d. For instance, let $(g_{i,1} \dots, g_{i,d})$, $1 \leq i \leq d$ denote the d multivariate syzygies associated with G_1. Then $x_j(g_{i,1} \dots, g_{i,d})$, $1 \leq j \leq n$ are syzygies at $\nu = 4$, and we will count all of these as[5] nd. For the Dob encryption scheme we of course have $n = d$, so this distinction may seem unnecessary at the moment, but later, in section 5, we will also consider the case $n < d$ as an attacker may fix certain variables.

For $\nu = 4$, we find the following expression:

$$\dim_4(\mathcal{S}(\mathcal{F})) = (2n - 1)d, \tag{18}$$

where we note that the term $2nd$ has been generated by G_1 and G_2, as described above.

[5] Not all of these will be linearly independent in $\mathcal{S}(\mathcal{F})$. For example, the d syzygies associated with $(X^{2^m} + X^2)G_1$ will correspond to syzygies in $\mathcal{T}(\mathcal{F}^h)$. This does not really matter, as the expressions eq. (18) and eq. (19) corrects for this.

For $\nu = 5$, we have

$$\dim_5(\mathcal{S}(\mathcal{F})) = \left(2\binom{n}{2} - n - 2d - 20\right)d. \tag{19}$$

Once more, some of these terms can be understood from the syzygies of lower degrees. The contribution from the polynomials G_1 and G_2 from $\nu = 3$ will now be the $2\binom{n}{2}d$ term. The term '$-d$' from $\nu = 4$ will now cause the '$-nd$' term.

Experimental Setup. The experiments used to test eq. (18) and eq. (19) have been done as follows. The public polynomials of nude Dob are first generated, and we consider their upper homogeneous part, $\mathcal{F}^h$, over $\overline{B}(d)$. $\mathrm{Dim}_\nu(\mathcal{S}(\mathcal{F}))$ is computed as the dimension of the kernel of the homogeneous Macaulay matrix $\overline{M}_\nu(\mathcal{F}^h)$, minus $\dim_\nu(\mathcal{T}(\mathcal{F}^h))$. For $\nu = 4, 5$ we tested all odd d, $25 \leq d \leq 41$, all matching the values predicted by eq. (18) and eq. (19).

4.3 Degree Fall Polynomials of the (modified) Dob Scheme

We now have all the tools needed to write out explicit formulas for (variants of) the estimates $N_\nu^{(\alpha,\beta)}$, $\nu \leq 5$, for the Dob scheme. The approach for the formulas is as follows. Equation (11) is used as a foundation, and $\dim_\nu(\mathcal{S}(\mathcal{F}))$ is given according to section 4.2. For the dimension of the modifiers, and $\mathcal{P}_{M^{(\alpha,\beta)}M^{(2,1)}}$, we will combine the results discussed in section 3.2 and section 3.3. In particular, we will assume that the chosen modifying polynomials $\{v_1, \dots, v_k, q_1, \dots, q_t\}$ form a $(\nu+1)$–semi–regular system. The dimensions that are not covered by combining the results discussed so far, will be commented on separately. For the convenience of the reader, the non–trivial dimensions have been marked with an overbrace in the equations. The exceptions are eq. (24) and eq. (25), which are covered in greater depth in appendix B. Recall also our convention that $\binom{a}{b} = 0$, if $b > a$, and $\binom{a}{0} = 1$.

$\nu = 3$. At this degree we only consider $N^{(0,0)}$.

$$N_3^{(0,0)} = \overbrace{2d}^{\dim_3(\mathcal{S}(\mathcal{F}))} - \overbrace{\left((n-k)\binom{k}{2} + \binom{k}{3}\right)}^{\dim_3(V^2)} - \overbrace{nt}^{\dim_3(Q^1)} . \tag{20}$$

$\nu = 4$.

$$\begin{aligned} N_4^{(0,0)} = {} & \overbrace{(2n-1)d}^{\dim_4(\mathcal{S}(\mathcal{F}))} + \overbrace{d\left(t + \binom{k}{2}\right)}^{\dim_4(\mathcal{P}_{M^{(2,1)}})} - \overbrace{\left(t\binom{n}{2} - \binom{t}{2} - t\right)}^{\dim_4(Q^1)} \\ & - \overbrace{\left(\binom{k}{2}\binom{n-k}{2} + \binom{k}{3}(n-k) + \binom{k}{4}\right)}^{\dim_4(V^2)} + \overbrace{t\binom{k}{2}}^{\dim_4(Q^1 \cap V^2)} . \end{aligned} \tag{21}$$

At $\nu = 4$, we also consider the estimate $N_4^{(1,0)}$, i.e., multiplying everything with the k linear forms from the *ip*–modifier. In particular, this means that $(\mathcal{S}(\mathcal{F})_{M^{(1,0)}})_4$ is spanned by the combinations $v_j^h(g_{i,1}\ldots,g_{i,d})$, $1 \le j \le k$ and $1 \le i \le 2d$, where we recall that $(g_{i,1}\ldots,g_{i,d})$ denote the $2d$ multivariate syzygies associated with G_1 and G_2 (eq. (16))

$$\begin{aligned} N_4^{(1,0)} = \overbrace{2kd}^{\dim_4\left(\mathcal{S}(\mathcal{F})_{M^{(1,0)}}\right)} &- \overbrace{\left(\binom{k}{3}(n-k)+\binom{k}{4}\right)}^{\dim_4(V^3)} \\ &\overbrace{-\, t\left(k(n-k)+\binom{k}{2}\right)}^{\dim_4(Q^1V^1)}. \end{aligned} \tag{22}$$

$\boldsymbol{\nu = 5}$. At degree 5, $\mathcal{S}(\mathcal{F})_{M^{(2,1)}}$ (in eq. (14)) is no longer trivial. Indeed, it will now consist of the possible combinations $v_{j_1}^h v_{j_2}^h(g_{i,1}\ldots,g_{i,d})$ and $q_j^h(g_{i,1}\ldots,g_{i,d})$.

$$\begin{aligned} N_5^{(0,0)} = &\overbrace{\left(2\binom{n}{2}-n-2d-20\right)d}^{\dim_5(\mathcal{S}(\mathcal{F}))} - \overbrace{\left(t\binom{n}{3}-n\binom{t}{2}-tn\right)}^{\dim_5(Q^1)} \\ &\overbrace{-\left(\binom{k}{2}\binom{n-k}{3}+\binom{k}{3}\binom{n-k}{2}+\binom{k}{4}(n-k)+\binom{k}{5}\right)}^{\dim_5(V^2)} \\ &\overbrace{+\, t\left(\binom{k}{2}(n-k)+\binom{k}{3}\right)}^{\dim_5(Q^1\cap V^2)} \\ &\overbrace{+\, ntd+d\left(\binom{k}{2}(n-k)+\binom{k}{3}\right)-2dt-2d\binom{k}{2}}^{\dim_5\left(\mathcal{P}_{M^{(2,1)}}\right)}. \end{aligned} \tag{23}$$

As mentioned above, it is a bit more involved to derive $N_5^{(1,1)}$ and $N_5^{(2,1)}$, and we will refer to appendix B for more details. It would also appear that our assumptions are slightly off for these two estimates, as our experiments consistently yield $4d$ more degree fall polynomials than we are able to predict (see remark 3 for more details). We present the experimentally adjusted versions in Equations (24) and (25):

$$\begin{aligned} N_5^{(1,1)} = &\, d\left(k(2n-k-2)+t(2+k)+\binom{k}{3}+4\right)-\binom{t}{2}n-\binom{k}{3}\binom{n-k}{2} \\ &-\binom{k}{5}-\binom{k}{4}(n-k)-t\left(k\binom{n-k}{2}+\binom{k}{2}(n-k)-kt\right). \end{aligned} \tag{24}$$

$$
\begin{aligned}
N_5^{(2,1)} = 2d\left(\binom{k}{2} + t + 2\right) - \left(\binom{k}{4}(n-k) + \binom{k}{5}\right) \\
- t\left(\binom{k}{2}(n-k) + \binom{k}{3}\right) - \binom{t}{2}n.
\end{aligned}
\tag{25}
$$

5 Experimental Results on Degree Fall Polynomials

In the previous section we developed the theory on how to estimate the number of first fall polynomials, ending up with several formulas. This section is focused on the accuracy of these formulas, and how they can be used by an attacker. Note that since we are interested in the unique structure of the Dob encryption scheme, we will always assume that 'generic' degree fall polynomials do not interfere. More specifically, when inspecting a system of d polynomials in n variables at degree ν, we assume that d and n is chosen such that $D_{reg}(d, n) > \nu$.

5.1 Fixing Variables

The formulas separate d, the size of the field extension, and n, the number of variables. While the Dob encryption scheme uses $d = n$, an attacker can easily create an overdetermined system with $n < d$ by fixing some variables. This approach, known as the hybrid method, can be viewed as a trade–off between exhaustive search and Gröbner basis techniques, and its benefits are well–known for semi–regular sequences [3]. From eqs. (20) to (25), we find that for the relevant choices of parameters (d, t, k), a greater difference between n and d can increase the number of degree fall polynomials. This means that a hybrid method will have a more intricate effect on a Dob system, than what we would expect from random systems. To a certain extent, an attacker can "tune" the number of degree fall polynomials, by choosing the amount of variables to fix. Of course, if the intent is to find a solution of the polynomial system through a Gröbner basis, this comes at the added cost of solving the system 2^r times, where r is the number of fixed variables, but in section 6 we will present a different attack that circumvents this exponential factor.

Finally, one could ask whether it is reasonable to expect eqs. (20) to (25) to be accurate after fixing a certain number of variables. It is, for instance, possible that different degree fall polynomials will cancel out, as certain variables are fixed. However, if past experience with the hybrid method is any indicator, such cancellations are very rare, and we see no reason that the extension field structure increases the probability for such cancellations to happen. As we will see in section 5.3 this is supported by the experiments we have run; the formulas remain precise, even as n is varied.

5.2 Using the Degree Fall Formulas

We briefly recall how the formulas found in section 4.3 relate to the public polynomials of a Dob encryption scheme. Let $\mathcal{P}$ be the polynomial system associated with a Dob scheme of fixed parameters (d, n, t, k) (where n is as described in section 5.1). We expect the non–trivial dimension (i.e., the dimension of the part that is not generated by $\mathcal{T}(\mathcal{F})$) of the kernel of $\overline{M}_\nu(\mathcal{P})$ to be given by the maximal of the formulas $N_\nu^{(\alpha,\beta)}$, for $\nu = 3, 4, 5$.

If a step–wise algorithm such as F_4 is used, we expect the formulas to predict the number of degree falls polynomials, but *only* at the first fall degree. Suppose, for instance, that $N_3 = 0$, but $N_4 > 0$. Then this algorithm runs a second step at degree 4, using the newly found degree fall polynomials. This means that there are effectively more available polynomials in the system when (if) a step of degree 5 is performed, and in this case we do not expect the formulas we have for N_5 to be accurate.

Note in particular that if all the formulas we have are non–positive, an attacker is likely required to go up to step degree ≥ 6 in order to observe first fall polynomials.

5.3 Experimental Results

We have run a number of experiments with the Dob system of varying parameters (d, n, t, k). A subset of them is presented in table 2, and the rest can be found in appendix G. Gröbner bases of the systems were found using the F_4 algorithm implemented in the computational algebra system Magma. The script used for the experiments is available at [19].

In table 2 (and appendix G) we use the following notation. 'D_{ff}' is the experimentally found first fall degree. 'N (predicted)' is the number of first fall polynomials as predicted by the equations in section 4.3. 'N (Magma)' is the number of first fall polynomials read from the verbose output of Magma, written as 'degree : {# degree fall polynomials at this degree}'. The solving degree D_{solv} was found experimentally by Magma. This has been measured as the degree where the most time consuming step of the algorithm took place. In the instances where the algorithm did not run to completion due to memory constraints, we give D_{solv} as $\geq X$, where X is the degree of the step where termination occurred. The degree of regularity for semi–regular systems of the same size, $D_{reg}(d, n)$, is also given. 'Step Degrees' lists the degrees of the steps that are being performed by F_4 up until linear relations are found. Once a sufficient number of linear relations are found, Magma restarts F_4 with the original system, as well as these linear relations. This restart typically needs a few rounds before the entire basis is found, but its impact on the running time of the algorithm is negligible, which is why we have chosen to exclude it when listing the step degrees. For convenience, the step where first fall polynomials are found is marked in blue and the solving step marked in red. Purple is used to mark the steps where these two coincide.

Table 2: Degree fall polynomials for Dob encryption schemes of various parameters.

d	n	t	k	D_{ff}	N	N	D_{solv}	Step
		$(+)$	*(ip)*		(predicted)	(Magma)	$(D_{reg}(d,n))$	Degrees
53	53	0	0	3	$N_3^{(0,0)}$: 106	2:106	3 (9)	2,3,3
53	53	0	3	4	$N_4^{(0,0)}$: 1999	3:1999	4 (9)	2,3,4,4
53	53	3	0	4	$N_4^{(0,0)}$: 1596	3:1596	4 (9)	2,3,4,4
59	29	0	7	4	$N_4^{(1,0)}$: 21	3:21	5 (5)	2,3,4,4,5
37	25	2	3	4	$N_4^{(0,0)}$: 692	3:692	4 (5)	2,3,4,4
31	29	0	8	5	$N_5^{(1,1)}$: 478	4:478	5 (6)	2,3,4,5,5,5
31	30	0	8	5	$N_5^{(2,1)}$: 264	4:264	5 (6)	2,3,4,5,5,5,4
39	37	1	7	5	$N_5^{(2,1)}$: 136	4:136	≥ 6 (7)	2,3,4,5,5,5,6...
57	38	4	6	5	$N_5^{(1,1)}$: 2086	4:2086	≥ 6 (6)	2,3,4,5,5,6. . .
57	37	4	6	5	$N_5^{(1,1)}$: 2847	4:2847	5 (6)	2,3,4,5,5
129	50	6	6	5	$N_5^{(0,0)}$: 64024	4:64024	≥ 5 (6)	2,3,4,5,5...

A first observation is that in all experiments we find that 'N (predicted)' matches 'N (Magma)'. We also find that fixing variables affects the cross–over point between the formulas $N_\nu^{(\alpha,\beta)}$, as for instance seen in the rows 6 and 7. We note that $N_\nu^{(0,0)}$ tend to be dominant when $n << d$, and that $N_5^{(2,1)}$ only seems to have an impact when k is large and t is small.

For the majority of cases we observe that $D_{ff} = D_{solv}$ or $D_{solv} + 1$, but one should be careful in drawing any conclusions from this, seeing that our experiments are in practice limited to computations of $D < 6$. The relation between n and D_{solv} is also noteworthy. For instance, in row 9 we have $d = 57$ and $n = 38$; D_{ff} is 5, but $D_{solv} \geq 6$. In row 10 we fix one more variable, $n = 37$ (while keeping everything else as before), and find $D_{solv} = 5$.

Impact on Known Attacks. The solving degree of big field schemes are often estimated using the first fall degree. In cases where $D_{solv} > D_{ff}$, we observed instances where it is beneficial for an attacker to fix (a few) variables in order to lower the D_{solv} for each guess. Without a better understanding of D_{solv} and how it is affected by fixing variables, it seems that the approximation $D_{ff} \approx D_{solv}$ is conservative, yet reasonable, when estimating the complexity of direct/hybrid attacks against Dob system.

Another attack that may greatly benefit from the detailed formulas for degree fall polynomials obtained in section 3, is an adapted version of the distinguishing attack that was proposed for HFEv- (Section 5 in [11]). An attacker fixes random linear forms, and distinguishes between the cases where (some of) the fixed linear forms are in the span of $(v_1, \dots, v_k)$, and when none of them are, by the use of Gröbner basis techniques. Indeed, if *one* of the fixed linear forms are in this span, the number of degree fall polynomials will be the same as for a system with $k-1$

ip linear forms. Hence, a distinguisher based on the formulas presented here will work even without a drop in first fall degree, making the attack more versatile.

The deeper understanding for how the modifiers work allows for an even more efficient attack on the Dob scheme, which we now present.

6 A New Attack on the Dob Encryption Scheme

In the previous two sections we have studied how degree fall polynomials can occur in the Dob scheme, and have verified the accuracy of our resulting formulas through experiments. In this section we will show how all these insights can be combined to a novel attack. In section 6.1, we shall see that adding an extra polynomial to the system can leak information about the modification polynomials. We will see how this information can be used to retrieve (linear combinations of) the secret *ip* linear forms, and the homogeneous quadratic part of the Q_+ modification, in sections 6.2 and 6.3. We investigate how Gröbner basis algorithms perform with this extra information in section 6.4, and finally discuss the complexity of the entire attack in section 6.5.

6.1 Adding an Extra Polynomial

In section 3.1 we discussed how products of the modifiers and public polynomials affect the number of degree fall polynomials, through $\mathcal{P}_{M^{(2,1)}}$. One would also expect a similar effect to take place when adding a random polynomial to the system.

Consider a set of parameters for the Dob scheme, where the number of first fall polynomials is determined by $N_\nu^{(0,0)}$, for some $\nu > 3$. Let $\mathcal{P}$ be the public key of this scheme, and consider a randomly chosen homogeneous polynomial p_R of degree $\nu - 2$. As it is unlikely that the randomly chosen p_R has any distinct interference with $\mathcal{P}$, we expect $(\langle p_R\rangle \cap M^{(2,1)})_\nu$ to be generated by the possible combinations $p_R q_i^h$, and $p_R(v_j v_l)^h$. Furthermore, since the generators of $\mathcal{S}(\mathcal{F})$ have degree at least 3, we do not expect any collision between $\psi^{\mathcal{P}^h}(\mathcal{S}(\mathcal{F}))$ and $\langle p_R\rangle$ at degree ν (cf. section 3.3). From these considerations, we estimate the number of degree fall polynomials for the system $\{\mathcal{P}, p_R\}$ at degree ν to be:

$$N_\nu(\{\mathcal{P}, p_R\}) = N_\nu^{(0,0)}(\mathcal{P}) + t + \binom{k}{2}. \tag{26}$$

We ran a few experiments that confirm this intuition, the details are given in table 3. First, we confirmed that the degree fall polynomials of $\mathcal{P}$ were indeed given by $N_\nu^{(0,0)}(\mathcal{P})$, before applying Magma's implementation of the F$_4$ algorithm on the system $\{\mathcal{P}, p_R\}$. Recall also our convention that $\binom{0}{2} = 0$ when applying eq. (26).

With all this in mind, assume for the moment that $d = n$, and consider a homogeneous Macaulay matrix of $\{\mathcal{P}^h, p_R\}$ at degree ν, $\overline{M}_\nu(\{\mathcal{P}^h, p_R\})$. Any

Table 3: First fall polynomials of Dob encryption schemes with an added, randomly chosen polynomial p_R.

d	n	deg(p_R)	t	k	D_{ff}	N	N
			(Q_+)	(ip)		(predicted)	(Magma)
31	29	2	2	2	4	$N_4 : 705$	3:705
45	30	2	6	0	4	$N_4 : 342$	3:342
75	39	3	6	6	5	$N_5 : 4695$	4:4695
39	37	3	6	0	5	$N_5 : 9036$	4:9036

element in the (left) kernel of this matrix can in general be written as:

$$h_R p_R + \sum_{i=1}^{d} h_i p_i^h = 0, \tag{27}$$

for some homogeneous quadratic polynomials $h_i \in \overline{B}(d)_{\nu-2}$, $1 \leq i \leq d$, and $h_R \in \overline{B}(d)_2$. From the discussion above, we expect that the only way p_R contributes to these kernel elements is through the trivial syzygies, multiplications with p_i^h or p_R, and through multiplying with the generators of $M^{(2,1)}$. It follows that any polynomial h_R, from eq. (27), will be in the span of[6]

$$\mathcal{H} := \{p_1^h, \dots, p_d^h, p_R, q_1^h, \dots, q_t^h, (v_1 v_2)^h, \dots, (v_{k-1} v_k)^h\}. \tag{28}$$

Hence, given enough kernel elements of $\overline{M}_\nu(\{\mathcal{P}^h, p_R\})$, a set of generators of Span($\mathcal{H}$) can be found. In the next subsection we will generalise this observation to the case where a number of variables are fixed, i.e. $n < d$.

6.2 Gluing Polynomials

Let W_η denote a non-empty subset of r variables, i.e. $W_\eta = \{x_{\eta_1}, \dots, x_{\eta_r}\}$ for integers $1 \leq \eta_1 < \dots < \eta_r \leq d$. For $n = d - r$, there is a natural projection map associated to W_η, $\pi_{W_\eta} : B(d) \to B(d)/W_\eta \simeq B(n)$, that fixes the variables in W_η to 0. For any polynomial system $\mathcal{R}$ over $B(d)$, we will also write $\pi_{W_\eta}(\mathcal{R})$ to mean the system consisting of all polynomials in $\mathcal{R}$ under π_{W_η}. Suppose now that the number of first fall polynomials of a Dob system $\mathcal{P}$ is given by $N_\nu^{(0,0)}$, after fixing r variables to 0, i.e., $n = d - r$. Let W_η be the set of variables we fix. Following a similar line of reasoning as in section 6.1, we find that $\pi_{W_\eta}(h_R)$ from a kernel element of the Macaulay matrix associated with $\pi_{W_\eta}(\{P^h, p_R\})$ will no longer be in the span of $\mathcal{H}$, but rather lie in the span of $\pi_{W_\eta}(\mathcal{H})$. To ease notation, we will write $\mathcal{H}_\eta = \pi_{W_\eta}(\mathcal{H})$. A natural question is whether we can

[6] If p_R has degree ≥ 3, then the syzygy $p_R^2 + p_R = 0$ will be of degree $> \nu$. In this case p_R will not be among the generators of $\mathcal{H}$. We shall see later, in Remark (2), that the effect of p_R can also be removed in the degree 2 case, but at an added cost to the run time.

recover $\mathcal{H}$, by using different variable sets $W_1, \ldots, W_\rho$, and finding generators for the associated polynomial sets $\mathcal{H}_1, \ldots, \mathcal{H}_\rho$. We answer this question positively in this subsection.

Let $\widetilde{W_\eta} := \{x_1, \ldots, x_d\} \setminus W_\eta$ denote the complement of W_η, and note that $\mathcal{H}_\eta$ only contains information about the set of monomials $A(W_\eta) := \{x_i x_j \mid x_i, x_j \in \widetilde{W_\eta}\}$. In order to guarantee that the family $\mathcal{H}_1, \ldots, \mathcal{H}_\rho$ can give complete information about $\mathcal{H}$ we need to ensure that for any choice of $1 \leq i < j \leq d$, we have $x_i, x_j \in \widetilde{W_\eta}$ for at least one $1 \leq \eta \leq \rho$. In other words, the sets $\widetilde{W_1}, \ldots, \widetilde{W_\rho}$ must cover all possible quadratic monomials.

In practice, both d and the size r of the variable sets will be determined by the chosen Dob parameters[7]. This naturally leads to the following problem:

Definition 3 (The (Quadratic) (r,d)–Covering Problem). *For integers $1 < r < d-1$, find the smallest number ρ of variable sets, each of size r, such that*

$$A(W_1) \cup \ldots \cup A(W_\rho) = \{x_i x_j \mid 1 \leq i < j \leq d\}.$$

In Appendix E we present a constructive solution to this problem, which provides a good upper bound for ρ that is sufficient for our use case. The upper bound is given by the following lemma

Lemma 6. *The (Quadratic) (r,d)–Covering Problem is upper bounded by*

$$\rho \leq \binom{\left\lceil \frac{d}{\lfloor (d-r)/2 \rfloor} \right\rceil}{2}.$$

We illustrate the strategy for recovering $\mathcal{H}$ in the simple case when $d = 3r$. In this particular case, the method above yields $\rho = 3$, where W_1, W_2 and W_3 are pairwise, disjoint variable sets. We may write the following matrix:

$$\begin{array}{c} \\ H_1 \\ H_2 \\ H_3 \end{array} \begin{array}{cccccc} W_1 * W_1 & W_1 * W_2 & W_1 * W_3 & W_2 * W_2 & W_2 * W_3 & W_3 * W_3 \end{array} \\ \begin{array}{c} H_1 \\ H_2 \\ H_3 \end{array} \begin{bmatrix} 0 & 0 & 0 & * & * & * \\ * & 0 & * & 0 & 0 & * \\ * & * & 0 & * & 0 & 0 \end{bmatrix}$$

Here $W_i * W_j$, $i, j \in \{1, 2, 3\}$, is understood as a list of the monomials $x_a x_b$ where $x_a \in W_i$ and $x_b \in W_j$ (under any fixed ordering and $a \neq b$), and we write H_l to mean the rows associated with a fixed set of generators for $\mathcal{H}_l$. A 0 in the matrix means that the respective submatrix is the zero matrix, whereas $*$ denotes that the submatrix may take non-zero values. By construction, if the submatrix whose rows are H_l, and columns are $W_i * W_j$, is denoted by $*$, then it forms a set of generators for $\mathcal{H}$ restricted to the monomials in $W_i * W_j$. In

[7] We will see later that the gluing also requires some overlap between the variable sets, but this is not a problem for the parameters we are interested in.

particular, the submatrix with columns $W_3 * W_3$ and rows H_1 spans the same row-space as the submatrix with columns $W_3 * W_3$ and rows H_2. We will use this observation to construct a new matrix, denoted $H_1 \cap_{W_3} H_2$, that combine the useful information from H_1 and H_2 in the following procedure.

1. Since $\{p_1^h, \ldots, p_d^h, p_R\}$ are known, we start by finding $t+\binom{k}{2}$ vectors in the row space of H_2 that are linearly independent of $\pi_{W_2}(\{p_1^h, \ldots, p_d^h, p_R\})$. Denote the set of these vectors Y_2.
2. If $|W_3 * W_3| >> d + t + \binom{k}{2} + 1$, then for each vector $y_i \in Y_2$, we can expect a unique vector z_i in the row space of H_1, such that $y_i + z_i$ is 0 along the columns associated with $W_3 * W_3$. Find such an z_i for each $y_i \in Y_2$ through Gaussian elimination.
3. We now have $t + \binom{k}{2}$ pairs (y_i, z_i) that are used to define the $(t + \binom{k}{2}) \times \binom{d}{2}$ matrix $(H_1 \cap_{W_3} H_2)$ over $\mathbb{F}_2$ in the following manner. For each row index i_0 and column index j_0, we define the entry at $[i_0, j_0]$ to be

$$(H_1 \cap_{W_3} H_2)[i_0, j_0] = \begin{cases} y_{i_0}[j_0], & \text{if } j_0 \text{ is associated with a monomial in } W_3 * W_3 \\ y_{i_0}[j_0] + z_{i_0}[j_0], & \text{otherwise.} \end{cases}$$

The above procedure uses the common information found in the columns of $W_3 * W_3$ to combine vectors from H_1 and H_2. We may think of this as "gluing" polynomials along $W_3 * W_3$, hence the name of the technique. Now consider the following matrix.

$$\begin{array}{c cccccc} & W_1 * W_1 & W_1 * W_2 & W_1 * W_3 & W_2 * W_2 & W_2 * W_3 & W_3 * W_3 \\ (H_1 \cap_{W_3} H_2) & * & 0 & * & * & * & * \\ H_3 & * & * & 0 & * & 0 & 0 \end{array}$$

Note in particular that the polynomials associated with $(H_1 \cap_{W_3} H_2)$ forms a set of generators for $\pi_{W_1 * W_2}(\mathcal{H})$. In order to recover the information of the monomials in $W_1 * W_2$, we need only glue the vectors of $(H_1 \cap_{W_3} H_2)$, with combinations from the row space of H_3, using the same procedure as described above. Since both $(H_1 \cap_{W_3} H_2)$ and H_3 may take non–zero values at $W_1 * W_1$ and $W_2 * W_2$, we expect the gluing to result in $t + \binom{k}{2}$ unique polynomials if $|(W_1 * W_1) \cup (W_2 * W_2)| >> d + t + \binom{k}{2} + 1$. By construction, all of the resulting $t + \binom{k}{2}$ polynomials associated with $(H_1 \cap_{W_3} H_2) \cap_{W_1} H_3$ will be in the span of $\langle p_1^h, \ldots, p_d^h, p_R, q_1^h, \ldots, q_t^h, \ldots (v_i v_j)^h \ldots \rangle$, but none of them in the span of $\langle p_1^h, \ldots, p_d^h, p_R \rangle$. Hence we define $\mathcal{G}$ to be the set consisting of the polynomials $\{p_1^h, \ldots, p_d^h, p_R\}$, as well as the polynomials associated with $(H_1 \cap_{W_3} H_2) \cap_{W_1} H_3$, and note that $\mathcal{G}$ is, by construction, a system of polynomials that are linearly equivalent to $\mathcal{H}$.

As a proof of concept, we implemented retrieving $\mathcal{G}$ from a toy example of the Dob scheme, with $d = 45$, $t = 6$ and $k = 0$, using the method described above. The interested reader can find more details in appendix C, example 3.

The General Case In the case of a general family of variable sets $W_1, \ldots, W_\rho$, we will not be able to set up the straightforward matrices that was shown above. The gluing process can still be done in a similar, iterative manner. For instance, the submatrix associated with $\mathcal{H}_\eta$ will have 0 for each monomial $x_i x_j$ where x_i or $x_j \in W_\eta$, and $*$ otherwise. As above, we expect to be able to glue $\mathcal{H}_\eta$ with $\mathcal{H}_\psi$ if the number of their common $*$–monomials exceeds $d + t + \binom{k}{2} + 1$.

6.3 Retrieving the Linear Forms from *ip*

Suppose now that a set of generators $\mathcal{G}$ for Span($\mathcal{H}$) has been found, as described in section 6.2. The goal is to recover k linear forms that are generators for $\langle v_1, \ldots, v_k \rangle$. In order to simplify our arguments we will assume $k \geq 5$. The cases $2 \leq k \leq 4$ will be discussed in Remark 1.

Consider the kernel of the homogeneous Macaulay matrix $\overline{M}_3(\mathcal{G})$. From the definition of $\mathcal{H}$ (*eq.* (28)), we find that Span($\mathcal{H}$) contains all the homogeneous nude Dob–polynomials $f_1^h, \ldots, f_d^h$, as well as all the combinations $(v_i v_j)^h$, $1 \leq i < j \leq k$. Each polynomial $(v_i v_j)^h$ generates the two kernel elements $v_i(v_i v_j)^h$ and $v_j(v_i v_j)^h$ (which are trivial when working over $\overline{B}(d)$). The nude Dob–polynomials will generate the $2d$ kernel elements associated with the degree fall polynomials discussed in section 4.2. We would like to separate these two types of kernel elements. To this end, we suggest constructing a smaller system, $\mathcal{G}'$, by removing three polynomials from $\mathcal{G}$, that are in the span of $\{p_1^h, \ldots, p_d^h\}$. Indeed, the idea is that this will work as a self–imposed minus modifier, which will remove the effect of the Dob–polynomials of $\mathcal{G}$ at degree 3.

On the other hand, some kernel elements generated by combinations of the $(v_i v_j)^h$–elements can still be observed for $\mathcal{G}'$ at degree 3. More specifically, suppose $\mathcal{G}'$ was created from $\mathcal{G}$ by removing p_1^h, p_2^h and p_3^h. Then Span($\mathcal{G}'$) may not necessarily contain $(v_1 v_j)^h$ itself, for any $2 \leq j \leq k$, but it will contain the combination $(v_1 v_j)^h + b_{1,j} p_1^h + b_{2,j} p_2^h + b_{3,j} p_3^h$, for some $b_{1,j}, b_{2,j}, b_{3,j} \in \mathbb{F}_2$. By considering these equations for all j, and eliminating p_1^h, p_2^h and p_3^h, we find that Span($\mathcal{G}'$) will contain a polynomial $z_1 = \sum_{j=2}^{k} a_j (v_1 v_j)^h$, where $a_2, \ldots, a_k \in \mathbb{F}_2$ are not all 0, using the assumption that $k \geq 5$. The polynomial $v_1 z_1$ will subsequently be reduced to 0 over $\overline{B}(d)$. Similarly, we are guaranteed to find polynomials $z_2, \ldots, z_k$. We assume that these are the only contributors to the kernel. In particular, this means that each kernel element of $\overline{M}_3(\mathcal{G}')$ can be written as $\sum l_i g_i = 0$, with $g_i \in \mathcal{G}'$, and each l_i a linear form in Span($\{v_1, \ldots, v_k\}$). It follows that an attacker can retrieve a basis $v_1^*, \ldots, v_k^*$ of $\langle v_1, \ldots, v_k \rangle$, by determining k linearly independent l_i's from these kernel elements.

Remark 1. In the text above, we suggest removing $a = 3$ polynomials from $\mathcal{G}$, and assumed $k \geq 5$. We note that removing $a = 2$ polynomials is the smallest number needed for the prediction[8] $2d - ad$ to be non–positive, but this setting

[8] This formula follows a similar line of reasoning as in section 4.3, but with the minus modifier instead of Q_+. Cf. also [22] for a study on how the minus modifier affects degree fall polynomials for a somewhat related scheme.

could lead to some complications if the prediction turns out to be slightly off. Hence, $a \geq 3$ seems to be reasonable in order to exclude any interference from the Dob–structure at degree 3. Subsequently, we assume $k \geq a + 2$ in order to guarantee the existence of the polynomials z_i.

When $k = 4$, an attacker might choose $a = 2$, as described above. Some experiments also seem to suggest that we can still find enough generators for $\langle v_1, v_2, v_3, v_4 \rangle$, even when choosing $a = 3$ (even though our arguments does not hold in this case). Hence we do not expect $k = 4$ to pose any real challenge for an attacker. Lastly, if $k = 2, 3$, then $\binom{k}{2}$ is so small that an attacker can skip this step of the attack altogether and simply guess the values for the $v_i v_j$–combinations directly in the step described in the next section.

The retrieval of $\mathcal{G}$ and $v_1^*, \ldots, v_k^*$, as described in this subsection, has been implemented and verified on the toy example with parameters $d = 63$, $t = 1$ and $k = 4$. This is further described in example 4, in appendix C.

6.4 Solving the Extended Dob System

Assume now that an attacker has followed the steps described in the previous subsections, and has recovered a system $\mathcal{G}$ (section 6.2), as well as a basis $\{v_1^*, \ldots, v_k^*\}$ that generates $\langle v_1, \ldots, v_k \rangle$ (section 6.3). Now fix a set of generators $q_1^*, \ldots, q_k^*$ for the polynomials that are in $\mathrm{Span}(\mathcal{G})$, but not in

$$\mathrm{Span}(\{p_1^h, \ldots, p_d^h, p_R, (v_i^* v_j^*)^h \mid 1 \leq i < j \leq k\ \}).$$

With all this information, we consider the associated *extended Dob system*, $\mathcal{P}_E$, defined by:

$$\mathcal{P}_E := \{p_1, \ldots, p_d, p_R, q_1^*, \ldots, q_t^*, v_1^*, \ldots, v_k^*\}. \tag{29}$$

For any given ciphertext, an attacker with access to an extended Dob system can guess constant values for the polynomials $p_R, q_1^*, \ldots, q_t^*, v_1^*, \ldots, v_k^*$, and check the guess by finding a Gröbner basis for $\mathcal{P}_E$.

Remark 2. It might be in the interest of an attacker to find a system $\mathcal{P}_E$ that does not depend on the random element p_R. If this is the case, one can choose a second random element p'_R, and construct a second system $\mathcal{P}'_E = \{p_1, \ldots, p_d, p'_R, q_1'^*, \ldots, q_t'^*, v_1'^*, \ldots, v_k'^*\}$. A third system, $\mathcal{P}''_E = \{p_1, \ldots, p_d, q_1''^*, \ldots, q_t''^*, v_1''^*, \ldots, v_k''^*\}$, independent of the random elements p_R and p'_R, can now be found by determining generators for $\mathrm{Span}(\mathcal{P}_E) \cap \mathrm{Span}(\mathcal{P}'_E)$. Solving the latter system $\mathcal{P}''_E$ could be easier, as one does not have to guess values for the random element. As a result, this could be a beneficial trade–off if the attack is not dominated by finding extended Dob systems. By abuse of notation, we will also call this latter system $\mathcal{P}''_E$ an extended Dob system, as long as we are careful about the factor 2 in the complexity estimates.

In order to get a better understanding of solving extended Dob systems, we introduce the following modification for multivariate schemes.

Definition 4. *For a polynomial system $\mathcal{P}'$, we define the modification $\mathcal{L}_+$ by choosing l_0 linear forms, and appending linear combinations of them to each polynomial in $\mathcal{P}'$.*

Consider an extended Dob system, $\mathcal{P}_E$, where all coefficients have been guessed correctly. Since q_i^* does not contain any information about the linear part of the q_i–polynomials, it follows that $\mathrm{Span}(\mathcal{P}_E)$ will contain a Dob system that is only modified with the $\mathcal{L}_+$–modification, where $l_0 = t$. Moreover, this Dob system has d equations and $d - k$ variables[9]. The problem of estimating the complexity of finding a solution to $\mathcal{P}_E$, can then be reduced to that of estimating the complexity of finding a Gröbner basis for Dob with the $\mathcal{L}_+$–modification. While a thorough analysis of this $\mathcal{L}_+$–modification is beyond the scope of this work, we point out a couple of immediate properties.

Firstly, seeing that the first fall degree only depends on the upper homogeneous part of a polynomial system, it is unaffected by the $\mathcal{L}_+$–modification. In particular, we expect $2d$ degree fall polynomials at degree 3, as in the case for nude Dob (section 4.2). Secondly, if running an algorithm such as F_4, a second batch of degree fall polynomials will emerge at the first step of degree 4. To see this, note that Dob with the $\mathcal{L}_+$–modification can be written over the quotient ring $\mathbb{F}_{2^d}[X]/\langle X^{2^d} + X\rangle$ as

$$F_{\mathcal{L}_+}(X) = X(X^{2^m} + X^2) + L(X) + C_E, \tag{30}$$

where C_E is a constant in $\mathbb{F}_{2^d}$, and $L(X) = \sum_{i=1}^{m} c_i X^{2^i}$, with $c_i \in \mathbb{F}_{2^d}$, is a polynomial of binary weight one. $XF_{\mathcal{L}_+}$ is one of the combinations that induce degree fall polynomials at degree 3, and $X^4XF_{\mathcal{L}_+}$ will correspond to cubic[10] (multivariate) polynomials found at the second step of degree 3. Upon running a subsequent step at degree 4, the polynomial $L(X)X^4XF_{\mathcal{L}_+}$ will correspond to d multivariate cubic polynomials, and would hence be counted as degree fall polynomials.

We ran a few experiments for extended Dob systems, $\mathcal{P}_E$, the results of which can be found in appendix F.

6.5 Complexity of the Attack

The attack proposed in this section has two main parts. The first step is to construct an extended Dob system, $\mathcal{P}_E$. In the second step, an attacker solves this system for a particular ciphertext. Suppose an attacker fixes $d - n$ variables in order to find ρ polynomial systems $\mathcal{H}_1, \ldots, \mathcal{H}_\rho$ from the kernel elements of Macaulay matrices of degree $D_0 \geq 3$. The gluing operations, determining the linear forms $v_1^*, \ldots, v_k^*$, and the quadratic forms $q_1^*, \ldots, q_t^*$ only involve Macaulay

[9] Here we implicitly assume that k variables have been eliminated by the linear forms v_i^*.

[10] For nude Dob, the polynomial X^5F can be used to create linear polynomials (eq. (34)). The crucial difference is that in this case, the linear term X can be cancelled out at degree 3, whereas this is not possible for a general $L(X)$.

matrices of degree at most three. Hence, we expect the first step to be dominated by recovering generators for the polynomial systems $\mathcal{H}_i$. While the optimal choice of attack parameters may depend on the parameters of the Dob encryption scheme, as a rule of thumb it seems best to first minimize D_0, then n, and lastly ρ. In practice, minimizing n involves choosing the smallest n such that $D_{reg}(d,n) > D_0$, for a fixed d. Kernel elements of the resulting sparse, homogeneous Macaulay matrix can be found using a variant of the Wiedemann algorithm [30] (see also [6] for an implementation of a version adapted to the XL algorithm). Section VI of [30] shows that one kernel vector can be retrieved after three iterations with probability > 0.7, and as a simplification we estimate the complexity of finding a sufficient number of kernel elements in each of the ρ Macaulay matrices as $\frac{3}{0.7}\left(t+\binom{k}{2}\right)\binom{n}{D_0}^2\binom{n}{2}$. Recall from remark 2 that the first step is performed twice if the attacker wishes to remove the effect of p_R from $\mathcal{P}_E$; let $\delta = 1$ denote if this is the case, and $\delta = 0$ otherwise. It follows that the total attack complexity can be estimated as

$$\mathcal{C}_{\text{Attack}} = \max\left\{ 2^{\delta}\rho\frac{3}{0.7}\left(t+\binom{k}{2}\right)\binom{n}{D_0}^2\binom{n}{2}, \mathcal{C}_{\mathcal{P}_E,\delta} \,\middle|\, \delta \in \{0,1\}\right\}, \qquad (31)$$

where $\mathcal{C}_{\mathcal{P}_E,\delta}$ denotes the complexity of finding a solution for $\mathcal{P}_E$ (with or without p_R, depending on δ). While we do not have a general estimate for the complexity this second step, we discuss how to estimate it in the case of the 80–bit secure parameter set proposed in Section 2.4 of [20], in the following.

Security of the Suggested Parameters. Let $d = 129$, and $t = k = 6$ for the Dob encryption scheme. Using equations (3) and (21) we find that it is not possible to choose an n such that $N_4^{(0,0)}$ is positive, and $D_{reg}(129, n) > 4$. For degree 5, we find that $n = 50$ is the smallest number such that $N_5^{(0,0)}$ is positive, and $D_{reg}(129, 50) > 5$. Indeed, for this choice of parameters, we get:

$$N_5^{(0,0)}(129, 50, 6, 6) = 64024,$$

which is exactly the number of degree fall polynomials observed in the last row of table 2. For this choice of parameters, ρ is upper bounded by 15, due to lemma 6. In this case we can do even better, and use $\rho = 11$, as described in appendix E. Choosing $\delta = 1$, we find that the first step requires about 2^{63} operations. For step two, we note from table 4 in appendix F that the extended Dob system with modifications $t = k = 6$ has a solving degree of 4 in all the experiments we can run. Conjecturing that this behaviour extends to $d = 129$, we estimate the complexity of step two to be $\mathcal{C}_{\mathcal{P}_E,1} = 2^{12}\binom{123}{4}^{\omega}$, where the factor 2^{12} is the cost of finding the correct constants for $q_1^*, \ldots, q_6^*$ and $v_1^*, \ldots, v_6^*$. We have also used $123 = 129 - 6$ as the number of variables in this system, seeing that 6 variables are eliminated by the linear forms v_i^*.

Using $\omega = 2.4$, step two is estimated at 2^{67}. Using Strassen's algorithm with $\omega = 2.8$ (a rather pessimistic choice for an attacker as it assumes that it is not possible to take advantage of the sparse matrix structure of the systems), the estimate is 2^{77} for step two. Either option leads to a time complexity below the proposed 80–bit security.

7 The Security of the Dobbertin Permutation for Cryptographic Use

A natural question to ask is whether it is possible to find parameters for an efficient and secure version of the Dob encryption scheme. As our attack can be split in two phases, one could either try to make either of these infeasible for an attacker. We have seen that the modifications of the Dob encryption scheme is not as effective as initially hoped in hiding the degree fall polynomials of nude Dob. Furthermore, an attacker has a lot of flexibility in fixing variables, and gluing together polynomials that reveals information about the secret modifications. Even if secure parameters could be found for degree five, there is always the question of how the number of degree fall polynomials grows for larger degrees, i.e., determining N_ν for $\nu > 5$. For these reasons it seems likely that a significant increase to t, k, and/or d is needed, which would in turn have a large negative impact on decryption time and/or public key size.

Another idea could be to make solving the extended Dob system (phase two of the attack) infeasible. We note for instance that if the suggested parameters (see section 6.5) had instead used $t = 12$ and $k = 0$, then the extended system would not have been susceptible to a straightforward hybrid attack, since the computations would likely go up to at least degree five for each guess (see table 4 and the surrounding discussion in appendix F). We do, however, stress that essentially basing the security on the $\mathcal{L}_+$ modification (definition 4) seems like a risky endeavour: an attacker is still able to learn a lot of information about the structure of the system from its degree fall polynomials. This extra information could potentially be exploited in a more sophisticated attack.

On the other hand, the analysis presented in this work may not prove much of a threat to the use of the Dob permutation in signature schemes. The authors of [20] suggested the minus modification for a Dob signature scheme. While there is reason to believe that this modification has similar characteristics to the Q_+ modification (we note that the behaviour of the Q_+ modification is somewhat reminiscent to what was analysed in [22], though the central maps differ), the key difference is that signing time does not depend exponentially on the number of polynomials removed. For instance, in [20] a version of the Dob signature scheme is suggested using $d = 257$, and removing 129 polynomials for 128–bit security. It seems unlikely that our techniques will be successful when such a large number of modifications are in place, even when degrees > 5 are taken into account.

Lastly, we note that the analysis presented here has solely been focused on the Dobbertin permutation, and hence the security of the generalisations discussed in [20], i.e., the families ‘Pat’, ‘Mac’ and ‘Super Two–Face’, remains an open question.

8 Conclusions

We have presented an analysis of the effectiveness the Q_+ and *ip* modifications against algebraic attacks. The theory was then applied to the Dob encryption scheme, along with a novel attack on this construction. Not only does the attack break the suggested parameter set, its flexibility and effectiveness allows us to conclude that the Dobbertin permutation seems unsuited for use in encryption schemes.

There are several directions where the ideas presented here may inspire future work. Firstly, the modifications are treated as ideals, whose dimensions can be examined. If different types of modifications, such as minus and vinegar, can be included in this framework, it could lead to a deeper understanding of the security of an even larger subclass of big–field schemes. Secondly, the attack introduces new tools for the cryptanalysis of multivariate schemes. The gluing technique allows an attacker to collect useful information after fixing a number of variables. As there is no need for correct guesses, the exponential factor usually associated with hybrid methods is avoided. Furthermore, the technique does not rely on heuristic assumptions on the relation between the first fall and solving degrees.

In light of this, we believe that security analyses of big–field multivariate schemes ought not only focus on the first fall degree directly, but also how this degree changes when fixing variables. Cryptographers wishing to design encryption schemes by adding limited modification to an otherwise weak polynomial system should be particularly aware of the effect presented in this work.

Acknowledgements

Morten Øygarden has been funded by The Research Council of Norway through the project "qsIoT: Quantum safe cryptography for the Internet of Things". The authors would like to thank Carlos Cid for useful discussions on this work.

References

1. D. Apon, D. Moody, R. Perlner, D. Smith-Tone, and J. Verbel. Combinatorial rank attacks against the rectangular simple matrix encryption scheme. In *International Conference on Post-Quantum Cryptography*, pages 307–322. Springer, 2020.
2. M. Bardet, J.-C. Faugère, and B. Salvy. Complexity of Gröbner basis computation for Semi-regular Overdetermined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$. 2003. [Research Report] RR-5049, INRIA, inria-00071534.
3. L. Bettale, J.-C. Faugère, and L. Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2009.
4. C. Carlet. Vectorial boolean functions for cryptography. In Y. Crama and P. L. Hammer, editors, *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, pages 398–469. Cambridge University Press, 2010.
5. R. Cartor and D. Smith-Tone. EFLASH: A New Multivariate Encryption Scheme. In C. Cid and M. Jacobson Jr., editors, *Selected Areas in Cryptography – SAC 2018*, volume 11349 of *Lecture Notes in Computer Science*, pages 281–299. Springer International Publishing, 2019.

6. C.-M. Cheng, T. Chou, R. Niederhagen, and B.-Y. Yang. Solving quadratic equations with XL on parallel architectures. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 356–373. Springer, 2012.
7. D. A. Cox, J. Little, and D. O'shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.
8. J. Ding. A new variant of the Matsumoto-Imai cryptosystem through perturbation. In *International Workshop on Public Key Cryptography*, pages 305–318. Springer, 2004.
9. J. Ding and J. E. Gower. Inoculating multivariate schemes against differential attacks. In *International Workshop on Public Key Cryptography*, pages 290–301. Springer, 2006.
10. J. Ding and T. J. Hodges. Inverting HFE systems is quasi-polynomial for all fields. In *Annual Cryptology Conference*, pages 724–742. Springer, 2011.
11. J. Ding, R. Perlner, A. Petzoldt, and D. Smith-Tone. Improved cryptanalysis of HFEv- via projection. In *International Conference on Post-Quantum Cryptography*, pages 375–395. Springer, 2018.
12. J. Ding and D. Schmidt. Cryptanalysis of HFEv and internal perturbation of HFE. In *International Workshop on Public Key Cryptography*, pages 288–301. Springer, 2005.
13. H. Dobbertin. Almost perfect nonlinear power functions on gf (2/sup n/): the welch case. *IEEE Transactions on Information Theory*, 45(4):1271–1275, 1999.
14. V. Dubois, L. Granboulan, and J. Stern. Cryptanalysis of HFE with internal perturbation. In *International Workshop on Public Key Cryptography*, pages 249–265. Springer, 2007.
15. J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.
16. J.-C. Faugère and A. Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In *Annual International Cryptology Conference*, pages 44–60. Springer, 2003.
17. P.-A. Fouque, L. Granboulan, and J. Stern. Differential cryptanalysis for multivariate schemes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 341–353. Springer, 2005.
18. J. W. Hoffman, X. Jia, and H. Wang. *Commutative Algebra: An Introduction.* Stylus Publishing, LLC, 2016.
19. `https://github.com/Simula-UiB/Attack-On-The-Dob-Encryption-Scheme`.
20. G. Macario-Rat and J. Patarin. Two-face: New public key multivariate schemes. In *International Conference on Cryptology in Africa*, pages 252–265. Springer, 2018.
21. T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and C. G. Günther, editors, *Advances in Cryptology — EUROCRYPT '88*, pages 419–453, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.
22. M. Øygarden, P. Felke, H. Raddum, and C. Cid. Cryptanalysis of the multivariate encryption scheme EFLASH. In *Cryptographers' Track at the RSA Conference*, pages 85–105. Springer, 2020.
23. J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In *Annual International Cryptology Conference*, pages 248–261. Springer, 1995.
24. J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 33–48. Springer, 1996.

25. P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.
26. D. Smith-Tone and J. Verbel. A rank attack against extension field cancellation. In *International Conference on Post-Quantum Cryptography*, pages 381–401. Springer, 2020.
27. A. Szepieniec, J. Ding, and B. Preneel. Extension field cancellation: A new central trapdoor for multivariate quadratic systems. In *Post-Quantum Cryptography*, pages 182–196. Springer, 2016.
28. C. Tao, H. Xiang, A. Petzoldt, and J. Ding. Simple matrix–a multivariate public key cryptosystem (MPKC) for encryption. *Finite Fields and Their Applications*, 35:352–368, 2015.
29. Y. Wang, Y. Ikematsu, D. H. Duong, and T. Takagi. The secure parameters and efficient decryption algorithm for multivariate public key cryptosystem EFC. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 102(9):1028–1036, 2019.
30. D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE transactions on information theory*, 32(1):54–62, 1986.
31. C. Wolf and B. Preneel. Taxonomy of public key schemes based on the problem of multivariate quadratic equations. Cryptology ePrint Archive, Report 2005/077, 2005. `https://eprint.iacr.org/2005/077`.
32. T. Yasuda, Y. Wang, and T. Takagi. Multivariate encryption schemes based on polynomial equations over real numbers. In *International Conference on Post-Quantum Cryptography*, pages 402–421. Springer, 2020.

A Trivial Syzygies under $\psi^{\mathcal{P}^h}$

The image $\psi^{\mathcal{P}^h}(\mathcal{T}(\mathcal{F}^h))$, where $\mathcal{T}(\mathcal{F}^h)$ denotes the trivial syzygies, warrants some extra attention. Write $p_i^h = f_i^h + \sum_j a_{i,j}m_j$, where m_j denote the modifiers q_i^h and $(v_iv_l)^h$. Then the image of a Koszul syzygy is

$$\psi^{\mathcal{P}^h}((0,\ldots,0,f_{i_0}^h,0\ldots,0,f_{j_0}^h,0\ldots,0) = f_{i_0}^h\Bigg(\sum_j a_{j_0,j}m_j\Bigg) + f_{j_0}^h\Bigg(\sum_j a_{i_0,j}m_j\Bigg).$$

Note that the same polynomial can be written as

$$\Bigg(\sum_j a_{j_0,j}m_j\Bigg)p_{i_0}^h + \Bigg(\sum_j a_{i_0,j}m_j\Bigg)p_{j_0}^h = f_{i_0}^h\Bigg(\sum_j a_{j_0,j}m_j\Bigg) + f_{j_0}^h\Bigg(\sum_j a_{i_0,j}m_j\Bigg).$$

A similar observation can be done for the field syzygies, which ensures that $\langle\psi^{\mathcal{P}^h}(\mathcal{T}(\mathcal{F}^h))\rangle \subseteq M^{(2,1)}\langle\mathcal{P}^h\rangle$.

B Deriving Formulas for Degree Fall Polynomials

$\mathbf{N_5^{(1,1)}}$: Let us start by examining $(\mathcal{S}(\mathcal{F})_{M^{(1,1)}})_5$. The polynomials involving the quadratic polynomials from Q_+, q_i^h, are easy to classify, as they would only appear as products with the $2d$ degree fall polynomials at $\nu = 3$ (from eq. (16)). The elements containing the *ip* linear forms are slightly more involved. At first glance, the $\nu = 3$ syzygies will generate $2d \cdot \dim_2(V^1)$, but we also need to take into consideration the cancellations appearing at $\nu = 4$ (which sums up to the $-d$ term in eq. (18)). Assuming that none of these cancellations can be factorized by a linear form in $\mathrm{Span}(v_1, \ldots, v_k)$ (which is highly likely when $n >> k$), we will need to subtract by $-kd$ to account for these cancellations.

Turning our attention to the modifiers, we can combine (v) and (ii) from Lemma 2, to get

$$\dim_5(M^{(2,1)}M^{(1,1)}) = \dim_5(M^{(3,2)}) + \dim_5(V^1Q^1) - \dim_5(M^{(3,2)} \cap V^1Q^1).$$

Expecting that $(Q^2 \cap V^3)_5$ is empty, and using Lemma 2 (iv), we can further rewrite this as

$$\begin{aligned}\dim_5(M^{(2,1)}M^{(1,1)}) = {}& \dim_5(Q^2) + \dim_5(V^3) + \dim_5(V^1Q^1) \\ & - \dim_5(Q^2 \cap V^1Q^1) - \dim_5(V^3 \cap V^1Q^1).\end{aligned}$$

Example 1 (c) covers $\dim_5(V^1Q^1)$, and we will deal with the intersections through ad hoc arguments. We expect $\langle Q^2 \cap V^1Q^1\rangle_5$ to be generated by the the possible combinations $q_iq_jv_l$, so we estimate its dimension to be $k\binom{t}{2}$. Similarly, $\langle V^3 \cap V^1Q^1\rangle_5$ is expected to be generated by the combinations $v_iv_jv_rq_l$, and its dimension will be counted by $t\binom{k}{3}$.

Lastly, we examine $\mathcal{P}_{M^{(1,1)}M^{(2,1)}}$. At degree 5 the only possible combinations are $v_iv_jv_rp_l$, and $v_iq_jp_l$, and we need not have to worry with intersections, as we did for $\mathcal{P}_{M^{(2,1)}}$. All this information sums up to the following:

$$\begin{aligned}\big(N_5^{(1,1)}\big)' = {}& \overbrace{d\left(2k(n-k) + 2\binom{k}{2} + 2t - k\right)}^{\dim_5(\mathcal{S}(\mathcal{F})_{M^{(1,1)}})} - \overbrace{\binom{t}{2}n}^{\dim_5(Q^2)} \\ & - \overbrace{\left(\binom{k}{3}\binom{n-k}{2} + \binom{k}{4}(n-k) + \binom{k}{5}\right)}^{\dim_5(V^3)} \\ & - \overbrace{t\left(k\binom{n-k}{2} + \binom{k}{2}(n-k) + \binom{k}{3}\right) + k\left(t^2 - \binom{t}{2}\right)}^{\dim_5(Q^1V^1)} \\ & + \overbrace{\binom{t}{2}k}^{\dim_5(Q^2 \cap V^1Q^1)} + \overbrace{\binom{k}{3}t}^{\dim_5(V^3 \cap V^1Q^1)} + \overbrace{d\left(kt + \binom{k}{3}\right)}^{\dim_5(\mathcal{P}_{M^{(1,1)}M^{(2,1)}})} .\end{aligned} \tag{32}$$

Remark 3. We have run tests for $\dim_5(\mathcal{S}(\mathcal{F})_{M^{(1,1)}})$, $\dim_5(M^{(2,1)}M^{(1,1)})$ and $\dim_5(\mathcal{P}_{M^{(1,1)}M^{(2,1)}})$, and separately they agree with what we have counted above. However, when running tests for $(N_5^{(1,1)})'$ as a whole, we find that the theoretical formula presented in eq. (32) consistently undershoots the number of degree fall polynomials by $4d$. For this reason, we adjust eq. (24) in the main part of the text by this value, i.e., $N_5^{(1,1)} = \left(N_5^{(1,1)}\right)' + 4d$.

$\mathbf{N_5^{(2,1)}}$: The degree five part of $\mathcal{S}(\mathcal{F})_{M^{(2,1)}}$ will be in the span of the degree fall polynomials at degree 3 (from G_1 and G_2 eq. (16)), multiplied with the modifiers q_i and v_jv_l. An application of Lemma 2 (iv) and (v) leads to

$$\dim_5(M^{(2,1)}M^{(2,1)}) = \dim_5(V^4) + \dim_5(Q^2) + \dim_5(V^2Q^1)$$

Example 1 (b) is used to compute $\dim_5(V^2Q^1)$, and we furthermore expect no polynomials of degree five in $\mathcal{P}_{M^{(2,1)}M^{(2,1)}}$. All this sums up to the following estimate:

$$\begin{aligned}\left(N_5^{(2,1)}\right)' = {} & \overbrace{2d\left(\binom{k}{2}+t\right)}^{\dim_5(\mathcal{S}(\mathcal{F})_{M^{(2,1)}})} - \overbrace{\left(\binom{k}{4}(n-k)+\binom{k}{5}\right)}^{\dim_5(V^4)} \\ & - t\overbrace{\left(\binom{k}{2}(n-k)+\binom{k}{3}\right)}^{\dim_5(Q^1V^2)} - \overbrace{\binom{t}{2}}^{\dim_5(Q^2)} n \ .\end{aligned} \tag{33}$$

Similarly to what was discussed in remark 3, we also find that the theoretically predicted $\left(N_5^{(2,1)}\right)'$ is off by $4d$ in experiments. Hence, we adjust for this in eq. (25) by setting $N_5^{(2,1)} = \left(N_5^{(2,1)}\right)' + 4d$.

C Experimental Examples

In order to test our attack strategy, we implemented and verified the following two toy examples in Magma. We checked that we do indeed find $t + \binom{k}{2}$ polynomials that are in $\mathrm{Span}(p_1^h, \ldots, p_d^h, p_R, q_1^h, \ldots, q_t^h, \ldots, (v_iv_j)^h, \ldots)$, but not in $\mathrm{Span}(p_1^h, \ldots, p_d^h, p_R)$. For the latter example we also verified that we retrieve k linear forms in $\mathrm{Span}(v_1, \ldots, v_k)$. The implementation is available at [19].

Example 3 *The first toy example is that of a Dob encryption scheme where $d = 45$, $t = 6$ and $k = 0$. Fixing no variables, $n = d$, we find that the equations eqs.* (20) *to* (22) *are negative, and hence we do not expect this system to have any degree fall polynomials at degrees ≤ 4. If we instead fix 15 variables, $n = 30$, we get $N_4 = N_4^{(0,0)} = 336$. If we, in addition, add a randomly chosen homogeneous quadratic polynomial p_R to the system, we get 342 degree fall polynomials at degree 4 (see eq.* (26)*).*

Following section 6.2, we split the variables into three disjoint sets: $W_1 = \{x_1, \ldots, x_{15}\}$, $W_2 = \{x_{16}, \ldots, x_{30}\}$ and $W_3 = \{x_{31}, \ldots, x_{45}\}$. Let $\mathcal{P}$ denote the public polynomials of the scheme, and for $i = 1, 2, 3$, compute the kernel of $\overline{M}_4(\pi_{W_i}(\{\mathcal{P}, p_R\}))$. Let $\mathcal{H}_i$ be the system of polynomials that gets multiplied with p_R in creating these kernels, and find a basis for it (which will be of dimension

$d+t+1=52$). The polynomial sets $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ are now glued together as detailed in section 6.2. Note in particular that we do not expect any problems with the gluing, seeing that $|W_3 * W_3| = \binom{15}{2} = 105 > 52$.

Example 4 *The second toy example had parameters $d = 63$, $t = 1$ and $k = 4$. Fixing no variables, we find $N_4 = N_4^{(1,0)} = 25$. If we fix 21 variables, we find that $N_4^{(0,0)}$ is dominant, i.e., $N_4 = N_4^{(0,0)} = 445$. Adding a random quadratic polynomial yields 452 degree fall polynomials at degree 4 (see eq. (26)).*

As in the example above, we divide into three equal sets: $W_1 = \{x_1, \ldots, x_{21}\}$, $W_2 = \{x_{22}, \ldots, x_{42}\}$ and $W_3 = \{x_{43}, \ldots, x_{63}\}$, and followed the steps described in Sections 6.2 and 6.3.

D Nude Dob is Fully Broken at Degree 3

In [20] it is stated that experiments indicate that nude Dob has a solving degree 3. We will show that this is indeed the case. In the following, all computations are over either $B(d)$ or $\mathbb{F}_{2^d}[X]/\langle X^{2^d} + X\rangle$. Consider $F(X) + C = 0$, where F is as defined in eq. (15), and $C \in \mathbb{F}_{2^d}$ a ciphertext we wish to solve for. Tedious hand calculation shows that

$$\left.\begin{array}{l} C^2\left((1+X^2)(XF)\right)^{2^m} + (C^2+X^4)F^{2^{m+1}} + X^{2^{m+1}}(C^2F^2+F^4) \\ +C^{2^m+2}F^{2^m} + C^{2^m+2}XF + \left((X^4+X^2)(XF) + (X^2+X)CF\right)^2 \end{array}\right\}(I)$$
$$= \left.\begin{array}{l} X^{16} + (C^{2^{m+1}} + C^{2^m+2} + C^4 + C^2)X^4 \\ +(C^2+1)X^8 + (C^{2^m+2}+C^4)X^2 + C^{2^m+3}X. \end{array}\right\}(II) \qquad (34)$$

The polynomial (II) is linearized and of degree 16. Thus its zeros form subspace of dimension at most 4. It follows that (II) will correspond to a linear system $l_1(x_1, \ldots, x_d) = \ldots = l_d(x_1, \ldots, x_d) = 0$ of rank at least $d-4$, from which a plaintext from an intercepted ciphertext can be easily recovered.

It remains to show that polynomial (I) can be computed from the public key, using polynomials of degree at most 3. Recall from section 4.2 that XF correspond to degree fall polynomials down to degree two. Each such polynomial will correspond to a solution $a_{i,j}, \gamma_{i,j}, \beta_i, \delta \in \mathbb{F}_2$, for the equation

$$\begin{gathered}(a_{1,0} + a_{1,1}x_1 + \ldots + a_{1,d}x_d)p_1 + \ldots + (a_{d,0} + a_{d,1}x_1 + \ldots + a_{d,d}x_d)p_d + \\ \sum \gamma_{i,j}x_ix_j + \sum \beta_i x_i + \delta = 0.\end{gathered}$$

As described in section 4.2, we expect this solution space to be of dimension $2d$. Let $d_1, \ldots, d_{2d}$ be a basis of the degree fall polynomials derived in this step, i.e., a basis of the partial polynomials $\sum \gamma_{i,j}x_ix_j + \sum \beta_i x_i + \delta$ from this solution space. Since the only terms in (I) of 2–weight four are generated from (XF) and can be substituted by the above degree fall polynomials, we may find solutions $a'_{i,j}, \beta'_i, \delta' \in \mathbb{F}_2$ for the following system.

$$\begin{gathered}(a'_{1,0} + a'_{1,1}x_1 + \ldots + a'_{1,d}x_d)p_1 + \ldots + (a'_{d,0} + a'_{d,1}x_1 + \ldots + a'_{d,d}x_d)p_d + \\ (a'_{d+1,1}x_1 + \cdots + a'_{d+1,d}x_d)d_1 + \cdots + (a'_{3d,0} + a'_{3d,1}x_1 + \cdots + a'_{3d,d}x_d)d_{2d} + \\ \beta'_1x_1 + \cdots + \beta'_dx_d + \delta' = 0.\end{gathered}$$

In particular, the linear forms from (II) can be written $l_j = \sum \beta'_i x_i + \delta'$, where the β' and δ'–coefficient will be associated with solutions of this system.

Since all the systems described above only includes polynomials of degree at most three, finding a plaintext remains practical, even for $d = 129$. In practice one can also apply algorithms that can exploit degree fall polynomials, such as F$_4$. If this is the case, the polynomials associated with XF will be found in the first step of degree three, and the linear polynomials (II) will be found in the ensuing step of degree three.

E Proof of Lemma 6

By a slight abuse of notation we will consider $\widetilde{W}_\eta$ to include integers, by listing the index of the variables it contains. Recall the (r, d) covering problem, which can be stated as follows: for given d and $r < d-1$, find ρ subsets $\widetilde{W}_\eta \subset \{1, \dots, d\}$ of size $d - r$, such that for any pair (i, j) where $1 \le i < j \le d$, $\{i, j\} \subset \widetilde{W}_\eta$ for at least one η.

Proof (of Lemma 6). Let $s = \lfloor (d-r)/2 \rfloor$. We divide $\{1, \dots, d\}$ into blocks of size s:

$$C_b = \{(b-1)s + 1, \dots, bs\}, \text{ for } 1 \le b \le \lfloor d/s \rfloor$$

.

Let the sets $\widetilde{W}_\eta$ for $1 \le \eta \le \binom{\lfloor d/s \rfloor}{2}$ be defined as the union of C_a and C_b, for all choices of $1 \le a < b \le \lfloor d/s \rfloor$. In the case $d - r$ is odd, we also add one arbitrary extra number to each set to make sure that each $\widetilde{W}_\eta$ contains exactly $d - r$ numbers.

Any $\{i, j\} \subset \{1, \dots, s\lfloor d/s \rfloor\}$ will then be contained in at least one $\widetilde{W}_\eta$. If both i and j belong to the same block C_b, then all $\widetilde{W}_\eta$ involving C_b will contain $\{i, j\}$. If $i \in C_a$ and $j \in C_b$ for $a \neq b$, then the set $\widetilde{W}_\eta = C_a \cup C_b$ will contain $\{i, j\}$. Hence the $\binom{\lfloor d/s \rfloor}{2}$ sets constructed will cover all pairs from $\{1, \dots, s\lfloor d/s \rfloor\}$.

If s divides d we are done. Otherwise, to cover all pairs of numbers in $\{1, \dots, d\}$ it is sufficient to create $\lfloor d/s \rfloor$ new $\widetilde{W}$-sets consisting of
$\{s\lfloor d/s \rfloor + 1, \dots, d\} \cup C_b \cup \{s - (d - s\lfloor d/s \rfloor) \text{ extra numbers}\}$,
where $1 \le b \le \lfloor d/s \rfloor$, and the extra numbers are arbitrary. The total number of sets will then be $\binom{\lceil d/s \rceil}{2}$, and replacing s with $\lfloor (d-r)/2 \rfloor$ we get Lemma 6.

For the particular case $d = 129, r = 79$ (which is used in Section 6.5) we get $\rho \le 15$. Doing the exercise in practice we find that $\rho = 11$ is sufficient to solve the problem by extending the block C_5 to cover all numbers $101, \dots, 129$, and modifying slightly the sets involving C_5.

F Experiments with Extended Dob Systems

In table 4 we have run some experiments on the extended Dob System, without the random polynomial q_R (see remark 2). We have chosen to fix $k = 6$, and vary $t = 3, 6, 10$. As noted in section 6.4, all the systems has $2d$ degree fall polynomials at degree 3. Furthermore, additional degree fall polynomials will be found at the first step of degree 4. This can be observed under "Step Degrees", where the initial degrees are $2, 3, 3, 4, 4(\ldots)$. The exceptions are when $t = 3$, where a solution is found already at degree 3. Despite the low first fall degree, the solving degree seems to grow with t.

Table 4: Step and Solving Degrees for Extended Dob Systems

d	t (Q_+)	k (ip)	D_{solv}	Step Degrees
55	3	6	3	2,3,3,3,3
67	3	6	3	2,3,3,3,3
55	6	6	4	2,3,3,4,4,3,3
61	6	6	4	2,3,3,4,4,3,4
65	6	6	4	2,3,3,4,4,4
67	6	6	4	2,3,3,4,4,4
55	10	6	5	2,3,3,4,4,5
67	10	6	≥ 5	2,3,3,4,4,5...

G Experiments with the Dob Encryption System

There is substantial freedom in the choice of parameters, d, n, t, k, that can be associated to a Dob encryption system (with some fixed variables). In light of this, we find that elaborate experimentation is necessary in order to gain confidence in the degree fall estimates presented in Equations (20) – (25) of section 4.3. We hope to make a stride towards such confidence by presenting various experiments in Tables 5 – 8 (in addition to what was presented in Table 2 of section 5). The setup of the tables is as described in section 5.3. An entry where none of our formulas predict a positive value is marked with a "-" under "N (predicted)". We also do need register the experimentally found number of degree fall polynomials, "N (Magma)", if the registered first fall degree is the same as $D_{reg}(d, n)$. Experiments that ran out of memory has been marked with an inequality in D_{solv}, and "..." under "Step Degrees". The last number in "Step Degrees" marks the step that ran out of memory. It is worth noting that in all the experiments we have run, the number of degree fall polynomials we predict, "N (predicted)", matches exactly the number of registered first fall polynomials. Furthermore, in the cases where "N (predicted)" is marked with "-", the experimental first fall degree is indeed ≥ 6.

Table 5: Dob encryption scheme for various parameters, $D_{ff} = 3, 4$.

d	n	t	k	D_{ff}	N	N	D_{solv}	Step
		(Q_+)	(ip)		(predicted)	(Magma)	$(D_{reg}(d,n))$	Degrees
53	53	0	5	4	$N_4^{(1,0)}: 45$	3:45	5 (9)	2,3,4,4,5,4
49	49	0	0	3	$N_3^{(0,0)}: 98$	2:98	3 (9)	2,3,3
29	29	3	0	4	$N_4^{(0,0)}: 528$	3:528	4 (6)	2,3,4,4
29	29	4	0	4	$N_4^{(0,0)}: 155$	3:155	4 (6)	2,3,4,4,4
33	32	4	0	4	$N_4^{(0,0)}: 237$	3:237	4 (6)	2,3,4,4,4
29	29	0	2	3	$N_3^{(0,0)}: 31$	2:31	3 (6)	2,3,3,3,3,3
29	29	0	3	4	$N_4^{(0,0)}: 739$	3:739	4 (6)	2,3,4,4
31	31	0	3	4	$N_4^{(0,0)}: 822$	3:822	4 (6)	2,3,4,4
31	30	0	3	4	$N_4^{(0,0)}: 842$	3:842	4 (6)	2,3,4,4
31	31	0	4	4	$N_4^{(1,0)}: 139$	3:139	4 (6)	2,3,4,4,4
29	29	0	4	4	$N_4^{(1,0)}: 131$	3:131	4 (6)	2,3,4,4,4
33	33	0	4	4	$N_4^{(1,0)}: 147$	3:147	4 (7)	2,3,4,4,4
35	35	0	4	4	$N_4^{(1,0)}: 155$	3:155	4 (7)	2,3,4,4,4
29	25	0	4	4	$N_4^{(0,0)}: 250$	3:250	4 (5)	2,3,4,4,4
31	31	0	5	4	$N_4^{(1,0)}: 45$	3:45	≥ 5 (6)	2,3,4,4,5...
29	29	1	4	4	$N_4^{(1,0)}: 25$	3:25	5 (6)	2,3,4,4,5,4
35	26	0	6	4	$N_4^{(1,0)}: 5$	3:5	5 (5)	2,3,4,4,5
59	29	0	7	4	$N_4^{(1,0)}: 21$	3:21	5 (5)	2,3,4,4,5
31	29	2	2	4	$N_4^{(0,0)}: 702$	3:702	4 (6)	2,3,4,4,3
37	24	0	5	4	$N_4^{(0,0)}: 204$	3:204	4 (5)	2,3,4,4
37	25	0	5	4	$N_4^{(1,0)}: 165$	3:165	4 (5)	2,3,4,4,3
37	24	1	4	4	$N_4^{(0,0)}: 508$	3:508	4 (5)	2,3,4,4
37	25	1	4	4	$N_4^{(0,0)}: 434$	3:434	4 (5)	2,3,4,4
79	33	0	6	4	$N_4^{(0,0)}: 500$	3:500	4 (5)	2,3,4,4
83	34	0	6	4	$N_4^{(0,0)}: 561$	3:561	4 (5)	2,3,4,4

Table 6: Dob encryption scheme for various parameters, $D_{ff} \geq 5$.

d	n	t	k	D_{ff}	N	N	D_{solv}	Step
		(Q_+)	(ip)		(predicted)	(Magma)	$(D_{reg}(d,n))$	Degrees
53	53	0	8	≥ 6	-	-	≥ 6 (9)	2,3,4,5,6. . .
37	37	5	0	5	$N_5^{(0,0)}$: 12617	3:397, 4:12220	5 (7)	2,3,4,5,4,3
35	30	0	8	5	$N_5^{(1,1)}$: 1568	4:1568	5 (6)	2,3,4,5,5
35	34	0	8	5	$N_5^{(3)}$: 224	4:224	6 (7)	2,3,4,5,5,6
41	31	0	9	5	$N_5^{(3)}$: 218	4:218	5 (6)	2,3,4,5,5,5
35	35	1	6	5	$N_5^{(2)}$: 2714	4:2714	5 (7)	2,3,4,5,5,5
31	29	2	4	5	$N_5^{(1)}$: 5869	4:5869	5 (6)	2,3,4,5,5
31	31	0	6	5	$N_5^{(2)}$: 4407	4:4407	5 (6)	2,3,4,5,5
33	32	0	6	5	$N_5^{(2)}$: 4984	4:4984	5 (6)	2,3,4,5,5
33	32	0	7	5	$N_5^{(2)}$: 2596	4:2596	5 (6)	2,3,4,5,5
33	32	0	8	5	$N_5^{(3)}$: 244	4:244	6 (6)	2,3,4,5,5,5,6
33	32	0	9	6	-	-	6 (6)	2,3,4,5,6,6
33	31	0	8	5	$N_5^{(2,1)}$: 314	4:314	5 (6)	2,3,4,5,5,5
31	28	0	8	5	$N_5^{(1,1)}$: 1172	4:1172	5 (6)	2,3,4,5,5
33	28	0	9	6	-	-	6 (6)	2,3,4,5,6
35	28	0	6	5	$N_5^{(1,1)}$: 5964	3:49, 4:5915	5 (6)	2,3,4,5,4,5
37	35	6	0	5	$N_5^{(0,0)}$: 8048	4:8048	5 (7)	2,3,4,5,5
37	37	6	0	5	$N_5^{(0,0)}$: 6364	4:6364	5 (7)	2,3,4,5,5,3
39	37	6	0	5	$N_5^{(0,0)}$: 9030	4:9030	5 (7)	2,3,4,5,5,3
37	35	7	0	5	$N_5^{(0,0)}$: 2969	4:2969	5 (7)	2,3,4,5,5,5
37	35	8	0	6	-	4:4817 5:96104	6 (7)	2,3,4,5,6,5
39	38	6	0	5	$N_5^{(0,0)}$: 8136	4:8136	5 (7)	2,3,4,5,5,3
39	38	5	0	5	$N_5^{(0,0)}$: 14940	3:429, 4:14511	5 (7)	2,3,4,5,4,3
37	36	7	0	5	$N_5^{(0,0)}$: 1644	4:1644	5 (7)	2,3,4,5,5,5,3
39	38	2	4	5	$N_5^{(0,0)}$: 5458	4:5458	5 (7)	2,3,4,5,5
39	38	4	2	5	$N_5^{(0,0)}$: 16112	4:16112	5 (7)	2,3,4,5,5
37	36	2	4	5	$N_5^{(0,0)}$: 5578	4:5578	5 (7)	2,3,4,5,5
39	39	0	6	5	$N_5^{(1,1)}$: 6255	4:6255	5 (7)	2,3,4,5,5
37	37	0	6	5	$N_5^{(1,1)}$: 5769	4:5769	5 (7)	2,3,4,5,5
35	35	0	6	5	$N_5^{(1,1)}$: 5299	4:5299	5 (7)	2,3,4,5,5
33	33	0	6	5	$N_5^{(1,1)}$: 4845	4:4845	5 (7)	2,3,4,5,5
37	36	0	6	5	$N_5^{(1,1)}$: 5940	4:5940	5 (7)	2,3,4,5,5
39	35	0	6	5	$N_5^{(1,1)}$: 6883	4:6883	5 (7)	2,3,4,5,5
37	35	0	6	5	$N_5^{(1,1)}$: 6091	4:6091	5 (7)	2,3,4,5,5
37	34	0	6	5	$N_5^{(1,1)}$: 6222	4:6222	5 (7)	2,3,4,5,5
35	34	0	6	5	$N_5^{(1,1)}$: 5454	4:5454	5 (7)	2,3,4,5,5
35	33	0	6	5	$N_5^{(1,1)}$: 5589	4:5589	5 (7)	2,3,4,5,5
33	31	0	6	5	$N_5^{(1,1)}$: 5103	4:5103	5 (6)	2,3,4,5,5

Table 7: Dob encryption scheme for various parameters, $D_{ff} \geq 5$.

d	n	t	k	D_{ff}	N	N	D_{solv}	Step
		(Q_+)	(ip)		(predicted)	(Magma)	$(D_{reg}(d,n))$	Degrees
37	37	1	6	5	$N_5^{(1,1)}: 2816$	4:2816	5 (7)	2,3,4,5,5,5
39	38	1	6	5	$N_5^{(1,1)}: 3304$	4:3304	5 (7)	2,3,4,5,5,5
39	39	1	6	5	$N_5^{(1,1)}: 2910$	4:2910	5 (7)	2,3,4,5,5,5
37	36	1	6	5	$N_5^{(1,1)}: 3182$	4:3182	5 (7)	2,3,4,5,5,5
43	37	1	6	5	$N_5^{(1,1)}: 5384$	4:5384	5 (7)	2,3,4,5,5
43	39	1	6	5	$N_5^{(1,1)}: 4718$	4:4718	5 (7)	2,3,4,5,5,5
41	39	1	6	5	$N_5^{(1,1)}: 3814$	4:3814	5 (7)	2,3,4,5,5,5
41	38	1	6	5	$N_5^{(1,1)}: 4184$	4:4184	5 (7)	2,3,4,5,5,5
37	36	2	6	5	$N_5^{(1,1)}: 400$	4:400	≥ 6 (7)	2,3,4,5,5,6…
37	30	2	6	5	$N_5^{(1,1)}: 3100$	4:3100	5 (6)	2,3,4,5,5
37	30	3	6	5	$N_5^{(1,1)}: 1350$	4:1350	5 (6)	2,3,4,5,5
37	31	2	6	5	$N_5^{(1,1)}: 2730$	4:2730	5 (6)	2,3,4,5,5
37	32	2	6	5	$N_5^{(1,1)}: 2328$	4:2328	5 (6)	2,3,4,5,5
41	35	2	6	5	$N_5^{(1,1)}: 2578$	4:2578	5 (6)	2,3,4,5,5,5
41	34	2	6	5	$N_5^{(1,1)}: 3028$	4:3028	5 (6)	2,3,4,5,5
41	34	3	6	5	$N_5^{(1,1)}: 630$	4:630	≥ 6 (6)	2,3,4,5,5,6…
41	35	3	6	≥ 6	-	-	≥ 6 (6)	2,3,4,5,6…
47	33	3	6	5	$N_5^{(1,1)}: 3603$	4:3603	5 (6)	2,3,4,5,5
37	29	4	6	5	$N_5^{(1,1)}: 231$	4:231	5 (6)	2,3,4,5,5,5
43	32	5	6	6	-	-	6 (6)	2,3,4,5,6
47	32	5	6	5	$N_5^{(1,1)}: 34$	4:34	6 (6)	2,3,4,5,5,6
61	36	6	6	6	-	-	6 (6)	2,3,4,5,6
75	39	6	6	5	$N_5^{(0,0)}: 4674$	4:4674	5 (6)	2,3,4,5,5
33	31	0	7	5	$N_5^{(1,1)}: 3009$	4:3009	5 (6)	2,3,4,5,5
33	30	0	7	5	$N_5^{(1,1)}: 3387$	4:3387	5 (6)	2,3,4,5,5
37	33	0	7	5	$N_5^{(1,1)}: 3900$	4:3900	5 (6)	2,3,4,5,5
37	34	0	7	5	$N_5^{(1,1)}: 3473$	4:3473	5 (7)	2,3,4,5,5
37	35	0	7	5	$N_5^{(1,1)}: 3011$	4:3011	5 (7)	2,3,4,5,5,5
35	30	0	7	5	$N_5^{(1,1)}: 4179$	4:4179	5 (6)	2,3,4,5,5
35	33	0	7	5	$N_5^{(1,1)}: 3024$	4:3024	5 (7)	2,3,4,5,5
39	35	0	7	5	$N_5^{(1,1)}: 3943$	4:3943	5 (7)	2,3,4,5,5
39	36	0	7	5	$N_5^{(1,1)}: 3474$	4:3474	5 (7)	2,3,4,5,5,5
39	37	0	7	5	$N_5^{(1,1)}: 2970$	4:2970	5 (7)	2,3,4,5,5,5
39	35	0	8	5	$N_5^{(2,1)}: 394$	4:394	≥ 6 (7)	2,3,4,5,5,5,6…
41	34	0	8	5	$N_5^{(1,1)}: 1408$	4:1408	5 (6)	2,3,4,5,5,5
49	36	0	8	5	$N_5^{(1,1)}: 4060$	4:4060	5 (6)	2,3,4,5,5
57	38	0	8	5	$N_5^{(1,1)}: 7000$	4:7000	5 (6)	2,3,4,5,5
55	37	0	8	5	$N_5^{(1,1)}: 6638$	4:6638	5 (6)	2,3,4,5,5
53	37	0	8	5	$N_5^{(1,1)}: 5494$	4:5494	5 (6)	2,3,4,5,5

Table 8: Dob encryption scheme for various parameters, $D_{ff} \geq 5$.

d	n	t	k	D_{ff}	N	N	D_{solv}	Step
		(Q_+)	(ip)		(predicted)	(Magma)	$(D_{reg}(d,n))$	Degrees
49	34	3	6	5	$N_5^{(1,1)}$: 3894	4:3894	5 (6)	2,3,4,5,5
51	35	3	6	5	$N_5^{(1,1)}$: 4195	4:4195	5 (6)	2,3,4,5,5
53	35	4	6	5	$N_5^{(1,1)}$: 2525	4:2525	5 (6)	2,3,4,5,5
51	35	4	6	5	$N_5^{(1,1)}$: 1669	4:1669	5 (6)	2,3,4,5,5,5
49	33	4	6	5	$N_5^{(1,1)}$: 2219	4:2219	5 (6)	2,3,4,5,5
57	36	4	6	5	$N_5^{(1,1)}$: 3564	4:3564	5 (6)	2,3,4,5,5
57	35	4	6	5	$N_5^{(1,1)}$: 4237	4:4237	5 (6)	2,3,4,5,5
65	37	6	6	5	$N_5^{(1,1)}$: 780	4:780	5 (6)	2,3,4,5,5,5
81	41	0	7	5	$N_5^{(0,0)}$: 25534	4:25534	5 (6)	2,3,4,5,5
81	42	0	7	5	$N_5^{(1,1)}$: 23613	4:23613	5 (6)	2,3,4,5,5
81	43	0	7	5	$N_5^{(1,1)}$: 23487	4:23487	5 (6)	2,3,4,5,5
83	41	0	7	5	$N_5^{(0,0)}$: 29450	4:29450	5 (6)	2,3,4,5,5
83	42	0	7	5	$N_5^{(0,0)}$: 24910	4:24910	5 (6)	2,3,4,5,5
83	43	0	7	5	$N_5^{(1,1)}$: 24643	4:24643	5 (6)	2,3,4,5,5
37	30	2	7	5	$N_5^{(1,1)}$: 1127	4:1127	5 (6)	2,3,4,5,5
41	31	3	7	5	$N_5^{(1,1)}$: 58	4:58	6 (6)	2,3,4,5,5,6
45	32	2	8	5	$N_5^{(1,1)}$: 88	4:88	6 (6)	2,3,4,5,5,6
89	43	3	5	5	$N_5^{(0,0)}$: 55986	2:424, 3:8514, 4:47048	5 (6)	2,3,4,5,3
93	44	2	6	5	$N_5^{(0,0)}$: 46246	4:46246	5 (6)	2,3,4,5,5

Paper III

On the Effect of Projection on Rank Attacks in Multivariate Cryptography

Morten Øygarden, Daniel Smith–Tone, and Javier Verbel.
To appear in *PQCrypto: International Conference on Post-Quantum Cryptography*, 2021.

On the Effect of Projection on Rank Attacks in Multivariate Cryptography

Morten Øygarden[1], Daniel Smith-Tone[2,3], and Javier Verbel[4]

[1] Simula UiB, Norway
morten.oygarden@simula.no
[2] National Institute of Standards and Technology, USA
daniel.smith@nist.gov
[3] University of Louisville, USA
[4] Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE
javier.verbel@tii.ae

Abstract. The multivariate scheme HFEv- used to be considered a promising candidate for a post-quantum signature system. First suggested in the early 2000s, a version of the scheme made it to the third round of the ongoing NIST post-quantum standardization process. In late 2020, the system suffered from an efficient rank attack due to Tao, Petzoldt, and Ding. In this paper, we inspect how this recent rank attack is affected by the projection modification. This modification was introduced to secure the signature scheme PFLASH against its predecessor's attacks. We prove upper bounds for the rank of projected HFEv- (pHFEv-) and PFLASH under the new attack, which are tight for the experiments we have performed. We conclude that projection could be a useful tool in protecting against this recent cryptanalysis.

Keywords: post-quantum cryptography, multivariate cryptography, min-rank

1 Introduction

Multivariate cryptography has received increased attention over the last years, due to its potential of providing quantum–safe public key cryptosystems. Signature schemes based on these ideas seemed particularly promising, with one finalist, Rainbow [12], and one alternate candidate, G*e*MSS [8], reaching the third and current round of the NIST post–quantum standardization process. Recently, new attacks have been presented against both of these candidates [3, 24]. The rank attack against G*e*MSS seems particularly effective, breaking all the suggested parameters for this scheme.

A similar story took place over a decade ago, when the signature scheme SFLASH was broken [14]. In the aftermath, it was discovered that this attack can be avoided by projecting the input space [13], and the amended scheme, PFLASH [9], has withstood cryptanalysis up until this point. In this article, we study the effect of projection on the new rank attack from [24], with a particular interest in the setting of HFEv- (the core of the GeMSS scheme), and PFLASH. After briefly describing the schemes and the attack, we prove that the attack also applies to PFLASH, breaking all of the proposed parameters. We then provide upper bounds for the rank in both the setting of HFEv- and PFLASH. We test the validity of these results through experiments, before concluding with a discussion on possible secure parameters and the impact these changes have on signing time.

Notation. For readability, we use the following notational conventions throughout the article. $\mathbb{F}_q^{n_1 \times n_2}$ will denote the space of matrices of size $n_1 \times n_2$ over $\mathbb{F}_q$, and matrices will be written in **bold**. Row (resp. column) entries in matrices will be written as an integer modulo n_1 (resp. n_2). For two matrices $\mathbf{A}$ and $\mathbf{B}$, we let $\mathbf{A}|\mathbf{B}$ denote their horizontal concatenation, and $\mathbf{A} \oplus \mathbf{B} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$ is the direct sum. Maps over $\mathbb{F}_q$ will be written using capital letters, while maps over extension fields, $\mathbb{F}_{q^n}$, will be written with lowercase letters.

2 Big Field Cryptosystems

We start by describing a general big field cryptosystem, with the vinegar, minus and projection modifiers. Let q be the power of a prime, n a positive integer, and fix an isomorphism $\phi : \mathbb{F}_q^n \to \mathbb{F}_{q^n}$. Define $\psi = \phi \times \mathrm{id}_v : \mathbb{F}_q^{n+v} \to \mathbb{F}_{q^n} \times \mathbb{F}_q^v$, where $\psi = \phi$ if $v = 0$. A quadratic central map is chosen of the form $F = \phi^{-1} \circ f \circ \psi : \mathbb{F}_q^{n+v} \to \mathbb{F}_q^n$, where f is specifically chosen in a way such that it is efficient to find preimages of it. Choose a linear map $U = (S \oplus \mathrm{id}_v) \circ U' : \mathbb{F}_q^{n+v-p} \to \mathbb{F}_q^{n+v}$, where both $S : \mathbb{F}_q^{n-p} \to \mathbb{F}_q^n$ and $U' : \mathbb{F}_q^{n+v-p} \to \mathbb{F}_q^{n+v-p}$ are linear maps of full rank. Let $T : \mathbb{F}_q^n \to \mathbb{F}_q^{n-a}$ be a linear map of full rank. Then the public key is created as the composition $P = T \circ F \circ U : \mathbb{F}_q^{n+v-p} \to \mathbb{F}_q^{n-a}$. Figure 1 gives an overview of the construction. We will say that the scheme uses the minus modification if $a > 0$, the vinegar modification if $v > 0$, and the projection modification if $p > 0$.

$$\begin{tikzcd}
 & \mathbb{F}_{q^n} \times \mathbb{F}_q^v \arrow[r, "f"] & \mathbb{F}_{q^n} \arrow[d, "\phi^{-1}"] & \\
\mathbb{F}_q^{n+v-p} \arrow[r, "U"] & \mathbb{F}_q^{n+v} \arrow[u, "\psi"] \arrow[r, "F"] & \mathbb{F}_q^n \arrow[r, "T"] & \mathbb{F}_q^{n-a}
\end{tikzcd}$$

Fig. 1: Diagram of a general big field scheme with minus, vinegar and projection modifiers.

HFEv-. The signature scheme HFEv- is based on the HFE central map proposed in [21]. It inspired two submissions to the NIST post–quantum standardization process: GeMSS [8] and Gui [11], where the former advanced to the third round as an alternate candidate. Fix a positive integer D, and denote the vinegar variables by $\mathbf{x_v} = (x_{n+1}, \ldots, x_{n+v})$. The central map is constructed from a polynomial f of the form

$$f_{hfe}(X, \mathbf{x_v}) = \sum_{\substack{i,j\in\mathbb{N} \\ q^i+q^j\leq D}} \alpha_{i,j} X^{q^i+q^j} + \sum_{\substack{i\in\mathbb{N} \\ q^i\leq D}} \beta_i(\mathbf{x_v}) X^{q^i} + \gamma(\mathbf{x_v}),$$

where $\alpha_{i,j} \in \mathbb{F}_{q^n}$, the β_i's are linear maps $\mathbb{F}_q^v \to \mathbb{F}_{q^n}$, and γ is a quadratic map $\mathbb{F}_q^v \to \mathbb{F}_{q^n}$. The rank attack introduced in [24], which we will recall in the next section, breaks GeMSS with the proposed parameters for the third round of the NIST Standardization process [8].

PFLASH. The signature scheme PFLASH [13, 9] is based on the C^* cryptosystem [18], and it uses the projection and minus modifiers. Since there are no vinegar modifiers, we will simply write $U = S$ for the input map. For an integer $0 < \theta < n - 1$, the central map is based on the monomial $f_{C^*} = X^{1+q^\theta}$, which is a bijection when $\gcd(q^\theta + 1, q^n - 1) = 1$. In this case, f_{C^*} can be inverted by exponentiation. With the secret key, one can also compute bilinear relations of inputs and outputs of the central map [20], which can be used to find preimages of the public key, as used in [7]. We also refer to [6] for more information on the security of PFLASH.

3 New Rank Attack

In this section, we briefly recall the new rank attack against HFEv-, that was introduced in [24]. More information about the underlying constructions can also be found in [2]. For simplicity, we consider $\mathbb{F}_q$ to be a field of odd characteristic in this section, but note that the results generalize to even fields as well (see e.g., Section 6.3 in [2]). In particular, the results in later sections will also hold in the binary case. Recall that $\mathbf{x_v} = (x_{n+1}, \ldots, x_{n+v})$ denotes the vinegar variables, and that all matrix entries are counted modulo n. For $X \in \mathbb{F}_{q^n}[X]$ we will write $\underline{X} = (X, X^q, \ldots, X^{q^{n-1}})$.

Proposition 1 ([24]). *Let f_{hfe} be an HFEv- polynomial over $\mathbb{F}_{q^n}$. Then,*

$$f_{hfe}(\underline{X}, \mathbf{x_v}) = (\underline{X}, \mathbf{x_v}) \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{D} \end{bmatrix} (\underline{X}, \mathbf{x_v})^\top,$$

where $\mathbf{A} = [\alpha_{i,j}] \in \mathbb{F}_{q^n}^{n\times n}$, $\mathbf{B} = [\beta_{i,j}] \in \mathbb{F}_{q^n}^{n\times v}$ and $\mathbf{D} = [\delta_{i,j}] \in \mathbb{F}_{q^n}^{v\times v}$. Also, for each $0 \leq k < n$

$$(f_{hfe}(\underline{X}, \mathbf{x_v}))^{q^k} = (\underline{X}, \mathbf{x_v}) \mathbf{F}^{*k} (\underline{X}, \mathbf{x_v})^\top,$$

where $\mathbf{F}^{*k} \in \mathbb{F}_{q^n}^{(n+v)\times(n+v)}$ *and its* (i,j)*-coordinate is given by*

$$\begin{cases} \alpha_{i-k,j-k}^{q^k} & \textit{if } 0 \leq i,j < n-1 \\ \beta_{i-n,j-k}^{q^k} & \textit{if } n \leq i < n+v \textit{ and } 0 \leq j < n \\ \beta_{i-k,j-n}^{q^k} & \textit{if } n \leq j < n+v \textit{ and } 0 \leq i < n \\ \delta_{i-n,j-n}^{q^k} & \textit{otherwise.} \end{cases}$$

Let $\mathbf{M} \in \mathbb{F}_{q^n}^{n\times n}$ be an invertible matrix associated with a vector basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ (see Proposition 2 [2]), and let us consider an HFEv- public key $(P_1, \ldots, P_{n-a}) = T \circ F \circ U$. If $\mathbf{P}_i$ is the symmetric matrix such that $P_i(\mathbf{x}) = \mathbf{x}\mathbf{P}_i\mathbf{x}^\top$, then we have

$$(\mathbf{x}\mathbf{P}_1\mathbf{x}^\top, \ldots, \mathbf{x}\mathbf{P}_{n-a}\mathbf{x}^\top) = (\mathbf{x}\mathbf{W}\mathbf{F}^{*0}\mathbf{W}^\top\mathbf{x}^\top, \ldots, \mathbf{x}\mathbf{W}\mathbf{F}^{*(n-1)}\mathbf{W}^\top\mathbf{x}^\top)\mathbf{M}^{-1}\mathbf{T},$$

where $\mathbf{W} = \mathbf{U}\tilde{\mathbf{M}}$ and $\tilde{\mathbf{M}} = \mathbf{M} \oplus \mathbf{I}_v$. By symmetry we have the following matrix equation

$$(\mathbf{P}_1|\cdots|\mathbf{P}_{n-a}) = \left(\mathbf{W}\mathbf{F}^{*0}\mathbf{W}^\top|\cdots|\mathbf{W}\mathbf{F}^{*(n-1)}\mathbf{W}^\top\right)\left(\mathbf{M}^{-1}\mathbf{T} \otimes \mathbf{I}_{n+v}\right). \quad (1)$$

For any vector $\mathbf{u} \in \mathbb{F}_{q^n}^{n+v}$, we define

$$\mathbf{u}\mathbf{F}^* := \begin{bmatrix} \mathbf{u}\mathbf{F}^{*0} \\ \vdots \\ \mathbf{u}\mathbf{F}^{*(n-1)} \end{bmatrix} \in \mathbb{F}_{q^n}^{n\times(n+v)}, \text{ and } \mathbf{u}\mathbf{P}^* := \begin{bmatrix} \mathbf{u}\mathbf{P}_1 \\ \vdots \\ \mathbf{u}\mathbf{P}_{n-a} \end{bmatrix} \in \mathbb{F}_{q^n}^{(n-a)\times(n+v)}.$$

Notice that if the central map of the given public key $(P_1, \ldots, P_{n-a})$ has univariate degree at most D, then

$$\operatorname{rank}(\mathbf{e}\mathbf{F}^*) \leq \lceil \log_q(D) \rceil,$$

where $\mathbf{e} \in \mathbb{F}_{q^n}^{n+v}$ is any vector of weight one. Since $p = 0$, $\mathbf{W}$ is nonsingular, and by equation (1), we have

$$\operatorname{rank}(\mathbf{u}\mathbf{P}^*) \leq \lceil \log_q(D) \rceil,$$

where $\mathbf{u} = \mathbf{e}\mathbf{W}^{-1}$. In [24] the authors find such a vector $\mathbf{u}$ by solving an instance of the MinRank problem with $n+v$ matrices in $\mathbb{F}_q^{(n-a)\times(n+v)}$ and target rank $\lceil \log_q(D) \rceil$. Furthermore, [24] shows how this vector $\mathbf{u}$ can be used to recover an equivalent key for $(P_1, \ldots, P_{n-a})$. That is, to find linear maps T', U' and a HFEv- central map F' of degree at most D, such that

$$(P_1, \ldots, P_{n-a}) = T' \circ F' \circ U'.$$

The complexity of this attack is dominated by performing the MinRank step to recover $\mathbf{u}$. This computation in turn relies heavily on the rank of $\mathbf{u}\mathbf{P}^*$, which will be our primary focus in the next sections.

4 Effect of Projection on the New Rank Attack

We now turn our attention to how the projection modification affects the recently introduced rank attack that was described in the previous section. The first thing to notice is that the invertibility of the input transformation S is required to justify the rank bound. Thus, one may wonder whether the projection modifier masks the rank property just as it was shown to protect PFLASH from the attack on SFLASH, see [14, 22].

Despite the similarities between the HFE and C^* central maps, we find that there are subtle differences in how projection affects the different schemes. As a result, we consider the two settings separately in the following subsections.

4.1 Projection and the HFE Central Map

We adopt an approach dual to that of [25], where removing equations was shown to be equivalent to increasing the degree of the central map. Specifically, we prove that projection is equivalent to increasing the degree of the central map. Thus pHFEv- with degree bound D and projection p is an instance of HFEv- with degree bound q^pD.

For any $\mathbb{F}_q$-subspace K of $\mathbb{F}_{q^n}$ there exists a linear polynomial of the form

$$\min_K(X) = \prod_{\alpha \in K}(X-\alpha),$$

having K as its kernel. This polynomial is also known as the minimal polynomial of K, see [10]. We start by showing the following result.

Lemma 1. *There is a bijective correspondence between k-dimensional subspaces of $\mathbb{F}_{q^n}$ and $(n-k)$-dimensional subspaces of $\mathbb{F}_{q^n}$ given by*

$$W \mapsto Im(min_W(X)).$$

Proof. Let $\mathcal{V}_k$ be the collection of k-dimensional subspaces of $\mathbb{F}_{q^n}$. Define the map $\psi_k : \mathcal{V}_k \to \mathcal{V}_{n-k}$ by $\psi(W) = \mathrm{Im}(\min_W(X)) = W'$. Note that since $\min_W(X)$ has kernel of dimension k, and is $\mathbb{F}_q$–linear, the space W' will have dimension $n-k$, and ψ_k is thus well–defined. Moreover, $\min_{W'}(\min_W(X)) = 0$, and by degree considerations we have, more exactly, $\min_{W'}(\min_W(X)) = X^{q^n} - X$.

Suppose that

$$\min_W(X) = \sum_{i=0}^{k}\alpha_i X^{q^i} \quad \text{and} \quad \min_{W'}(X) = \sum_{i=0}^{n-k}\beta_i X^{q^i}.$$

Then we observe that the composition is

$$\begin{aligned}\min_{W'} \circ \min_{W}(X) &= \sum_{i=0}^{n-k}\sum_{j=0}^{k} \alpha_j^{q^i}\beta_i X^{q^{i+j}} \\ &= \sum_{r=0}^{n}\left(\sum_{\substack{0\leq i\leq n-k \\ 0\leq j\leq k,\ j+i=r}} \alpha_j^{q^i}\beta_i\right) X^{q^r} = X^{q^n} - X.\end{aligned} \tag{2}$$

Recalling that $\alpha_k = \beta_{n-k} = 1$, we find that this relation produces a system of n bilinear equations in the $k-1$ coefficients α_j and the $n-k-1$ coefficients β_i. Now fix a space W' in the image of ψ_k, and let β_i be the fixed, associated constants of $\min_{W'}(X)$. Ordering the equations from $r = n-1$ to $r = 0$, we may sequentially solve for α_j. In fact, other than the Frobenius powers applied to the α_j values, the system is triangular, and hence uniquely solvable (see Appendix A for a small toy example of this). Thus, ψ_k is injective. Since the action of taking the orthogonal complement twice yields the original space, the number of subspaces of dimension k and of dimension $n-k$ are equal. It follows that ψ_k is also surjective, and hence a bijection.

Now let S be a linear map[5] $\mathbb{F}_q^n \to \mathbb{F}_q^n$ with kernel of dimension p. Using Lemma 1, we choose π to be the unique minimal polynomial such that $\phi^{-1}(Im(\pi)) = Im(S)$. Note that π has degree q^p. Then we have an exact sequence

$$\mathbb{F}_q^n \xrightarrow{\phi^{-1}\circ\pi\circ\phi} Im(S) \to 0.$$

Since $\mathbb{F}_q$-vector spaces are free (and therefore projective) $\mathbb{F}_q$-modules, there exists an S' such that the following diagram commutes:

$$\begin{tikzcd}
 & \mathbb{F}_q^n \arrow[dl, dashed, "S'"'] \arrow[d, "S"] & \\
\mathbb{F}_q^n \arrow[r, "\phi^{-1}(\pi(\phi))"] & Im(S) \arrow[r] & 0
\end{tikzcd}$$

If S' is singular, then its rank is at least $n-p$, and its kernel is then contained in the kernel of S. If necessary, we can replace S' with a nonsingular linear map by redefining its value on $ker(S')$ to map into $ker(\phi^{-1}\circ\pi\circ\phi)$. We may then without loss of generality choose S' to be of full rank. Thus, we obtain the matrix equation $\mathbf{S} = \mathbf{S'Q}$, where $\mathbf{xQ} = \phi^{-1}\circ\pi\circ\phi(\mathbf{x})$.

[5] This is a slight abuse of notation from the S defined in Section 2, which had $\mathbb{F}_q^{n-p}$ as its domain. This is easily remedied by composing with a projection along the $n-p$ first coordinates.

We may now apply this result in the case of an HFEv- scheme. In this case, we have the public key

$$\left[\mathbf{P}_1|\cdots|\mathbf{P}_{n-a}\right] = \left[\mathbf{U}\widetilde{\mathbf{M}}\mathbf{F}^{*0}\widetilde{\mathbf{M}}^\top\mathbf{U}^\top|\cdots|\mathbf{U}\widetilde{\mathbf{M}}\mathbf{F}^{*(n-1)}\widetilde{\mathbf{M}}^\top\mathbf{U}^\top\right]\left(\mathbf{M}^{-1}\mathbf{T}\otimes\mathbf{I}_n\right),$$

where $\widetilde{\mathbf{M}} = \mathbf{M}\oplus\mathbf{I}_v$ and $\mathbf{U} = \mathbf{U}'(\mathbf{S}\oplus\mathbf{I}_v)^6$. We observe that

$$\begin{aligned}\widetilde{\mathbf{U}}\widetilde{\mathbf{M}}\mathbf{F}^{*i}\widetilde{\mathbf{M}}^\top\widetilde{\mathbf{U}}^\top &= \mathbf{U}'(\mathbf{S}\oplus\mathbf{I}_v)\widetilde{\mathbf{M}}\mathbf{F}^{*i}\widetilde{\mathbf{M}}^\top(\mathbf{S}^\top\oplus\mathbf{I}_v)\mathbf{U}'^\top\\ &= \mathbf{U}'(\mathbf{S}'\mathbf{Q}\oplus\mathbf{I}_v)\widetilde{\mathbf{M}}\mathbf{F}^{*i}\widetilde{\mathbf{M}}^\top(\mathbf{Q}^\top\mathbf{S}'^\top\oplus\mathbf{I}_v)\mathbf{U}'^\top\\ &= \mathbf{U}'(\mathbf{S}'\mathbf{Q}\mathbf{M}\oplus\mathbf{I}_v)\mathbf{F}^{*i}(\mathbf{M}^\top\mathbf{Q}^\top\mathbf{S}'^\top\oplus\mathbf{I}_v)\mathbf{U}'^\top\end{aligned}$$

We may further rewrite the last expression to obtain

$$\mathbf{U}'(\mathbf{S}'\mathbf{M}\oplus\mathbf{I}_v)(\mathbf{M}^{-1}\mathbf{Q}\mathbf{M}\oplus\mathbf{I}_v)\mathbf{F}^{*i}(\mathbf{M}^\top\mathbf{Q}^\top\mathbf{M}^{-\top}\oplus\mathbf{I}_v)(\mathbf{M}^\top\mathbf{S}'^\top\oplus\mathbf{I}_v)\mathbf{U}'^\top$$

We finally note that

$$\mathbf{X}(\mathbf{M}^{-1}\mathbf{Q}\mathbf{M}\oplus\mathbf{I}_v)\mathbf{F}^{*i}(\mathbf{M}^\top\mathbf{Q}^\top\mathbf{M}^{-\top}\oplus\mathbf{I}_v)\mathbf{X}^\top = \mathbf{X}\mathbf{G}^{*i}\mathbf{X}^\top,$$

where $\mathbf{X} = \left[X\ X^q\ \cdots\ X^{q^{n-1}}\ x_1\cdots x_v\right]$ and where

$$G(X, x_1, \ldots, x_v) = F(\pi(X), x_1, \ldots, x_v).$$

Thus the public key can also be expressed as

$$\left[\mathbf{P}_1|\cdots|\mathbf{P}_{n-a}\right] = \left[\mathbf{U}''\widetilde{\mathbf{M}}\mathbf{G}^{*0}\widetilde{\mathbf{M}}^\top\mathbf{U}''^\top|\cdots|\mathbf{U}''\widetilde{\mathbf{M}}\mathbf{G}^{*(n-1)}\widetilde{\mathbf{M}}^\top\mathbf{U}''^\top\right]\left(\mathbf{M}^{-1}\mathbf{T}\otimes\mathbf{I}_n\right),$$

where $\mathbf{U}''$ is the nonsingular map $\mathbf{U}'(\mathbf{S}'\oplus\mathbf{I}_v)$. Thus, the pHFEv-$(n, D, a, v, p)$ public key is also an HFEv-(n, q^pD, a, v) public key.

This allows us to follow the same reasoning used in the attack of HFEv- with degree $D = q^{p+d}$, and we have proved the following upper bound.

Proposition 2. *Let $(\mathbf{P}_1, \ldots, \mathbf{P}_{n-a})$ be the symmetric matrices of the public key of an instance of pHFEv-(n, D, a, v, p), where p is the projection corank. Then there is a non–zero tuple $\mathbf{u}\in\mathbb{F}_{q^n}^{n-p}$ such that $\mathbf{u}\mathbf{P}^*$ has rank at most $p+d$, where $d = \lceil\log_q D\rceil$.*

We will test the tightness of this upper bound in Section 5.

4.2 Projection and the C^* Central Map

Define the symmetric matrix $\mathbf{F}^{*i}_{C^*}$, associated with $f^{q^i}_{C^*}$, in a manner similar to Proposition 1. Describing $\mathbf{F}^{*i}_{C^*}$ is simpler than what was done in Proposition 1, seeing that it is 1 at the entries $(i, \theta+i)$ and $(\theta+i, i)$, and 0 elsewhere (recall that

[6] Following our slight abuse of notation when compared with Section 2: U' will now be an invertible linear map $\mathbb{F}_q^{n+v}\rightarrow\mathbb{F}_q^{n+v}$

entries are counted modulo n). While we may apply the theory from Section 4.1, the problem is that we no longer have a bound D on the non–zero part of $\mathbf{F}_{C^*}^{*0}$. Following the same reasoning as before would have yielded an upper bound of $2+2p$ for the rank, but it is possible to do better.

We define $\mathbf{v} = (v_0, \ldots, v_{n-1}) = \mathbf{uSM} \in \mathbb{F}_{q^n}^n$, and examine what rank the matrix $\mathbf{vF}_{C^*}^*$ can take. Note that the entry v_i, for $i \in \mathbb{Z}_n$, will contribute to the two entries in positions

$$e_1(i) = (i, i+\theta) \qquad \text{and} \qquad e_2(i) = (i-\theta, i-\theta)\,, \tag{3}$$

in the matrix $\mathbf{vF}_{C^*}^*$. Fix an integer i_0, and consider the pair v_{i_0} and $v_{i_0+\theta}$. They will now contribute to four entries in $\mathbf{vF}_{C^*}^*$, but two of them, $e_1(i_0) = (i_0, i_0+\theta)$ and $e_2(i_0+\theta) = (i_0, i_0)$, appear in the same row. It follows that the pair v_{i_0} and $v_{i_0+\theta}$ can only make a contribution of at most three to the rank of $\mathbf{vF}^*$. This is the key observation for the following result.

Lemma 2. *Let $I = \{i_0, \ldots, i_{k-1}\}$ be a set of k integers in $\mathbb{Z}_n$, such that $i_{j+1} = i_j + \theta$, for $0 \leq j < k-1$. Consider the vector $\mathbf{v}_I = (v_0, \ldots, v_{n-1})$, where $v_j \in \mathbb{F}_{q^n} \setminus \{0\}$ if $j \in I$, and $v_j = 0$ otherwise. Then $\mathbf{v}_I\mathbf{F}_{C^*}^*$ has rank at most $k+1$.*

Proof. For $l = 1, 2$, let $E_l(x)$ be the $n \times n$ matrix that is 1 at entry $e_l(x)$ (as defined in (3)), and 0 elsewhere. Then we can write $\mathbf{v}_I\mathbf{F}_{C^*}^*$ as the sum

$$\mathbf{v}_I\mathbf{F}_{C^*}^* = \sum_{j=0}^{k-1} \bigl(E_1(i_j) + E_2(i_j)\bigr).$$

From the discussion prior to the lemma, we know that $E_1(i_{j_0}) + E_2(i_{j_0+1})$ has rank 1, for $0 \leq j_0 < k-1$. Hence, $\mathbf{v}_I\mathbf{F}_{C^*}^*$ can be written as the sum of $2k-(k-1)$ matrices of rank 1, which proves the upper bound.

The next step is to look at which of these vectors $\mathbf{v}_I$ we can find in the image of $\mathbf{SM}$. This leads to the following upper bound.

Proposition 3. *Let $(\mathbf{P}_1, \ldots, \mathbf{P}_{n-a})$ be the symmetric matrices of the public key of an instance of PFLASH with projection p. Then there is a non–zero tuple $\mathbf{u} \in \mathbb{F}_{q^n}^{n-p}$ such that $\mathbf{uP}^*$ has rank at most $2+p$.*

Proof. Let I be as defined in Lemma 2, and consider an associated vector $\mathbf{v}_I$, with the difference that $v_j \in \mathbb{F}_{q^n}$ if $j \in I$ (i.e., allowing 0 in these entries as well). $\mathbf{SM}$ has cokernel of dimension p, so choosing I of order $p+1$ will guarantee that there is a non–trivial way to choose the entries in $\mathbf{v}_I$ such that it lies in the image of $\mathbf{SM}$. This can seen by performing Gaussian elimination on $\mathbf{SM}$, where the entries corresponding to I are being eliminated last. If all v_j for $j \in I$ are non–zero, we are done by Lemma 2. Otherwise, suppose one of them is zero, say $v_{i_l} = 0$. Then we may split I into the two (potentially empty) sets $I_1 = \{i_0, \ldots, i_{l-1}\}$, and $I_2 = \{i_{l+1}, \ldots, i_p\}$. Upon considering the two associated vectors $\mathbf{v}_{I_1}$ and $\mathbf{v}_{I_2}$, we may write $\mathbf{v}_I\mathbf{F}_{C^*}^* = \mathbf{v}_{I_1}\mathbf{F}_{C^*}^* + \mathbf{v}_{I_2}\mathbf{F}_{C^*}^*$. Using Lemma 2 on $\mathbf{v}_{I_1}\mathbf{F}^*_{C^*}$ and $\mathbf{v}_{I_2}\mathbf{F}^*_{C^*}$, along with the fact that $|I_1| + |I_2| = p$ ensures that the rank of $\mathbf{v}_I\mathbf{F}^*_{C^*}$ sums up to at most $p + 2$.

Finally, the cases where several entries v_j, $j \in I$ are zero, are dealt with by induction on this argument.

This upper bound is tight for the experiments we have run for PFLASH; more information can be found in Section 5. For now, we note that the integer set I used in the proof of Proposition 3 is not unique, and we can even consider a more general class of sets, than what was discussed in Lemma 2. Indeed, from the entries in (3), we note that the pair v_{i_0} and $v_{i_0+2\theta}$ will in particular contribute to the entries $e_1(i_0) = (i_0, i_0+\theta)$ and $e_2(i_0+2\theta) = (i_0+\theta, i_0+\theta)$, each of which lies in the same column. Note that Lemma 2, and the proof of Proposition 3, could easily have been adopted to sets I where the consecutive indices have relative distance 2θ, as opposed to θ. Furthermore, we can use combinations of θ and 2θ for distance, as shown in the following result, which is a direct generalization of Lemma 2. The proof is identical to that of the aforementioned lemma.

Lemma 3. *Let $I = \{i_0, \ldots, i_{k-1}\}$ be a set of k integers in $\mathbb{Z}_n$, such that for $0 \leq j < k-1$, the difference $i_{j+1} - i_j$ is congruent to either θ or 2θ mod n. Consider the vector $\mathbf{v}_I = (v_0, \ldots, v_{n-1})$, where $v_j \in \mathbb{F}_{q^n} \setminus 0$ if $j \in I$, and $v_j = 0$ otherwise. Then $\mathbf{v}_I\mathbf{F}^*_{C^*}$ has rank at most $k + 1$.*

Number of Solutions for the MinRank Step. Recall that [24] suggests setting $u_0 = 1$, in order to avoid finding multiples of the same solution to the MinRank–step of the attack. Let I a set of the form described in Lemma 3. Note that any such I of order $p + 1$ could have been used to prove Proposition 3. Hence, we expect each choice of I to, in general, correspond to a unique solution u of the MinRank problem of rank $p + 2$. If $\gcd(n, \theta) = 1$, and $2(p + 1) < n$, there are $n2^p$ ways to construct I (2^p combinations of distances θ and 2θ, with n rotations).

We ran a few toy examples to test this theory, by running the MinRank–step for the parameters $q = 2$, $n = 13$, $\theta = 3$, and $p = 1, 2$ and 3. In each test we found all possible solutions $\mathbf{u}$, and inspected the corresponding $\mathbf{v} = \mathbf{uSM}$. In each test the number of solutions were indeed $n2^p$, and the $\mathbf{v}$-vectors corresponded to all the different choices for I.

Weak Choices of $\boldsymbol{n}$ and $\boldsymbol{\theta}$. In special cases, it would be possible to derive a lower upper bound than what was presented in Proposition 3. This can, for instance, happen if the set I from Lemma 3 of order $k \geq 1$ is a loop, in the sense that $i_{k-1} - i_0 \equiv \theta$ or 2θ mod n. This is possible if the following equation has a solution:

$$x\theta + y2\theta \equiv 0 \mod n, \ \ x, y \in \mathbb{Z}_{\geq 0}, \text{ and } x + y = k - 1. \tag{4}$$

Solutions for this condition, with low values of k, can be found when the least common multiple of n and θ is small, or equivalently, when $\gcd(n, \theta)$ is large.

Indeed, we can observe this effect in the last two rows of the right side of Table 1: in both tests we have $n = 14$ and $p = 4$, but they differ by $\theta = 5$ and 6. In the first case, we have $\gcd(14, 5) = 1$, and we find no solutions $\mathbf{u}$ such that $\mathbf{uP}^*$ has rank 5. In the second case we have $\gcd(14, 6) = 2$, and $x = 1$, $y = 3$ is a solution of (4), with $k = 5$. The resulting effect is that we are able to find solutions of $\mathbf{u}$ such that $\mathbf{uP}^*$ is of rank 5. We include the condition $\gcd(n, \theta) = 1$ in our other PFLASH experiments in order to exclude weak cases like these.

5 Experiments

In the previous section we proved an upper bound on the rank of $\mathbf{uP}^*$, for both pHFEv-, and PFLASH; we will now examine this bound through experiments.

All tests have been performed as follows. After creating the public key P, we construct $\mathbf{uP}^*$ with the indeterminate vector $\mathbf{u}$, where $u_0 = 1$. For rank r, we follow the minors modelling [17], by computing the $(r + 1) \times (r + 1)$ minors of $\mathbf{uP}^*$, and solving the associated polynomial system using the implementation of F_4 [15] in the Magma Computer Algebra System[7], see [4]. For efficiency, we did not always include all the minors when computing the Gröbner basis. We chose the rank r to be one less than, or equal, to the upper bound determined in Propositions 2 and 3 for pHFE- and PFLASH, respectively. Red marks that the polynomial system from the minors modelling at this rank was inconsistent, whereas blue indicates that we were able to find solutions. The results are presented in Table 1.

Table 1: Experimentally found rank of $\mathbf{uP}^*$ for various parameters of pHFE- (left) and PFLASH (right). The number X indicates that there are no $\mathbf{u}$ such that $\mathbf{uP}^*$ has rank $\leq X$. The number X means that we were able to find a solution $\mathbf{u}$ yielding $\mathbf{uP}^*$ of rank $\leq X$. See Section 4.2 for a discussion on †.

q	**n**	**a**	**p**	**D**	**Upper Bound**	**Rank of uP***
2	13	0	1	5	4	3, 4
2	13	0	2	5	5	4, 5
2	13	0	3	5	6	5
2	15	0	4	5	7	6
2	13	0	0	9	4	3, 4
2	13	4	1	9	5	4, 5
2	13	4	2	9	6	5, 6
2	17	6	1	9	5	4, 5
2	13	4	0	17	5	4, 5
2	13	4	1	17	6	5, 6
2	13	0	2	17	7	6

q	**n**	**a**	**p**	θ	**Upper Bound**	**Rank of uP***
2	21	0	1	13	3	2, 3
2	21	0	2	13	4	3, 4
4	31	0	1	7	3	2
4	13	0	3	5	5	4, 5
4	25	8	0	11	2	1, 2
4	25	8	1	11	3	2, 3
4	17	5	3	7	5	4, 5
2	15	1	4	7	6	5, 6
2	15	0	5	7	7	6
4	14	4	4	5	6	5
4	14	4	4	6	$6^\dagger$	5

[7] Any mention of commercial products does not indicate endorsement by NIST.

We note that in all our experiments, the upper bound seems to be tight. The notable exception is the last row on the right side of Table 1, where $\gcd(n, \theta) \neq 1$, as discussed in Section 4.2. The tests include cases where f_{C^*} is not a permutation, i.e., $\gcd(q^n - 1, q^\theta + 1) \neq 1$, and this does not seem to have an effect on this attack. Finally, the target r and the dimension of $\mathbf{uP}^*$ cannot be too close, in order to ensure that the solutions we find are truly a result of the extension field structure of the scheme. We have chosen to keep $(n - a) > r + 3$ in our experiments. Indeed, in an earlier experiment with pHFE- of parameters $q = 2$, $n = 13$, $a = 4$, $p = 2$ and $D = 17$, we found a unique solution to u at $r = 6$, even though our upper bound is seven here. Upon further inspection, this solution was in the subfield $\mathbb{F}_q$ (as opposed to being in $\mathbb{F}_{q^n}$ proper, which is the case for the other tests), and we have not been able to find such solutions when rerunning the case. Hence, we conclude that this was a "false positive" caused by the small parameters of the test.

6 Complexity

In this section we compute the complexity of signing for pHFEv- and PFLASH. The inversion methods are quite disparate, so, again, we separate the exposition.

6.1 pHFEv- Signing

For this subsection we consider the base field $q = 2$. This is what was used in the GeMSS submission, which is what we will use as a baseline for comparing pHFEv-. The most complex step of the inversion of an HFEv- public key lies in the application of the Berlekamp algorithm, see [1], for inverting the central map. In the case of pHFEv-, there is a tension between the complexity of inverting the degree D polynomial and the number, 2^p, of times that the polynomial must be inverted.

As shown in Section 4, an instance of pHFEv-(n, D, a, v, p) is also an instance of HFEv-$(n, 2^p D, a, v)$. Thus, we may always invert pHFEv-(n, D, a, v, p) by using the inversion procedure for HFEv-$(n, 2^p D, a, v)$. On the other hand, we may invert the instance of pHFEv- by inverting the central map of degree D, until the preimage lies in the image of the input projection. For each preimage, the probability that it lies in the image of a corank p projection is 2^{-p}. To see which is the better of the two methods, we begin by making the analysis in [8] for the complexity of inversion more tight.

As noted in [8, Theorem 1], the complexity of Berlekamp applied to a polynomial of degree D is $\mathcal{O}\left(M_{2^n}(D)(n + \log_2 D)\log_2 D\right)$, where $M_{2^n}(D)$ is the number of operations in the field $\mathbb{F}_{2^n}$ required to multiply two polynomials of degree D. The well-known formula, see [5], for this quantity

$$M_{2^n}(D) = \mathcal{O}\left(D \log_2 D \log_2 \log_2 D\right)$$

produces a complexity of

$$\mathcal{O}\left(D(\log_2 D)^2(n + \log_2 D)\log_2 \log_2 D\right).$$

The above quantity only provides the algebraic complexity of polynomial inversion over $\mathbb{F}_{2^n}$. Since each multiplication in $\mathbb{F}_{2^n}$ requires $2n^2+n$ bit operations, we have that inverting the central map has a bit complexity of

$$\mathcal{O}\left((2n^2+n)D\log_2(D)^2(n+\log_2 D)\log_2\log_2 D\right).$$

Since we are considering values of $\log_2 D$ that are far less than n, we may further simplify to obtain the approximate bit complexity

$$Cn^3 D\log_2(D)^2\log_2\log_2 D,$$

for some constant C. We note that $\log_2\log_2 D$ may be as large as three or four, for the values of D needed to secure against [24]. It is thus a nontrivial factor in this expression.

Since the complexity of inverting pHFEv-(n, D, a, v, p) is 2^p times the complexity of inverting HFEv-(n, D, a, v), it is a factor of

$$\frac{(p+\log_2 D)^2\log_2(p+\log_2 D)}{\log_2(D)^2\log_2\log_2 D}$$

faster than inverting the scheme as an instance of HFEv-$(n, 2^pD, a, v)$.

Thus, securing the parameters of G*e*MSS while maintaining the array of parameters merely requires applying the projection modifier with a sufficiently large corank p to secure the scheme from the attack of [24]. We should note that projection does have the negative effect of increasing the signature failure rate by a factor of approximately e^{2^p}, but the rate is still $\exp(2^p - 2^{a+v})$ which is negligible for any realistic parameters.

Parameters for pHFEv-. Let $d = \lceil\log_2 D\rceil$. Similar to [24], we use the support minors equations to derive a bilinear system in n_x+n_y variables, where $n_x = n+v$ and $n_y = \binom{n'}{d+p}$, and $n' = \left\lceil\frac{(n+v)(d+p+1)}{n-a}\right\rceil + d + p + 1$. Such a bilinear system is expected to be solved at degree 3. The overall complexity of solving this system is then given by $\mathcal{O}\left((n_x n_y^2 + n_x^2 n_y)^\omega\right)$, where ω is the linear algebra constant.

In Appendix B, Table 2, we consider the third round parameters of G*e*MSS, and compute the size of the projection that is needed to achieve the required security level.

6.2 PFLASH Signing

For PFLASH, we recommend using the private key to derive the linearization equations proven to exist by Patarin in [20]. With these equations the legitimate user can find a preimage of the public key in one step instead of inverting the input and output transformations and using exponentiation to invert the central map.

As shown in Section 4, the rank of $\mathbf{uP}^*$ is $p+2$. The parameters suggested in [9] had $p = 1$, which makes them vulnerable to the rank attack we have studied.

It is, once again, possible to protect against this by increasing the projection. However, the signing time will now be multiplied by a factor q^p, which favours the use of a small ground field, maybe even $q = 2$. In this setting, direct methods may also become an issue. Particularly a generalized version of the analysis presented in [19], perhaps using some of the notions from [26] should be considered. This is, however, beyond the scope of this article, and we leave it as an open question to determine if and how secure and efficient parameters for PFLASH may be chosen.

7 Conclusion

We have studied how projection affects the new rank attack from [24]. For the pHFEv- and PFLASH systems we have derived an upper bound on how the rank grows with the projection p, which in turn can be used to estimate the complexity of the attack as a whole. These bounds were furthermore observed to be tight in experiments.

While projection is a cheap modification for encryption systems, it does increase the signing time for signature schemes, typically by a factor of q for each dimension. Nevertheless, in the HFEv- setting, we note that projecting is a useful alternative to simply increasing the degree D. PFLASH can also be made secure against rank attacks by increasing p, but we believe more analysis on direct attacks are needed before we can suggest potential parameters.

References

1. E. R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24(111):pp. 713–735, 1970.
2. L. Bettale, J.-C. Faugère, and L. Perret. Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. *Designs, Codes and Cryptography*, 69(1):1–52, 2013.
3. W. Beullens. Improved Cryptanalysis of UOV and Rainbow. Cryptology ePrint Archive, Report 2020/1343, 2020. `https://eprint.iacr.org/2020/1343`.
4. W. Bosma, J. Cannon, and C. Playoust. The magma algebra system i: The user language. *J. Symb. Comput.*, 24(3–4):235–265, Oct. 1997.
5. D. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28:693–701, July 1991.
6. R. Cartor and D. Smith-Tone. An updated security analysis of PFLASH. In *International Workshop on Post-Quantum Cryptography*, pages 241–254. Springer, 2017.
7. R. Cartor and D. Smith-Tone. EFLASH: a new multivariate encryption scheme. In *International Conference on Selected Areas in Cryptography*, pages 281–299. Springer, 2018.
8. A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. GeMSS: A Great Multivariate Short Signature (Round 3 submission). Technical report, National Institute of Standards and Technology, 2020. `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

9. M.-S. Chen, B.-Y. Yang, and D. Smith-Tone. PFLASH-secure asymmetric signatures on smart cards. In *Lightweight Cryptography Workshop*, 2015.
10. T. Daniels and D. Smith-Tone. Differential properties of the HFE cryptosystem. In M. Mosca, editor, *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, volume 8772 of *Lecture Notes in Computer Science*, pages 59–75. Springer, 2014.
11. J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, and B.-Y. Yang. GUI. Technical report, National Institute of Standards and Technology, 2017. `https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions`.
12. J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, B.-Y. Yang, M. Kannwischer, and J. Patarin. Rainbow (round 3 submission). Technical report, National Institute of Standards and Technology, 2020. `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions`.
13. J. Ding, V. Dubois, B.-Y. Yang, O. C.-H. Chen, and C.-M. Cheng. Could SFLASH be repaired? In *International Colloquium on Automata, Languages, and Programming*, pages 691–701. Springer, 2008.
14. V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern. Practical Cryptanalysis of SFLASH. In A. Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2007.
15. J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.
16. F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303, 2014.
17. F. Levy-dit Vehel, J.-C. Faugère, and L. Perret. Cryptanalysis of MinRank. In *Annual International Cryptology Conference*, pages 280–296. Springer, 2008.
18. T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 419–453. Springer, 1988.
19. M. Øygarden, P. Felke, H. Raddum, and C. Cid. Cryptanalysis of the multivariate encryption scheme EFLASH. In *Cryptographers' Track at the RSA Conference*, pages 85–105. Springer, 2020.
20. J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In *Annual International Cryptology Conference*, pages 248–261. Springer, 1995.
21. J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 33–48. Springer, 1996.
22. D. Smith-Tone. Properties of the discrete differential with cryptographic applications. In N. Sendrier, editor, *PQCrypto*, volume 6061 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2010.
23. V. Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.
24. C. Tao, A. Petzoldt, and J. Ding. Improved Key Recovery of the HFEv- Signature Scheme. Cryptology ePrint Archive, Report 2020/1424, 2020. `https://eprint.iacr.org/2020/1424`.
25. J. Vates and D. Smith-Tone. Key recovery attack for all parameters of HFE-. In T. Lange and T. Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, volume 10346 of *Lecture Notes in Computer Science*, pages 272–288. Springer, 2017.

26. M. Øygarden, P. Felke, and H. Raddum. Analysis of Multivariate Encryption Schemes: Application to Dob. Cryptology ePrint Archive, Report 2020/1442, 2020. https://eprint.iacr.org/2020/1442.

A Toy Example of Composing Minimal Polynomials

We provide a small toy example of the bilinear system from the proof of Lemma 1. Consider $n = 5$ and $k = 2$. Then, by Equation (2), and recalling $\alpha_2 = \beta_3 = 1$, we have

$$\begin{aligned}\min_{W'} \circ \min_{W}(X) = {} & X^{q^5} - X \\ & = \alpha_0\beta_0 X + (\beta_0\alpha_1 + \beta_1\alpha_0^q)X^q + (\beta_0 + \beta_1\alpha_1^q + \beta_2\alpha_0^{q^2})X^{q^2} \\ & + (\beta_1 + \beta_2\alpha_1^{q^2} + \alpha_0^{q^3})X^{q^3} + (\alpha_1^{q^3} + \beta_2)X^{q^4} + X^{q^5}.\end{aligned}$$

If the β_j's are known constants, we note that α_1 is uniquely determined by the equation $\alpha_1^{q^3} + \beta_2 = 0$. Subsequently, α_0 will be uniquely determined by $\alpha_0^{q^3} + \beta_2\alpha_1^{q^2} + \beta_1 = 0$.

B G*e*MSS Minrank Complexity

In Table 2, we consider the third round parameters of G*e*MSS, and compute the size of the projection that is needed to achieve the required security level. We do this for two values of ω: $\omega_1 = 2.37$ is the best known asymptotic bound [16], and $\omega_2 = 2.81$ is the more realistic value from Strassen's algorithm [23].

Table 2: Complexity of the MinRank attack from [24] against the GeMSS parameters with projection. The value p_1 (resp. p_2) is the minimum projection needed to achieve security with ω_1 (resp. ω_2), and C_{ω_1} (resp. C_{ω_2}) denotes $\log_2$ of the resulting complexity.

Scheme	(n, v, D, a)	p_1	C_{ω_1}	p_2	C_{ω_2}
GeMSS128	(174, 12, 513, 12)	2	136	0	139
BlueGeMSS128	(175, 14, 129, 13)	4	140	1	128
RedGeMSS128	(177, 15, 17, 15)	6	131	4	128
WhiteGeMSS128	(175, 12, 513, 12)	2	136	0	139
CyanGeMSS128	(177, 13, 129, 14)	4	140	1	128
MagentaGeMSS128	(178, 15, 17, 15)	6	131	4	128
GeMSS192	(265, 20, 513, 22)	7	192	5	201
BlueGeMSS192	(265, 23, 129, 22)	9	192	7	201
RedGeMSS192	(266, 25, 17, 23)	12	192	10	205
WhiteGeMSS192	(268, 21, 513, 21)	7	192	5	201
CyanGeMSS192	(270, 22, 129, 23)	9	192	7	201
MagentaGeMSS192	(271, 24, 17, 24)	12	192	10	205
GeMSS256	(354, 33, 513, 30)	14	263	10	267
BlueGeMSS256	(358, 32, 129, 34)	16	267	11	256
RedGeMSS256	(358, 35, 17, 34)	18	258	14	256
WhiteGeMSS256	(364, 29, 513, 31)	14	263	10	263
CyanGeMSS256	(364, 32, 129, 31)	16	263	12	263
MagentaGeMSS256	(366, 33, 17, 33)	19	263	15	267

Paper IV

An Algebraic Attack on Ciphers with Low–Degree Round Functions: Application to Full MiMC

Maria Eichlseder, Lorenzo Grassi, Reinhard Lüftenegger, Morten Øygarden, Christian Rechberger, Markus Schofnegger, and Qingju Wang.
In *International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt)*, pages 477–506. Springer, 2020.

An Algebraic Attack on Ciphers with Low-Degree Round Functions: Application to Full MiMC

(Full Version)

Maria Eichlseder[1], Lorenzo Grassi[1,2], Reinhard Lüftenegger[1],
Morten Øygarden[3], Christian Rechberger[1], Markus Schofnegger[1], and
Qingju Wang[4]

[1] IAIK, Graz University of Technology (Austria)
[2] Digital Security Group, Radboud University, Nijmegen (The Netherlands)
[3] Simula UiB (Norway)
[4] SnT, University of Luxembourg (Luxembourg)
firstname.lastname@iaik.tugraz.at
lgrassi@science.ru.nl
morten.oygarden@simula.no
qingju.wang@uni.lu

Abstract. Algebraically simple PRFs, ciphers, or cryptographic hash functions are becoming increasingly popular, for example due to their attractive properties for MPC and new proof systems (SNARKs, STARKs, among many others).

In this paper, we focus on the algebraically simple construction MiMC, which became an attractive cryptanalytic target due to its simplicity, but also due to its use as a baseline in a competition for more recent algorithms exploring this design space.

For the first time, we are able to describe key-recovery attacks on all full-round versions of MiMC over $\mathbb{F}_{2^n}$, requiring half the code book. In the chosen-ciphertext scenario, recovering the key from this data for the n-bit full version of MiMC takes the equivalent of less than $2^{n-\log_2(n)+1}$ calls to MiMC and negligible amounts of memory.

The attack procedure is a generalization of higher-order differential cryptanalysis, and it is based on two main ingredients. First, we present a higher-order distinguisher which exploits the fact that the algebraic degree of MiMC grows significantly slower than originally believed. Secondly, we describe an approach to turn this distinguisher into a key-recovery attack without guessing the full subkey. Finally, we show that approximately $\lceil \log_3(2 \cdot R) \rceil$ more rounds (where $R = \lceil n \cdot \log_3(2) \rceil$ is the current number of rounds of MiMC-n/n) can be necessary and sufficient to restore the security against the key-recovery attack presented here.

The attack has been practically verified on toy versions of MiMC. Note that our attack does not affect the security of MiMC over prime fields.

Keywords: Algebraic attack · MiMC · Higher-order differential

1 Introduction

The design of symmetric cryptographic constructions exhibiting a clear and ideally low-degree algebraic structure is motivated by many recent use cases, for example the increasing popularity of new proof systems such as STARKs [8], SNARKs (e.g., Pinocchio [43]), Bulletproofs [19], and other concepts like secure multi-party computation (MPC). To provide good performance in these new applications, ciphers and hash functions are designed in order to minimize specific characteristics (e.g., the total number of multiplications, the depth, or other parameters related to the nonlinear operations). In contrast to traditional cipher design, the size of the field over which these constructions are defined has only a small impact on the final cost. In order to achieve this new performance goal, some crucial differences arise between these new designs and traditional ones. For example, we can consider the substitution (S-box) layer, that is, the operation providing nonlinearity in the permutation: In these new schemes, the S-boxes composing this layer are relatively large compared to the ones used in classical schemes (e.g., they operate over 64 or 128 bits instead of 4 or 8 bits) and/or they can usually be described by a simple low-degree nonlinear function (e.g., $x \mapsto x^d$ for some d). Examples of these schemes include LowMC [4], MiMC [3], Jarvis/Friday [6], GMiMC [2], HadesMiMC [30], *Vision*/*Rescue* [5], and Starkad/Poseidon [29].

The structure of these schemes has a significant impact on the attacks that can be mounted. While statistical attacks (including linear [41] and differential [11] ones) are among the most powerful techniques against traditional schemes, algebraic attacks turned out to be especially effective against these new primitives. In other words, these constructions are naturally more vulnerable to algebraic attacks than those which do not exhibit a clear and simple algebraic structure. For example, this has been shown in [1], in which algebraic strategies covering the full-round versions of the attacked primitives are described. Although the approaches can be quite different, most of them exploit the low degree of the construction.

In this paper, we focus on MiMC [3]. The MiMC design constructs a cryptographic permutation by iterated cubing, interleaved with additions of random constants to break any symmetries. A secret key is added after every such round to obtain a block cipher. The design of MiMC is very flexible and can work with binary strings as well as integers modulo some prime number. Security analysis by the designers rules out various statistical attacks, and the final number of rounds is derived from an analysis of attack vectors that exploit the simple algebraic structure. We remark that the designers chose the number of rounds with a minimal security margin for efficiency. For a more detailed specification and a summary of previous analysis, we refer to Section 2.3.

Since its publication in 2016, MiMC has become the preferred choice for many use cases that benefit from a low multiplication count or algebraic simplic-

Table 1: Various attacks on MiMC. In this representation, n denotes the block size (and key size). The unit for the attack complexity is usually the cost of a single encryption (number of multiplications over $\mathbb{F}_{2^n}$ necessary for a single encyption). The SK and KR attacks can be implemented using chosen plaintexts CP and/or chosen ciphertexts CC. The memory complexity is negligible for all approaches listed.

Type	n	Rounds	Time	Data	Source
KR*	129	38	$2^{65.5}$	$2^{60.2}$ CP	[40]
SK	129	80	2^{128} XOR	2^{128} CP/CC	Section 4.1
SK	n	$\lceil \log_3(2^{n-1}-1) \rceil - 1$	2^{n-1} XOR	2^{n-1} CP/CC	Section 4.1
KK	129	160 ($\approx 2 \times$ full)	–	2^{128}	Section 4.3
KK	n	$2 \cdot \lceil \log_3(2^{n-1}-1) \rceil - 2$	–	2^{n-1}	Section 4.3
KR	129	82 (full)	$2^{122.64}$	2^{128} CC	Section 5
KR	255	161 (full)	$2^{246.67}$	2^{254} CC	Section 5
KR	n	$\lceil n \cdot \log_3(2) \rceil$ (full)	$\leq 2^{n-\log_2(n)+1}$	2^{n-1} CC	Section 5

KR $\equiv$ Key-Recovery, KR* $\equiv$ attack on a variant of MiMC proposed in a low-memory scenario, SK $\equiv$ Secret-Key Distinguisher, KK $\equiv$ Known-Key Distinguisher

ity [31,44]. It also serves as a baseline for various follow-up designs evaluated in the context of the public "STARK-Friendly Hash Challenge" competition[5].

1.1 Our Contribution

As the main results in this paper, we present

(1) a new upper bound for the algebraic degree growth in key-alternating ciphers with low-degree round functions,
(2) a secret-key higher-order distinguisher on almost full MiMC over $\mathbb{F}_{2^n}$,
(3) a known-key zero-sum distinguisher on almost double the rounds of MiMC,
(4) the first key-recovery attack on *full-round* MiMC over $\mathbb{F}_{2^n}$.

We also show that the technique we use for MiMC is sufficiently generic to apply to any permutation fulfilling specific properties, which we will define in detail. Our attacks and distinguishers on MiMC, as well as other attacks in the literature, are listed in Table 1.

Secret-Key Higher-Order Distinguishers. After recalling some preliminary facts about higher-order differentials, in Section 3 we analyze the growth of the algebraic degree for key-alternating ciphers whose round function can be described as a low-degree polynomial over $\mathbb{F}_{2^n}$.

For an SPN cipher over a field $\mathbb{F}$ where each round has algebraic degree δ, the algebraic degree of the cipher is expected to grow essentially exponentially in

[5] `https://starkware.co/hash-challenge/`

δ. Several analyses made in the literature [20,18,17] confirm this growth for most ciphers, except when the algebraic degree of the function is close to its maximum. As a result, the number of rounds necessary for security against higher-order differential attacks generally grows logarithmically in the size of $\mathbb{F}$. Different behaviour has been observed for certain non-SPN designs, such as some designs with partial nonlinear layers where the algebraic degree grows exponentially in some (not necessarily integer) value smaller than δ [26].

In Section 3, we show that if the round function can be described as an invertible low-degree polynomial function in $\mathbb{F}_{2^n}$, then the algebraic degree grows linearly with the number of rounds, and not exponentially as generally expected. More precisely, let d denote the exponent of the power function $x \mapsto x^d$ used to define the S-boxes. Then, we show that in the case of key-alternating ciphers over $\mathbb{F}_{2^n}$, the algebraic degree $\delta(r)$ as a function in the number of rounds r is

$$\delta(r) \in \mathcal{O}(\log_2(d^r)) = \mathcal{O}(r).$$

As an immediate consequence, our observation implies that roughly $n \cdot \log_d(2)$ rounds are necessary to provide security against higher-order differential attacks, much more than the expected $\approx \log_\delta(n-1)$ rounds.

Distinguishers on MiMC over $\mathbb{F}_{2^n}$. Our new bounds on the number of rounds necessary to provide security against higher-order differential cryptanalysis have a major impact on key-alternating ciphers with large S-boxes. A concrete example for this class of ciphers is MiMC [3], a key-alternating cipher defined over $\mathbb{F}_{2^n}$ (for odd $n \in \mathbb{N}$), where the round function is simply defined as the cube map $x \mapsto x^3$. Since any cubic function over $\mathbb{F}_{2^n}$ has algebraic degree 2, one may expect that approximately $\log_2(n)$ rounds are necessary to prevent higher-order differential attacks. Our new bound implies that a much larger number of rounds is required to provide security, namely approximately $n \cdot \log_3(2)$.

As a concrete example, in Section 4 we show that MiMC-n/n has a security margin of only 1 or 2 rounds against (secret-key) higher-order distinguishers (depending on n), which is much smaller than expected by the designers. Moreover, we can set up a known-key distinguisher for approximately double the number of rounds of MiMC, by showing that the same number of rounds is necessary to reach the maximum degree in the decryption direction. Our findings have been practically verified on toy versions.

We remark that the designers presented other non-random properties (including GCD and interpolation attacks) that can cover a similar number of rounds. The number of rounds proposed by the designers were chosen in order to provide security against key-recovery attacks based on these properties. As we are going to show, the number of rounds is not sufficient against our new attack based on a higher-order differential property.

Results using the Division Property. For completeness, in Section 4.5 we search for higher-order distinguishers for MiMC-n/n with the division property [45] proposed by Todo at Eurocrypt 2015, a powerful tool for finding the best integral distinguishers for block ciphers. By modeling the most recently proposed variant of the bit-based division property, which is called *three-subset bit-based division property without unknown subset* in [33], we are able to reproduce exactly the same higher-order distinguishers for cases with small n-bit S-boxes, where $n \in \{5, 7, 9\}$. However, as far as we know, it is an open problem to model the three-subset bit-based division property for a larger S-box of size bigger than 9 in practical time. Therefore, we conclude that the division property is unlikely to help us for the ciphers we focus on.

Key-Recovery Attack on MiMC-n/n and on Generic Ciphers. A trivial way to extend an r-round distinguisher to an $(r+1)$-round key-recovery attack is based on guessing the last round key, partially decrypting/encrypting, and finally exploiting the distinguisher to filter wrong key guesses. Unfortunately, this strategy does not work for MiMC, since guessing the full last round key required to invert the large S-box is equivalent to exhaustive key search. Another key-recovery approach that has been combined with integral distinguishers is based on interpolating the Boolean polynomials that define the final rounds. However, this strategy requires evaluating the distinguisher several times to collect enough equations, which is not feasible for our distinguisher due to its large data complexity.

In Section 5, we show how to solve this problem. Instead of guessing the last round key, we set up an equation over $\mathbb{F}_{2^n}$ with the master key as a variable. To obtain this equation, we symbolically express the zero sum at the input to the last round as a polynomial function of the key, whose coefficients depend on the queried ciphertexts. We show how the resulting polynomial equation can be solved efficiently to recover the key. As a result, in the chosen-ciphertext case only, recovering the key from this data for the *full* n-bit version of MiMC takes the equivalent of less than $2^{n-\log_2(n)+1}$ calls to MiMC, 2^{n-1} chosen ciphertexts, and negligible amounts of memory. Moreover, we show that approximately $\lceil \log_3(2 \cdot R) \rceil$ more rounds (where $R = \lceil n \cdot \log_3(2) \rceil$ is the current number of rounds of MiMC-n/n) can be necessary and sufficient to restore the security against the key-recovery attack presented here. This would, for example, imply that we need to add 5 more rounds for the most used version MiMC-129/129 (which currently has 82 rounds).

A Generic Strategy. Our strategy is an instance of a broader class of algebraic key-recovery approaches based on solving equations in the key variables. As such, it shares some ideas with other algebraic approaches like optimized interpolation attacks. However, while most algebraic key-recovery approaches of the last years construct and solve systems of many Boolean linear equations, we use a single univariate equation of higher degree that can be solved with polynomial factoring algorithms such as Berlekamp's algorithm. In Section 6, we outline a more detailed and generic procedure for such an attack. It is interesting to note that a comparatively old technique which basically disappeared for the cryptanalysis of AES-like ciphers turns out to be very competitive for schemes with large S-boxes.

2 Preliminaries

In this section, we recall the most important results about polynomial representations of Boolean functions and summarize the currently best known results regarding the growth of the algebraic degree in the context of SP networks. We also provide the specification of MiMC and give an overview of previous cryptanalytic results.

We emphasize that in general it is only possible to give a *lower* bound regarding the number of rounds which we can attack using higher-order differential techniques, in the following denoted as "necessary number of rounds to provide security". While upper-bounding the algebraic degree is more important from an adversary's point of view, lower bounds on the degree are much more relevant when arguing about security against algebraic attacks (such as e.g. [39,37,48,24]) from a designer's viewpoint. However, at the current state of the art and to the best of our knowledge, it seems hard to find such a lower bound for a given cipher without investigating concrete instances experimentally – which, of course, limits the scope of any analysis.

2.1 Polynomial Representations over Binary Extension Fields

We denote addition (and subtraction) in binary extension fields by the symbol $\oplus$. For $n \in \mathbb{N}$, every function $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ can be uniquely represented by an n-tuple $(F_1, F_2, \ldots, F_n)$ of polynomials over $\mathbb{F}_2$ in n variables with a maximum degree of 1 in each variable. In this representation, F_i is of the form

$$F_i(X_1, \ldots, X_n) = \bigoplus_{u=(u_1,\ldots,u_n)\in\{0,1\}^n} \varphi_i(u) \cdot X_1^{u_1} \cdot \cdots \cdot X_n^{u_n}, \qquad (1)$$

where the coefficients $\varphi_i(u)$ can be computed by the *Moebius transform*.

As is common, we denote functions $F : \mathbb{F}_2^n \to \mathbb{F}_2$ as *Boolean functions* and functions of the form $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$, for $n, m \in \mathbb{N}$, as *vectorial Boolean functions*.

Definition 1. *The algebraic normal form (ANF) of a Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2$, as given in Eq. (1), is the unique representation as a polynomial over $\mathbb{F}_2$ in n variables and with a maximum univariate degree of 1. The algebraic degree $\delta(F)$ of F – or δ for simplicity – is the degree of the above representation of F as a multivariate polynomial over $\mathbb{F}_2$. If $G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is a vectorial Boolean function and $(G_1, \ldots, G_n)$ is its representation as an n-tuple of multivariate polynomials over $\mathbb{F}_2$, then its algebraic degree $\delta(G)$ is defined as $\delta(G) := \max_{1\leq i\leq n} \delta(G_i)$.*

The link between the algebraic degree and the univariate degree of a vectorial Boolean function is well-known, and is for example established in [22]: the algebraic degree of $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ can be computed from its univariate polynomial representation, and is equal to the maximum hamming weight of the 2-ary expansion of its exponents.

Lemma 1. *Let $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ be a function and let $F(X) = \sum_{i=0}^{2^n-1} \varphi_i \cdot X^i$ denote the corresponding univariate polynomial description over $\mathbb{F}_{2^n}$. The algebraic degree $\delta(F)$ of F as a vectorial Boolean function is the maximum hamming weight*[6] *of its exponents, i.e., it is $\delta(F) = \max_{0 \le i \le 2^n-1} \{\mathrm{hw}(i) \mid \varphi_i \neq 0\}$.*

2.2 Higher-Order Differential Cryptanalysis

Higher-order differential attacks [39,37] form a prominent class of attacks exploiting the low algebraic degree of a nonlinear transformation such as a classical block cipher. If this degree is sufficiently low, an attack using multiple input texts and their corresponding output texts can be mounted. In more detail, if the algebraic degree of a Boolean function f is δ, then, when applying f to all elements of an affine vector space $\mathcal{V} \oplus c$ of dimension greater than δ and taking the sum of these values, the result is 0, i.e., $\bigoplus_{v \in \mathcal{V} \oplus c} f(v) = 0$.

Security Against Higher-Order Differential Attacks – State of the Art. To prevent higher-order differential attacks against iterated block ciphers, one would usually want the maximum algebraic degree to be reached (well) within the suggested number of rounds. To achieve this goal, and to assess the security margins, it is crucial to estimate how the algebraic degree grows with the number of rounds.

The algebraic degree of composing two functions, $F, G : \mathbb{F}_2^n \to \mathbb{F}_2^n$, can be generically bounded by

$$\deg(F \circ G) \le \deg(F) \cdot \deg(G), \tag{2}$$

and hence an upper bound is found by iterative use of this on the round function. The resulting bound does, however, fail to reflect the real growth of the algebraic degree for many cryptosystems, and the problem of estimating the growth has been widely studied in the literature. After the initial work of Canteaut and Videau [20], a tighter upper bound was presented by Boura, Canteaut, and De Cannière [18] at FSE'11. There, the authors show how to deduce a new bound for the algebraic degree of iterated permutations for a special category of SP networks over $(\mathbb{F}_{2^n})^t$, which includes functions that have a number $t \ge 1$ of balanced S-boxes as their nonlinear layer. Specifically, the authors show that the algebraic degree of the considered SP network grows almost exponentially, except when it is close to its maximum.

Proposition 1 ([18]). *Let F be a function from $\mathbb{F}_2^N$ to $\mathbb{F}_2^N$ corresponding to the concatenation of t smaller S-boxes $S_1, \ldots, S_t$ defined over $\mathbb{F}_2^n$. Then, for any function G from $\mathbb{F}_2^N$ to $\mathbb{F}_2^N$, we have*

$$\deg(G \circ F(\cdot)) \le \min\left\{\deg(F) \cdot \deg(G), N - \frac{N - \deg(G)}{\gamma}\right\}, \textit{ where} \tag{3}$$

[6] Given $x = \sum_{i=0}^{\chi} x_i \cdot 2^i$ for $x_i \in \{0, 1\}$, the hamming weight of x is $\mathrm{hw}(x) = \sum_{i=0}^{\chi} x_i$.

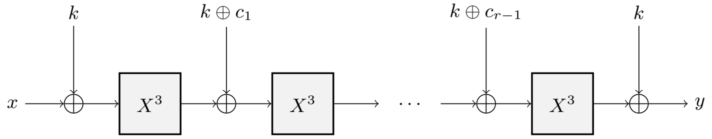

Fig. 1: The MiMC encryption function with r rounds.

$$\gamma = \max_{i=1,\ldots,n-1} \frac{n-i}{n-\delta_i} \leq n-1, \tag{4}$$

and where δ_i is the maximum degree of the product of any i coordinates of any of the smaller S-boxes.

Thus, the number of rounds necessary to prevent higher-order differential attacks is in general bigger than the one obtained using the trivial bound in Eq. (2).

2.3 Specification and Previous Analysis of MiMC

MiMC [3] is a key-alternating n-bit block cipher, where in each round the same n-bit key is added to the state. The nonlinear component of the construction is the evaluation of the cube function $f(x) = x^3$ over $\mathbb{F}_{2^n}$. Additionally, a different round constant is added in each round to break symmetries, where the first round constant is 0. The total number of rounds is then

$$r = \lceil n \cdot \log_3(2) \rceil ,$$

and we refer to Fig. 1 for a graphical representation of the encryption function.

MiMC is defined to work over prime fields and binary fields. In this paper, we focus on the binary field versions of MiMC[7], for which the block size n has to be odd in order for the S-box to be a permutation.

MiMC: Related Attacks in the Literature. The designers recommend MiMC with $\lceil n \cdot \log_3(2) \rceil$ rounds [3]. They derive this number of rounds by considering a variety of different key-recovery attacks on MiMC. According to their analysis, the most powerful attacks are interpolation [35] and GCD attacks. About higher-order differential attacks, the authors claim that "*the large number of rounds ensures that the algebraic degree of MiMC in its native field will be maximum or almost maximum. This naturally thwarts higher-order differential attacks* [...]".

The first attack on MiMC-n/n [40], presented at SAC 2019, targets a reduced-round version of MiMC proposed by the designers for a scenario in which the attacker has only limited memory, but it does not affect the security claims of

[7] Since the only subspaces of $\mathbb{F}_p$, where p is a prime number, are $\{0\}$ and $\mathbb{F}_p$ itself, our attack does not affect the security of MiMC over prime fields.

full-round MiMC. The Feistel version of MiMC was attacked shortly after, by using generic properties of the used Feistel construction instead of exploiting properties of the primitive itself [16]. Finally, a specific attack on MiMC using Gröbner bases was considered in [1]. The authors state that by introducing a new intermediate variable in each round, the resulting multivariate system of equations is already a Gröbner basis and thus the first step of a Gröbner basis attack is for free. However, recovering univariate polynomials from this representation and then applying techniques like the GCD attack will result in a prohibitively large computational complexity, since the recovered polynomials will be of degree $\approx 3^r$ after r rounds. Hence, the authors conclude that MiMC cannot be attacked directly by using known Gröbner basis techniques.

3 Higher-Order Differentials of Key-Alternating Ciphers

Our bound on the growth of the algebraic degree does not depend on the cubing of the round function in MiMC, so we introduce the following generalization of the result on MiMC from Section 2.3.

3.1 Setting

Let $E_k^r : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ be a key-alternating cipher defined by

$$E_k^r(x) := k_r \oplus R(\cdots R(k_1 \oplus R(k_0 \oplus x)) \cdots) \tag{5}$$

over $r \geq 1$ rounds, where $k_0, k_1, \ldots, k_r \in \mathbb{F}_{2^n}$ are derived from a master key $k \in \mathbb{F}_{2^n}$ using a key schedule. Each round function $R : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is defined as some invertible univariate polynomial function

$$R(x) := \rho_0 \oplus \bigoplus_{i=1}^{d} \rho_i \cdot x^i \tag{6}$$

of univariate degree $d \geq 3$, where $\rho_i \in \mathbb{F}_{2^n}$ and $\rho_d \neq 0$. We will, without loss of generality, assume $d \leq d_{\text{inv}}$, where d_{inv} denotes the degree of the compositional inverse of R (otherwise, an attacker would target the decryption function instead). Furthemore, we assume that the round function has *low* univariate degree, i.e., low compared to the size of $\mathbb{F}_{2^n}$. In other words, we work with $d \ll 2^n - 1$.

3.2 Growth of the Degree

In this section, we show that the algebraic degree δ of a key-alternating cipher E_k^r grows much slower than commonly presented in the literature. More precisely, in some cases it can grow linearly in the number of rounds and not exponentially.

Proposition 2. *Let E_k^r be a an r-round key-alternating block cipher with a round function R of degree d, as defined in Eq.* (5)*. If $r \leq \mathcal{R}_{lin} - 1$, where*

$$\mathcal{R}_{lin} = \left\lceil \log_d\left(2^{n-1} - 1\right) \right\rceil \approx (n-1) \cdot \log_d(2), \tag{7}$$

then the algebraic degree δ of E_k^r is at most $n-2$. Consequently, a (secret-key) higher-order distinguisher using at most 2^{n-1} data can be applied to E_k^r.[8]

Proof. Due to the relation between the word-level degree and the algebraic degree, E_k^r reaches its maximum algebraic degree of $n-1$ if at least one monomial with the exponent $2^n - 2^j - 1$ (for $0 \leq j < n$) appears in the polynomial representation. Indeed, note that all these monomials have an algebraic degree of $n-1$. Since the smallest exponent of this form is $2^n - 2^{n-1} - 1 = 2^{n-1} - 1$, and since the degree of E_k^r after r rounds is at most d^r, we require $d^r \geq 2^{n-1} - 1$ to make $x^{2^{n-1}-1}$ appear, or equivalently,

$$r \geq \lceil \log_d(2^{n-1} - 1) \rceil.$$

Hence, the degree is not maximal for $r < \lceil \log_d(2^{n-1} - 1) \rceil$ and a higher-order distinguisher using at most 2^{n-1} data can be applied. □

The Difficulty of Lower-Bounding the Growth of the Degree. We point out that it is always possible to set up a (secret-key) higher-order distinguisher if the number of rounds is smaller than $\mathcal{R}_{\text{lin}}$. However, a number of rounds greater than or equal to $\mathcal{R}_{\text{lin}}$ does not necessarily provide security.

One of the main problems in order to derive a sufficient condition for the number of rounds that provides security is the difficulty of analyzing the non-vanishing coefficients in the polynomial representation of E_k^r. Note, in general it is not easy to give a condition guaranteeing that a particular monomial appears, since many factors (including the secret key, the constant addition, and the details of the S-box) influence the result.

Without going into the details, we consider the influence of the S-box in some concrete examples. Working with $R(x) = x^d$ for a certain $3 \leq d \leq 2^n - 2$ (where $d \neq 2^{d'}$ for $d' \in \mathbb{N}$), we focus for simplicity only on two extreme cases $d = 2^{d'} \pm 1$. By exploiting Lucas's Theorem[9]:

- If $d = 2^{d'} + 1$ for some $d' \in \mathbb{N}$, then the output of a single round is sparse:

$$(x \oplus y)^{2^{d'}+1} = x^{2^{d'}+1} \oplus x^{2^{d'}} \cdot y \oplus y^{2^{d'}} \cdot x \oplus y^{2^{d'}+1}$$

  (note that it contains only 4 terms instead of $d + 1 = 2^{d'} + 2$).
- If $d = 2^{d'} - 1$ for some $d' \in \mathbb{N}$, then the output of a single round is full, since

$$(x \oplus y)^{2^{d'}-1} = \bigoplus_{i=0}^{2^{d'}-1} x^i \cdot y^{2^{d'}-1-i}.$$

[8] We denote our bound by $\mathcal{R}_{\text{lin}}$ to indicate the almost linear growth of the algebraic degree for this specific class of constructions.

[9] By Lucas's Theorem, $\binom{n}{m} \equiv \prod_{i=0}^{k} \binom{n_i}{m_i} \pmod 2$, it follows that where $n = \sum_{i=0}^{k} n_i \cdot 2^i$ and $m = \sum_{i=0}^{k} m_i \cdot 2^i$ is the 2-ary expansion of n and m, respectively.

Even if a single round is not sparse, the output of several combined rounds is not guaranteed to be full (even if it is in general dense). As a concrete example, while the output of $(x \oplus k_0)^3 \oplus k_1$ is full, the same is not true for

$$\begin{aligned}((x \oplus k_0)^3 \oplus k_1)^3 \oplus k_2 = x^9 \oplus x^8 \cdot k_0 \oplus x^6 \cdot k_1 \oplus x^4 \cdot k_0^2 \cdot k_1 \oplus x^3 \cdot k_1^2 \\ \oplus x^2 \cdot (k_0 \cdot k_1^2 \oplus k_0^2 \cdot k_1^2 \oplus k_0^4 \cdot k_1) \oplus x \cdot k_0^8 \oplus c(k_0, k_1, k_2),\end{aligned} \tag{8}$$

where both x^5 and x^7 are missing, and where $c(k_0, k_1, k_2)$ is a function that depends only on the keys. This simple example emphasizes the difficulty of analyzing the sparsity of the polynomial that defines E_k.

3.3 Comparison with Other Bounds

We now compare the new number of rounds necessary to provide security against secret-key higher-order distinguishers with other possible bounds. An alternative strategy is to apply generic bounds focusing on the algebraic degree of the round function, as recalled in Proposition 1. Recall that $\mathcal{R}_{\text{lin}}$ is the number of rounds from Proposition 2, and we will denote the number of round based on generic bounds by $\mathcal{R}_{\text{gen}}$. The comparison will make use of $\delta_{\text{lin}}(r)$, the upper bound on the algebraic degree after r rounds following Proposition 2. The upper bound from Eq. (3) will be denoted by $\delta_{\text{gen}}(r)$. Note that $\delta_{\text{gen}}(r)$ can, for example, take advantage of a slower growth in the algebraic degree, as in Eq. (8) by considering two rounds instead of one. Despite this, the overall trend of $\delta_{\text{gen}}(r)$ will still be exponential. On the other hand, if the round function can be described by a polynomial of low univariate degree d over $\mathbb{F}_{2^n}$, we expect a linear behaviour in $\delta_{\text{lin}}(r)$:

$$\delta_{\text{lin}}(r) \leq \lfloor \log_2(d^r + 1) \rfloor \approx r \cdot \log_2(d).$$

As a result, the round numbers $\mathcal{R}_{\text{lin}}$ and $\mathcal{R}_{\text{gen}}$ *necessary* to provide security grow respectively linearly and logarithmically in the size n of the field, namely

$$\mathcal{R}_{\text{lin}} \in \mathcal{O}(n) \qquad \text{and} \qquad \mathcal{R}_{\text{gen}} \in \mathcal{O}(\log_\delta(n)).$$

A concrete comparison of $\delta_{\text{lin}}(r)$ and $\delta_{\text{gen}}(r)$ for MiMC-129/129 is given in Fig. 2. In this setting we have $\delta_{\text{lin}}(r) = \lfloor \log_2(3^r + 1) \rfloor$, and $\delta_{\text{gen}}(r)$ has been derived using the observation that two rounds of MiMC have algebraic degree two (see Appendix A for more details). In particular, we find $\mathcal{R}_{\text{gen}} = 13$ and $\mathcal{R}_{\text{lin}} = 81$.

Remark. We emphasize that every (invertible) S-box/round function in $\mathbb{F}_2^n$ can be rewritten as a polynomial over $\mathbb{F}_{2^n}$. The crucial point here is that given a "random" S-box/round function over $\mathbb{F}_2^n$, the corresponding polynomial over $\mathbb{F}_{2^n}$ has in general a high univariate degree (e.g., $d \approx 2^n - \varepsilon$ for some small ε). In such a case, even if our argument still holds, the final result becomes meaningless, since $\log_d(2^n - 1) \approx \log_{2^n - \varepsilon}(2^n - 1) \approx 1$ is basically constant (i.e., it does not grow linearly with n). Hence, our results turn out to be relevant only for S-boxes/round functions for which the corresponding polynomial over $\mathbb{F}_{2^n}$ has "small" degree (namely, small compared to the field size, i.e., $d \ll 2^n$).

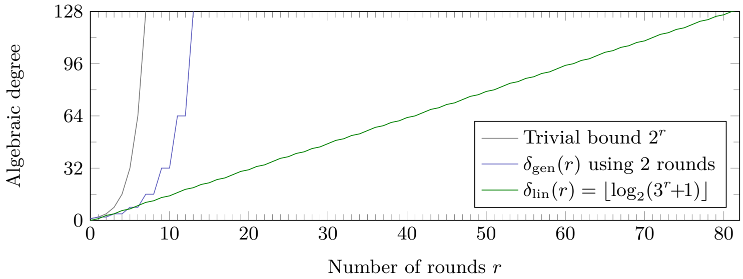

Fig. 2: Different upper bounds of the growth of the algebraic degree for MiMC-129/129. The trivial bound is 2^r. A tighter bound, $\delta_{\text{gen}}(r)$, exploits the observation that 2 rounds only have degree 2 (see Eq. (8)). Our new bound, $\delta_{\text{lin}}(r)$, is linear in the number of rounds.

4 Distinguishers for Reduced-Round and Full MiMC

Exploiting the previous result, we now discuss the possibility to set up higher-order differential distinguishers and attacks on MiMC [3]. We show that

(1) MiMC has a security margin of only 1 or 2 round(s) against (secret-key) higher-order distinguishers, depending on n, and that
(2) a zero-sum known-key distinguisher can be set up for approximately double the number of rounds of MiMC.

4.1 Secret-Key Higher-Order Distinguisher for MiMC

The results just presented allow to set up a nontrivial (secret-key) higher-order distinguisher on $\lceil \log_3(2^{n-1}-1)\rceil - 1$ rounds of MiMC, where $\lceil \log_3(2^{n-1}-1)\rceil - 1 < \lceil n \cdot \log_3(2) \rceil$ for all n. Consequently, the security margin is reduced to

$$1 \leq \lceil n \cdot \log_3(2) \rceil - \big(\lceil \log_3(2^{n-1}-1) \rceil - 1\big) \leq 2$$

rounds. To give some concrete examples, MiMC has 1 round of security margin for $n \in \{33, 63, 255\}$, and 2 rounds of security margin for $n \in \{31, 65, 127, 129\}$.

4.2 Practical Results

In this section we compare the results from Proposition 2 with practical results from scaled-down versions of MiMC. The tests[10] have been performed in the following way: Instead of computing the ANF of a keyed permutation (which

[10] The source code for the attacks and the tests is available on `https://github.com/IAIK/mimc-analysis`.

Table 2: Theoretical and practical round numbers *necessary* to prevent higher-order distinguishers for MiMC over $\mathbb{F}_{2^n}$.

Param.	Theoretical		Practical
n	$\mathcal{R}_{\text{lin}}$	$\mathcal{R}_{\text{gen}}$	$\mathcal{R}$
7	4	5	5
9	6	5	6
11	7	7	7
13	8	7	9
15	9	7	10
17	11	7	11
33	21	9	21
65	41	11	-
129	81	13	-

is expensive even for small field sizes), we evaluate the higher-order differential zero-sum property (as given in Section 2.2) for a specific input vector space. Namely, for random keys, random constants, and an input subspace of dimension $n-1$, we look for the minimum number of rounds r for which the corresponding sum of the ciphertexts is different from zero. Such a number corresponds to the number of rounds necessary to prevent higher-order distinguishers. In order to avoid the influence of weak keys or round constants, we repeated the tests multiple times (with new random keys and round constants). The practical number of rounds we give in each row is *the smallest number of rounds among all tested keys and round constants necessary* to prevent higher-order distinguishers. This means that a potentially higher number of rounds can be attacked by choosing the keys and round constants in a particular way.

The results, denoted $\mathcal{R}$, are given in Table 2. We also present $\mathcal{R}_{\text{lin}}$ (from Proposition 2) and $\mathcal{R}_{\text{gen}}$ (see Appendix A) for comparison. We emphasize that the theoretical values predicted by $\mathcal{R}_{\text{lin}}$ match the practical results in about half of the cases, and are off by at most one.

4.3 Known-Key Zero-Sum Distinguisher for MiMC

A known-key distinguisher is a scenario introduced in [38] where the attacker knows the key, and it is important in all settings in which no secret material is present. To succeed, the attacker has to discover some property of the attacked cipher that holds with a probability higher than for an ideal cipher, or is believed to be hard to exhibit generically. The goal of a known-key zero-sum distinguisher is to find a set of plaintexts and ciphertexts whose sums are equal to zero. To do this, the idea is to exploit the inside-out approach. By choosing a subspace of texts $\mathcal{V}$, one simply defines the plaintexts as the r_{dec}-round decryption of $\mathcal{V}$ and the ciphertexts as the r_{enc}-round encryption of $\mathcal{V}$. Such a distinguisher can then cover $r_{\text{enc}} + r_{\text{dec}}$ rounds. Examples of this approach are given in the literature for Keccak [18,7,10], Luffa [18,7], or PHOTON [49].

In the case of MiMC, the idea is to choose $\mathcal{V}$ as a subspace of $\mathbb{F}_{2^n}$ of dimension $n-1$. The maximum number of encryption rounds r_{enc} for which it is possible to guarantee a zero sum has been given in the previous paragraph. Based on Section 4.2, we can set up a known-key distinguisher on (more than) full MiMC-n/n. For our distinguisher on MiMC, we first recall the following result from [17].

Proposition 3 (Corollary 3 of [17]). *Let F be a permutation of $\mathbb{F}_2^n$. Then,* $\deg(F^{-1}) = n-1$ *if and only if* $\deg(F) = n-1$.

Corollary 1. *Let r_{enc} be the number of rounds necessary for MiMC over $\mathbb{F}_{2^n}$ to reach its maximum algebraic degree in the encryption direction. The same number of rounds is necessary for reaching the maximum algebraic degree in the decryption direction, i.e., $r_{dec} = r_{enc} = \lceil \log_3(2^{n-1}-1) \rceil$.*

It follows that, given a subspace $\mathcal{V} \subseteq \mathbb{F}_{2^n}$ of dimension $n-1$, the sums of the corresponding texts after $r_{\text{dec}}-1$ decryption rounds and $r_{\text{enc}}-1$ encryption rounds are always equal to zero, i.e.,

$$\underbrace{\bigoplus_{w \in \mathcal{V} \oplus v} R^{-(r_{\text{dec}}-1)}(w) = 0}_{\text{Zero sum}} \xleftarrow{R^{-(r_{\text{dec}}-1)}} \mathcal{V} \oplus v \xrightarrow{R^{r_{\text{enc}}-1}} \underbrace{0 = \bigoplus_{w \in \mathcal{V} \oplus v} R^{r_{\text{enc}}-1}(w)}_{\text{Zero sum}}$$

for each $v \in \mathbb{F}_{2^n}$. Hence, a known-key zero-sum distinguisher can be set up for

$$\begin{aligned} 2 \cdot (\lceil \log_3(2^{n-1}-1) \rceil - 1) &\approx 2(n-1) \cdot \log_3(2) - 2 = \\ &= \underbrace{n \cdot \log_3(2)}_{= \text{ full MiMC}} + [(n-2) \cdot \log_3(2) - 2] \end{aligned}$$

rounds of MiMC-n/n, which is much more than *full* MiMC-n/n.

4.4 Impact of the Known-Key Distinguisher on Full MiMC

Sponge Function. In [3], the authors propose a hash function by instantiating a sponge construction with MiMC$^\pi$, a fixed-key version of MiMC. The sponge hash function is indifferentiable from a random oracle up to $2^{c/2}$ calls to the internal permutation P (where c is the capacity) if P is modeled as a randomly chosen permutation [9]. Thus, even if it is not strictly necessary, it is desirable that MiMC is resistant against known-key distinguishers.

For completeness, we mention that even if there is a way to distinguish a permutation from a random one, it seems difficult to exploit a zero-sum distinguisher of the internal permutation of a sponge construction in order to attack the hash function. To give a concrete example, consider the case of Keccak: As a consequence of the zero-sum distinguisher found on 18-round Keccak-f[1600], the number of rounds has been increased from 18 to 24 in the second round of the SHA-3 competition in order to avoid "non-ideal" properties (see [18,10] for more details). However, the best known attack on the Keccak hash function can only be set up when using 6-/7-round Keccak-f [32].

In any case, we remark that such distinguishers based on zero sums cannot be set up for an arbitrary number of rounds, and they do indeed exploit the internal properties of a primitive using the inside-out approach found in this paper and in other literature. Hence, they cannot be considered meaningless.

Other Approaches. Even though the original MiMC paper only specifies a sponge-based hash function using MiMC, there are various applications and/or specific considerations that would make a block-cipher-based approach more advantageous (like, for example, being forced to use a block size which is too small for a sponge-based approach). Another way to turn a block cipher into a hash function is to use a compression function like the Davies–Meyer one together with something like the Merkle–Damgård construction. Similar to the case of sponge constructions, the security of such an algorithm is proven in the ideal cipher model [12]. This choice is, however, not supported by the MiMC designers, who use our results to support their advice against using a block-cipher-based approach (even though such implementations can still be found[11]). It follows that, since the attacker has control of the key in such scenarios, it is desirable for MiMC to be resistant against known- and chosen-key distinguishers, even if it does not seem to be strictly necessary.

4.5 Results Using the Division Property

Finally, in Appendix C we present our practical results obtained using "Mixed Integer Linear Programming (MILP)", which models the propagation of the (conventional) bit-based division property.

The (conventional) bit-based division property [47] was proposed to investigate integral characteristics of block ciphers at a bit level. With this approach, the integral property of each bit is studied independently. Naturally, this strategy allows to capture more information of the propagation than the word-level version, and thus integral characteristics for more rounds can be found with this new technique. For example, the integral distinguishers of SIMON32 have been improved from 10 rounds [45] (the current best result at word level) to 14 rounds [51] (obtained by the experimental method cited before).

Instead of separating the parity into the two cases "0" and "unknown" as for the (conventional) bit-based division property, three-subset bit-based division property [47] was introduced to enhance the accuracy of the conventional one, where the parity is separated into three sets, i.e., "0", "1", and "unknown". It shows that the three-subset bit-based division property can indeed be more accurate than the two-subset bit-based division property for some ciphers [34,52]. However, it becomes harder to efficiently model the three-subset division property propagation even for ciphers with simple structures. Recently, [33] pointed out

[11] `https://github.com/HarryR/ethsnarks/blob/master/src/gadgets/mimc.hpp`

that the three-subset division property has a couple of known problems when applied to cube attacks, and proposed a modified three-subset bit-based division without the “unknown” set to overcome these problems. By modeling this modified version of the three-subset bit-based division property for our cases with small n-bit S-boxes, where $n \in \{5, 7, 9\}$, we confirm the practical results given in Table 2.

However, as far as we know, it is still an open problem to model the (modified) three-subset bit-based division property for a larger S-box of size bigger than 9. The S-boxes we focus on in this paper can be described as a (low-degree) polynomial function in $\mathbb{F}_{2^n}$, where n is much larger than 9. Therefore, *the division property, which is commonly believed as the most efficient tool to find the best integral distinguishers, might not help us as much for the ciphers we focus on.*

5 Key-Recovery Attack on MiMC

Since the security margin of MiMC with respect to a (secret-key) higher-order distinguisher is of only 1 or 2 round(s) depending on n, it is potentially possible to extend a distinguisher to a key-recovery attack. Given a subspace $\mathcal{V}$ of plaintexts whose sum is equal to zero after r rounds, we can consider $r+1$ rounds, partially guess the last subkey and decrypt, and filter wrong key guesses that do not satisfy the zero sum:

$$\mathcal{V} \oplus v \xrightarrow{R^r(\cdot)} \underbrace{\bigoplus_{w \in \mathcal{V} \oplus v} R^r(w) = 0}_{\text{Higher-order distinguisher}} \xleftarrow[\text{Key guessing}]{R^{-1}(\cdot)} \underbrace{\{R^{r+1}(w) \mid w \in \mathcal{V} \oplus v\}}_{\text{Ciphertexts}}.$$

However, since the subkeys of MiMC are equal to the master key plus constants, and due to the single full-state S-box, even a (partial) decryption of a single round requires guessing the full key. As a result, a key-recovery attack on full MiMC based on this strategy seems infeasible.

In this section, we present an alternative strategy that allows to break full-round MiMC. Since a trivial key-guessing approach is inefficient, our idea is to construct a polynomial of low degree, which we can then try to solve.

5.1 Strategy of the Attack

From Proposition 2 and Proposition 3, a zero sum can be set up for at least $\lceil (n-1)\log_3(2) \rceil - 1 = \lceil n \log_3(2) \rceil - \varepsilon$ rounds in the encryption and decryption direction with a vector space $\mathcal{V} \oplus v$ of dimension $n-1$, where $\varepsilon \in \{1, 2\}$. Recalling that $\lceil n \cdot \log_3(2) \rceil$ is the number of rounds of full MiMC, we define r_{ZS}, r_{KR} as

$$r_{\text{ZS}} = \lceil (n-1)\log_3(2) \rceil - 1 \qquad \text{and} \qquad r_{\text{KR}} = 1 + \left(\lceil n \log_3(2) \rceil - \lceil (n-1)\log_3(2) \rceil\right),$$

where r_{ZS} is the number of rounds that we can cover with a zero sum, $r_{\text{KR}} = \lceil n \cdot \log_3(2) \rceil - r_{\text{ZS}} \in \{1, 2\}$.

Let $f^r(x, K)$ be the function corresponding to r rounds of $\text{MiMC}_k(\cdot)$ (and $f^{-r}(x, K)$ be r rounds of decryption, $\text{MiMC}_k^{-1}(\cdot)$), where x is the input text and

K is a symbolic variable that represents the secret key k. We intend to use these functions to create a polynomial from which we can deduce k. More precisely, for a fixed vector space $\mathcal{V} \oplus v$, we consider the equations

$$\underbrace{\bigoplus_{x \in \mathrm{MiMC}_k^{-1}(\mathcal{V}\oplus v)} f^{r_{\mathrm{KR}}}(x, K) = 0}_{=F(K)} \quad \text{and} \quad \underbrace{\bigoplus_{x \in \mathrm{MiMC}_k(\mathcal{V}\oplus v)} f^{-r_{\mathrm{KR}}}(x, K) = 0.}_{=G(K)} \tag{9}$$

After having received all x values from an oracle, the attacker can construct one of the polynomials $F(K) = 0$ or $G(K) = 0$. The secret key k can now be determined by finding the roots of either of these polynomials.

In the case of MiMC, the degree of a single encryption round is 3, while the degree of a single decryption round is $(2^{n+1} - 1)/3$ (which is significantly larger than 3 for large n). Due to the slow degree growth in the encryption direction of MiMC, we will focus on finding the roots of $F(K)$ given in Eq. (9).

Finding the Roots of Univariate Polynomials. Let $F(X) \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} + X\rangle$ be a univariate polynomial of degree D. Furthermore, let $M(D)$ denote a number such that multiplying two polynomials of degree $\leq D$ over $\mathbb{F}_{2^n}$ requires $\mathcal{O}(M(D))$ operations in $\mathbb{F}_{2^n}$. For instance, a straightforward method would yield $M(D) = D^2$, whereas $M(D) = D \cdot \log(D) \cdot \log\log(D)$ holds for methods based on fast Fourier transforms [21]. The *Berlekamp algorithm* for determining the roots of F is then expected to require $\mathcal{C} \in \mathcal{O}\left(M(D)\log(D)\log\left(2^n D\right)\right)$ operations in $\mathbb{F}_{2^n}$ (see [28, Chapter 14.5]).

5.2 Details of the Attack

Assume $\mathcal{V} \oplus v$ is a coset of a subspace $\mathcal{V}$ of dimension $n-1$. We define

$$\mathcal{W} = \mathrm{MiMC}_k^{-1}(\mathcal{V} \oplus v) \equiv \{\mathrm{MiMC}_k^{-1}(x) \in \mathbb{F}_{2^n} \mid x \in \mathcal{V} \oplus v\}$$

under a fixed secret key k. Here, we present the details of the attack for the cases $r_{\mathrm{KR}} = 1$ and $r_{\mathrm{KR}} = 2$, and we analyze the computational cost. We introduce the following notation:

$$\forall d \in \mathbb{N}: \qquad \mathscr{P}_d := \bigoplus_{x \in \mathcal{W}} x^d, \tag{10}$$

and whenever possible we will make use of the fact that squaring is a linear operation over $\mathbb{F}_{2^n}$. More specifically, computing $\mathscr{P}_{2d}$ only requires a single squaring operation once $\mathscr{P}_d$ is calculated:

$$\mathscr{P}_{2d} := \bigoplus_{x \in \mathcal{W}} x^{2d} = \left(\bigoplus_{x \in \mathcal{W}} x^d\right)^2 = \mathscr{P}_d^2. \tag{11}$$

This allows to reduce the total number of `XOR` operations.

Algorithm 1: Attack on MiMC – Case: $r_{\text{KR}} = 1$.

```
Input: Vector subspace $\mathcal{V}$ of ciphertexts of dimension $\dim(\mathcal{V}) = n - 1$.
Output: Secret key $k$.
$\mathscr{P}_1, \mathscr{P}_2, \mathscr{P}_3 \leftarrow 0$.
for $x \in \mathcal{V} \oplus v$ do
    $p \leftarrow \text{MiMC}_k^{-1}(x)$ from the decryption oracle.
    $\mathscr{P}_1 \leftarrow \mathscr{P}_1 \oplus p$.
    $q \leftarrow p^2$.
    $\mathscr{P}_3 \leftarrow \mathscr{P}_3 \oplus q \cdot p$.
$\mathscr{P}_2 \leftarrow (\mathscr{P}_1)^2$.
$F(K) = \mathscr{P}_1 \cdot K^2 \oplus \mathscr{P}_2 \cdot K \oplus \mathscr{P}_3$.
Find a solution $k$ of $F(K) = 0$ – see Section 5.1 (filter multiple solutions by
  brute force).
return $k$.
```

Case: $r_{\text{KR}} = 1$. Since a single round of MiMC is described by $(x \oplus k)^3 = k^3 \oplus k^2 \cdot x \oplus k \cdot x^2 \oplus x^3$, the function $F(K)$ is given by

$$F(K) = K^2 \cdot \mathscr{P}_1 \oplus K \cdot \mathscr{P}_2 \oplus \mathscr{P}_3.$$

A complete pseudo code of the attack can be found in Algorithm 1, which makes it easy to see that the cost of the attack is well approximated by

- $|\mathcal{V}| = 2^{n-1}$ multiplications,
- $|\mathcal{V}| = 2^{n-1} + 1$ squarings,
- $2 \cdot |\mathcal{V}| + 1 = 2^n + 1$ n-bit `XOR` operations,
- cost of finding the roots of a univariate polynomial of degree 2.

Case: $r_{\text{KR}} = 2$. The attack for the case $r_{\text{KR}} = 2$ is similar. From Eq. (8) (using $k_0 = k$, $k_1 = k \oplus c_1$ and $k_2 = 0$), the function $F(K)$ is described by

$$\begin{aligned} &F(K) = K^8 \cdot \mathscr{P}_1 \oplus K^5 \cdot \mathscr{P}_2 \oplus K^4 \cdot (\mathscr{P}_2 \cdot c_1 \oplus \mathscr{P}_1) \oplus K^3 \cdot (\mathscr{P}_4 \oplus \mathscr{P}_2) \\ &\oplus K^2 \cdot (\mathscr{P}_4 \cdot c_1 \oplus \mathscr{P}_3 \oplus \mathscr{P}_1 \cdot c_1^2) \oplus K \cdot (\mathscr{P}_8 \oplus \mathscr{P}_6 \oplus \mathscr{P}_2 \cdot c_1^2) \oplus (\mathscr{P}_9 \oplus \mathscr{P}_6 \cdot c_1 \oplus \mathscr{P}_3 \cdot c_1^2), \end{aligned}$$

where c_1 is the round constant of the first round. As also noted in Section 3.2, while $\mathscr{P}_9$ is the largest $\mathscr{P}_d$ in this expression, both $\mathscr{P}_5$ and $\mathscr{P}_7$ are missing, and hence do not need to be computed. A complete pseudo code of the attack can be found in Algorithm 2. Again, it is easy to see that the cost of the attack is well approximated by

- $2 \cdot |\mathcal{V}| + 6 = 2^n + 6$ multiplications,
- $2 \cdot |\mathcal{V}| + 4 = 2^n + 4$ squarings,
- $3 \cdot |\mathcal{V}| + 8 = 3 \cdot 2^{n-1} + 8$ n-bit `XOR` operations,
- cost of finding the roots of a univariate polynomial of degree 8.

Algorithm 2: Attack on MiMC – Case: $r_{\text{KR}} = 2$.

```
Input: Vector subspace $\mathcal{V}$ of ciphertexts of dimension $\dim(\mathcal{V}) = n - 1$.
Output: Secret key $k$.
$\mathscr{P}_1, \mathscr{P}_2, \mathscr{P}_3, \ldots, \mathscr{P}_9 \leftarrow 0$.
for $x \in \mathcal{V} \oplus v$ do
    $p \leftarrow \text{MiMC}_k^{-1}(x)$ from the decryption oracle.
    $\mathscr{P}_1 \leftarrow \mathscr{P}_1 \oplus p$.
    $q_2 \leftarrow p^2$.
    $q_3 \leftarrow q_2 \cdot p$.
    $\mathscr{P}_3 \leftarrow \mathscr{P}_3 \oplus q_3$.
    $q_6 \leftarrow q_3^2$.
    $\mathscr{P}_9 \leftarrow \mathscr{P}_9 \oplus q_6 \cdot q_3$.
$\mathscr{P}_2 \leftarrow (\mathscr{P}_1)^2$.
$\mathscr{P}_4 \leftarrow (\mathscr{P}_2)^2$.
$\mathscr{P}_6 \leftarrow (\mathscr{P}_3)^2$.
$\mathscr{P}_8 \leftarrow (\mathscr{P}_4)^2$.
$F(K) = K^8 \cdot \mathscr{P}_1 \oplus K^5 \cdot \mathscr{P}_2 \oplus K^4 \cdot (\mathscr{P}_2 \cdot c_1 \oplus \mathscr{P}_1) \oplus K^3 \cdot (\mathscr{P}_4 \oplus \mathscr{P}_2) \oplus K^2 \cdot$
  $(\mathscr{P}_4 \cdot c_1 \oplus \mathscr{P}_3 \oplus \mathscr{P}_1 \cdot c_1^2) \oplus K \cdot (\mathscr{P}_8 \oplus \mathscr{P}_6 \oplus \mathscr{P}_2 \cdot c_1^2) \oplus (\mathscr{P}_9 \oplus \mathscr{P}_6 \cdot c_1 \oplus \mathscr{P}_3 \cdot c_1^2)$.
Find a solution $k$ of $F(K) = 0$ (filter multiple solutions by brute force).
return $k$.
```

5.3 Complexity Estimation

As we have just seen, our attack requires half of the code book (namely, 2^{n-1} chosen ciphertexts). Here we show that our attacks are better than exhaustive search (from the computational point of view). In order to do this, we measure the time complexities in equivalent encryption operations.

A single encryption round in MiMC requires one addition, one squaring operation, and one multiplication in the extension field. Since the cost of a single n-bit `XOR` operation is much smaller than the cost of a multiplication over $\mathbb{F}_{2^n}$, and since the number of `XOR` operations is similar to the number of multiplications, in the following we do not consider `XOR` operations. After this simplification, we find that the time complexity of $r_{\text{KR}} = 1$ is dominated by 2^{n-1} squaring and multiplication operations or, equivalently, 2^{n-1} encryption rounds. A similar line of reasoning reveals that $r_{\text{KR}} = 2$ is comparable to 2^n encryption rounds.

Since the cost of solving a single low-degree equation is negligible, and one unit of encryption contains $\lceil n \cdot \log_3(2) \rceil$ rounds, it follows that the cost of our attacks is about

$$\frac{r_{\text{KR}} \cdot 2^{n-1}}{\lceil n \cdot \log_3(2) \rceil}$$

encryptions for $r_{\text{KR}} \in \{1, 2\}$. That is, the computational cost of the key-recovery part of our attacks is upper-bounded by $2^{n-\log_2(n)+1}$, and hence the total cost is smaller than that of a brute-force attack (namely, 2^n encryptions) for each $n \geq 3$.

5.4 Practical Verification

We implemented Algorithm 1 and Algorithm 2 in the computer algebra system Magma, and verified both algorithms for all odd integers $n \in [5, 35]$. We note that Algorithm 1 ($r_{\text{KR}} = 1$) yields the correct answer for all the tested $5 \le n \le 35$, even if $\lceil n \log_3(2) \rceil \neq \lceil (n-1) \log_3(2) \rceil$. Namely, in practice it is possible to cover one more round with a zero sum than what we theoretically expect. In other words, $\lceil (n-1) \log_3(2) \rceil$ rounds of the decryption function of MiMC fail to obtain the maximum algebraic degree for these parameters, which is reached after $\lceil (n-1) \log_3(2) \rceil + 1$ rounds (see Appendix B for more details on the degree growth of MiMC^{-1}). Since we are not able to prove this behavior for larger values of n, we leave it as an open question whether Algorithm 1 can be applied to MiMC for odd integers $n > 35$.

Considerations on Data and Computational Costs of this Attack. A possible drawback of our attack is the cost. Since we are not able to provide an estimation of the growth of the degree in the decryption direction, we can only exploit the fact that a certain number of rounds are necessary in order to achieve maximum degree. It follows that the attacker is forced to use half of the code book in order to set up the attack, which also has an impact on the computational cost.

Even if our attack is not practical, we believe it provides valuable theoretical insight. It is also in line with several other attacks found in the literature, which are set up under a similar assumption on the data and/or computational cost. To give some concrete examples, consider the case of zero-correlation attacks [14], which exploit linear approximations that hold with probability $\frac{1}{2}$. The crucial limitation for basic zero-correlation linear cryptanalysis is that it requires half of the code book. Only follow-up works have been able to reduce this data requirement, including the more powerful distinguisher called multiple zero-correlation (MPZC) linear distinguisher proposed in [15], which exploits the fact that there are numerous zero-correlation linear approximations in susceptible ciphers. While needing (close to) the full code book is an inherent property of zero-correlation attacks, the reason for the high data complexity in our case is purely due to the specification of MiMC and the attacked number of rounds, and not due to an inherent property of our attack.

Splice-and-cut meet-in-the-middle attacks and biclique attacks are other examples of attacks that often come with time complexities relatively close to exhaustive search. Indeed, an extension of the biclique approach first described in [13] has a brute-force phase for a number of rounds as part of the attack. It can in principle work for any number of rounds and is hence best described as a particular optimization of brute-force key guessing. However, later variants then showed examples where the gain over brute force was in the order of millions [36]. Still, we note that the complexity of biclique attacks scales differently than our attack, whose runtime cost depends strongly on the details of the target cipher MiMC.

Finally, we point out that any attack that is better than brute force is relevant, even if it requires unrealistic amounts of data or storage. Indeed, the main goal of cryptanalysis is finding a "certificated weakness", that is, an evidence that the cipher does not perform as advertised. In other words, in academic cryptography, a weakness or a break in a scheme is usually defined quite conservatively: It may require impractical amounts of time, memory, or data.

The Number of Rounds Needed for Security. It may be of interest to estimate the number of rounds needed for MiMC to be resistant against this attack. To this end, we bound the operations needed to compute all monomials of odd degree, up to a maximum degree D.

Lemma 2. *Let $1 \leq D \leq 2^n - 1$ and $x \in \mathbb{F}_{2^n}$. The overall number of operations needed to compute all odd powers x^i for $i \in [3, D]$ is given by 1 squaring and $\left\lfloor \frac{D-1}{2} \right\rfloor$ multiplications.*

Proof. From x, calculate and store $q := x^2$. The odd powers of x can now be successively computed as $x^{i+2} = x^i \cdot q$ for all odd integers i in the interval $[1, D-2]$. This yields a total of 1 squaring and $\left\lfloor \frac{D-1}{2} \right\rfloor$ multiplications. □

Assume for simplicity that $\lceil n \cdot \log_3(2) \rceil - 1$ rounds can be covered by a zero sum, and that the cost of solving the final polynomial equation is negligible. As before, we expect the time complexity to be dominated by the number of operations needed to construct the polynomial $F(K)$. Since the degree of this polynomial is upper-bounded by $3^{r_{\text{KR}}}$, by Lemma 2 at most $[(3^{r_{\text{KR}}} - 1)/2] \cdot 2^{n-1}$ multiplications are required to compute all monomials with odd exponents in $F(K)$ (where all monomials with even exponents are computed via Eq. (11)).

Since one encryption of MiMC costs $\lceil n \cdot \log_3(2) \rceil$ multiplications, the number of extra rounds ρ for MiMC must satisfy

$$(3^{\rho+1} - 1) \cdot 2^{n-2} \geq 2^n \cdot (\lceil n \cdot \log_3(2) \rceil + \rho)$$

in order to provide security against the attack just presented. This would, for example, require at least $\rho = 5$ extra rounds for $n = 129$ (more generally, if R is the number of rounds of MiMC-n/n, then $\rho \approx \lceil \log_3(2 \cdot R) \rceil$ more rounds are sufficient to restore the security[12]). We remark that *this rough estimation is not intended to replace the number of rounds proposed by the designers.*

6 An Algebraic Attack on Ciphers with Low-Degree Round Functions

Here we generalize the key-recovery attack on MiMC described in Section 5 and discuss a generic attack strategy for any block cipher working over $(\mathbb{F}_{2^n})^t$, where $n, t \in \mathbb{N}$, $t \geq 2$ and $n \geq 3$.

[12] In more details, $\rho \geq \log_3(4 \cdot (R + \rho) + 1) - 1$. The previous estimation is obtained by assuming $\rho \leq R/2$.

6.1 Setting

We consider an r-round block cipher $E_k^r : (\mathbb{F}_{2^n})^t \to (\mathbb{F}_{2^n})^t$ with

$$E_k^r(x) = (R_r \circ R_{r-1} \circ \cdots \circ R_1)(x \oplus k),$$

and where $R, R_i : (\mathbb{F}_{2^n})^t \to (\mathbb{F}_{2^n})^t$ are defined by $R_i(x) = R(x) \oplus k^{(i)}$. Here, R denominates the (nonlinear) round function. Since E_k^r consists of t components, we can write

$$E_k^r(x) = (E_{k,1}^r(x), \ldots, E_{k,t}^r(x)),$$

where $E_{k,i}^r : (\mathbb{F}_{2^n})^t \to \mathbb{F}_{2^n}$. We denote the compositional inverse of E_k^r by E_k^{-r}. We assume that

(1) the i-th round key $k^{(i)} \in (\mathbb{F}_{2^n})^t$ is derived from the master key $k = (k_1, \ldots, k_t) \in (\mathbb{F}_{2^n})^t$ by some *low-degree* (e.g., linear) key schedule,

(2) the round function R can be described by a polynomial

$$R(x = (x_1, \ldots, x_t)) = \bigoplus_{\substack{j=(j_1,\ldots,j_t)\in\{0,1,\ldots,2^n-1\}^t \\ j_1+\cdots+j_t\leq d}} \alpha_j \cdot x_1^{j_1} \cdot \cdots \cdot x_t^{j_t}$$

of *low-degree* d with coefficients $\alpha_j \in (\mathbb{F}_{2^n})^t$.

Our attack requires the symbolic evaluation of the encryption function $E_k^{r'}$ for a small number of rounds r' to be relatively easy, which motivates the requirements of a low-degree round function R and a low-degree key schedule. This ensures that the polynomial representation of $E_k^{r'}$ can be computed efficiently. In both cases, *low-degree* means *low compared to the size of the field* $\mathbb{F}_{2^n}$, i.e., $d \ll 2^n - 1$. A cipher in the literature that satisfies above assumptions and does indeed use low-degree round functions is, e.g., HadesMiMC [30].

6.2 Strategy of the Attack

The idea of our generic attack is to recover the secret master key k of a cipher E_k^r by exploiting a given higher-order distinguisher over the subset $\mathcal{X} \subseteq (\mathbb{F}_{2^n})^t$ covering $1 \leq r_{\mathrm{ZS}} < r$ rounds in the encryption or the decryption direction. For the sake of simplicity, we follow the approach of the attack on MiMC in Section 5 and assume that the higher-order distinguisher covers r_{ZS} rounds in the decryption direction.

In our attack, we symbolically evaluate $E_k^{r_{\mathrm{KR}}}(y)$ with respect to the remaining $r_{\mathrm{KR}} := r - r_{\mathrm{ZS}}$ rounds in the encryption direction and obtain polynomials $(1 \leq i \leq t)$

$$E_{(K_1,\ldots,K_t),i}^{r_{\mathrm{KR}}}(Y) \in \mathbb{F}_{2^n}[K_1, \ldots, K_t, Y_1, \ldots, Y_t]$$

over $\mathbb{F}_{2^n}$ with the master key words K_j and plaintext variables $(Y_1, \ldots, Y_t) =: Y$ as indeterminates – in short, one polynomial for each of the t components of $E_k^{r_{\mathrm{KR}}}(y)$. In general, we work with $r_{KR} \ll r_{ZS}$, since the symbolic evaluation of $E_k^{r_{\mathrm{KR}}}(y)$ is expensive.

Algorithm 3: Attack on a generic cipher E_k^r over $(\mathbb{F}_{2^n})^t$.

```
Input: Number of rounds $r$ of the cipher $E_k^r$, number of rounds $r_{\text{ZS}}$ in the
       decryption direction and a subset $\mathcal{X} \subseteq (\mathbb{F}_{2^n})^t$ satisfying the zero sum
       $\bigoplus_{x \in \mathcal{X}} E_k^{-r_{\text{ZS}}}(x) = 0$.
Output: Secret key $k = (k_1, \dots, k_t)$.
$r_{\text{KR}} \leftarrow r - r_{\text{ZS}}$.
for each $1 \leq i \leq t$ do
    Compute the symbolic evaluation
     $f_i = f_i(Y_1, \dots, Y_t, K_1, \dots, K_t) = E^{r_{\text{KR}}}_{(K_1,\dots,K_t),i}(Y_1, \dots, Y_t)$ of word $i$ in the
     encryption direction for $r_{\text{KR}}$ rounds.
    for each monomial $Y_1^{i_1} \dots Y_t^{i_t} \cdot K_1^{j_1} \dots K_t^{j_t}$ in $f_i$ with $i_1 + \dots + i_t \geq 1$ do
        $\mathscr{P}_{i_1,\dots,i_t} \leftarrow 0$.
        for each $x \in \mathcal{X}$ do
            $y = (y_1, \dots, y_t) \leftarrow E_k^{-r}(x)$, via the decryption oracle.
            $\mathscr{P}_{i_1,\dots,i_t} \leftarrow \mathscr{P}_{i_1,\dots,i_t} \bigoplus y_1^{i_1} \cdot \dots \cdot y_t^{i_t}$.
        Replace $Y_1^{i_1} \dots Y_t^{i_t} \cdot K_1^{j_1} \dots K_t^{j_t}$ with $\mathscr{P}_{i_1,\dots,i_t} \cdot K_1^{j_1} \cdot \dots \cdot K_t^{j_t}$.
    $F_i(K_1, \dots, K_t) \leftarrow f_i(K_1, \dots, K_t)$.
Find a solution $k = (k_1, \dots, k_t)$ of $F_1(k_1, \dots, k_t) = \dots = F_t(k_1, \dots, k_t) = 0$.
return $k = (k_1, \dots, k_t)$.
```

Having a zero sum after r_{ZS} rounds in the decryption direction with respect to the subset $\mathcal{X} \subseteq (F_{2^n})^t$ means that

$$\bigoplus_{x \in \mathcal{X}} E_k^{-r_{\text{ZS}}}(x) = 0.$$

The main observation behind our attack is the following: We exploit the relation[13]

$$0 = \bigoplus_{x \in \mathcal{X}} E_k^{-r_{\text{ZS}}}(x) = \bigoplus_{x \in \mathcal{X}} \left(E_k^{r_{\text{KR}}} \circ E_k^{-r}\right)(x) = \bigoplus_{y \in E_k^{-r}(\mathcal{X})} E_k^{r_{\text{KR}}}(y) \qquad (12)$$

to set up the following equations $(1 \leq i \leq t)$ over $\mathbb{F}_{2^n}$ in the variables $k_1, \dots, k_t$:

$$F_i(k_1, \dots, k_t) := \bigoplus_{y \in E_k^{-r}(\mathcal{X})} E^{r_{\text{KR}}}_{(k_1,\dots,k_t),i}(y) = 0. \qquad (13)$$

Again, $E^{r_{\text{KR}}}_{(k_1,\dots,k_t),i}(y)$ denotes the symbolic evaluation of the i-th word after r_{KR} rounds in the encryption direction with the master key words as variables $k_1, \dots, k_t$ and evaluated at $y \in \mathbb{F}_{2^n}$. Once we have set up the equation system arising from Eq. (13), we apply Gröbner basis techniques to solve this system over $\mathbb{F}_{2^n}$ for the key variables $k_1, \dots, k_t$.

In Algorithm 3 we summarize the approach of our generic attack and present a pseudo code of the attack procedure. For completeness, a rough complexity estimation of the attack is derived in Appendix E.

[13] Note that in this representation, $E_k^r = E_k^{r_{\text{ZS}}} \circ E_k^{r_{\text{KR}}}$ and $E_k^{-r_{\text{ZS}}} = E_k^{r_{\text{KR}}} \circ E_k^{-r}$.

6.3 Comparison with Related Work

Interpolation Attacks. Originally introduced as a standalone attack, interpolation attacks [35] are algebraic attacks that express the (potentially round-reduced) cipher as a polynomial equation with unknown, key-dependent coefficients, and recover these coefficients from known inputs and outputs. More recently, this approach has been combined as a key-recovery approach together with integral distinguishers.

Attack on CAST. In an attack [42] on the CAST cipher the authors use a higher-order differential distinguisher to set up an equation system and finally solve this systems for the key variables. In contrast to our attack, the authors of [42] work with linear equation systems over $\mathbb{F}_2$. While this is sufficient for CAST, working at bit level is in general more expensive than working on word level when analyzing ciphers that are natively defined at word level.

Optimized Interpolation Attacks. One type of optimized interpolation attacks was described in [23], where the authors find attacks on reduced-round versions of LowMC which are more efficient than previous attacks based on key guessing [25]. A similar attack was also used to break the full-round version of the Frit permutation in an Even–Mansour setting [26]. The overall strategy of this interpolation attack is to find a distinguisher (for example a constant sum in the encryption direction in the case of LowMC) with which one attacks the construction by finding the unknown monomials of the sums of the symbolic representations in the inverse direction. By determining these (key-dependent) monomials, the full key can eventually be found. Since the approach in [23] shares some similarities with our proposal, we describe the differences between these two strategies in detail.

The main difference regarding the two strategies concerns the way in which the system of equations $F_i(K) = 0$ is constructed and consequently solved:

- In [23], the idea is to construct the function using a "standard" interpolation technique. Specifically, the attacker does not care about the specification of the monomials of F, which are simply considered as unknowns. Hence, the idea is to recover (interpolate) the unknown coefficients of $F_K(C)$, and then use various ad-hoc techniques (which are not part of the framework described in this section) in order to recover the actual secret key.
- In our case, we heavily exploit the simple algebraic structure of the round function in order to construct the system of equations $F_i(K) = 0$. In other words, the system of equations is constructed by using a symbolic evaluation and not by interpolation techniques.

We emphasize that the possibility to set up one of the two attacks does not imply the possibility to set up the other one. For example, it seems hard to use the attack presented in [23] against full-round MiMC, while we show that our strategy can break it. Indeed, since we already need 2^{n-1} data for the distinguishing property (i.e., half of the code book), we do not see how to apply the approach from [23] to MiMC without further increasing the data complexity due to data needed for the interpolation step.

Attack on Pyjamask. Only recently, a similar attack on Pyjamask, competing in the ongoing NIST call for lightweight authenticated encryption, has been presented [27]. The authors propose an attack on the full block cipher Pyjamask-96 by combining higher-order differentials with an in-depth ad-hoc analysis of the system of equations obtained for 2.5 rounds of Pyjamask-96. As is the case for CAST, the attack is set up at bit level.

Cube Attacks. Although our attack and cube attacks [24] exploit low degrees in the polynomial description of a cipher, they are quite different from a conceptual point of view and can be regarded as two different cryptanalytic methods. To justify this conclusion, we briefly present the idea behind cube attacks and contrast them with our attack ideas.

Given a cipher with input variables $x_0, \ldots, x_{n-1}$ as the public variables (IV bits, plaintext bits, tweak bits, etc.), and $x_n, \ldots, x_{n+m-1}$ as the secret variables (key bits), the output of the cipher can be regarded as a polynomial $f = f(x)$ in $x = (x_0, \ldots, x_{n+m-1})$. For every set $I \subset \{0, \ldots, n-1\}$, f can be uniquely decomposed into

$$f = t_I \cdot f_{S(I)} + q,$$

where $t_I := \prod_{i \in I} x_i$ denotes the product of all variables indexed by elements in I, the polynomial $f_{S(I)}$ does not contain any variables from t_I, and where q misses at least one variable from t_I. The polynomial $f_{S(I)}$ is also called the *superpoly* with respect to I. For any subset $I \subseteq \{0, \ldots, n-1\}$ of size $|I|$, the authors of [24] call the set C_I of $2^{|I|}$ vectors, where all the $|I|$ variables indexed by I range over all possible combinations of elements in $\mathbb{F}_2$ and the remaining $n + m - |I|$ variables remain undetermined, a $|I|$-*dimensional Boolean cube*. Then the sum of f over all values in the cube C_I yields the equation of polynomials

$$\bigoplus_{v \in C_I} f(v) = f_{S(I)}.$$

Cube attacks consist of two steps. First, attackers recover the superpoly in the offline phase. In this phase, the attacker might need to try sufficiently many cubes and assignments for the remaining public variables such that the superpoly $f_{S(I)}$ is a balanced function of the secret variables. Moreover, determining the actual coefficients of $f_{S(I)}$ requires the additional assumption that the attacker is allowed to tweak both public and secret variables. Then, with this usable superpoly, during the online phase, the attacker leaves the secret variables undetermined and queries the encryption oracle with every value $c \in C_I$ and gets $f(c) \in \mathbb{F}$. Eventually, the attacker computes

$$f_I := \bigoplus_{c \in C_I} f(c).$$

The secret key information can be recovered by solving the corresponding equation system $f_I = f_{S(I)}$.

Compared with our attack, cube attacks involve an initial step of finding balanced superpolies that contain independent secret variables. Apart from that, cube attacks do *not* exploit the algebraic structure of a cipher, since they rely on the assumption of tweakable black box polynomials. In this sense, our attack is different, since it makes heavy use of the algebraic structure of a cipher when symbolically evaluating a certain number of rounds. Furthermore, cube attacks use the assumption that both key and plaintext variables are tweakable, while we rely on the assumption that some rounds of the cipher can be efficiently evaluated symbolically (which is why we work with low-degree round functions).

7 Concluding Remarks and Future Work

Reducing the Cost of the Attack. As shown in Appendix E, two steps – namely, (1st) the construction of the system of equations $F_i(k_1, \ldots, k_t) = 0$ for $1 \leq i \leq t$ and (2nd) solving such a system – mainly constitute the cost of the attack. In general, it could make sense to balance the costs of the two steps in order to either minimize the total cost of the attack or maximize the number of rounds that can be broken.

In more detail, consider the case in which the cost of the attack is well approximated by the cost of *constructing* the system of equations $F_i(K) = 0$. Since this cost grows with the size of the subspace $\mathcal{V}$, one strategy could be to consider a smaller subset $\mathcal{X}$.[14] Obviously, this implies in general the possibility to cover fewer rounds r_{ZS} using a higher-order distinguisher, which means that more rounds r_{KR} must be covered in general. However, the overall cost of the attack may benefit from this strategy. On the other hand, the case in which the attack cost is well approximated by the cost of *solving* the system of equations $F_i(K) = 0$ requires the opposite strategy.

Moreover, we point out that the attacks can be improved by exploiting the details of the cipher. To give a concrete example, consider the case of MiMC given in Algorithm 2: The attack and its computational complexity benefit from the fact that $F(K)$ does not depend on $\mathscr{P}_5$ or $\mathscr{P}_7$. As another example, consider the case of an SPN cipher where the round function is defined as

$$R(x = (x_1, \ldots, x_t)) = M \times (S(x_1), S(x_2), \ldots, S(x_t)),$$

where $M \in (\mathbb{F}_{2^n})^{t \times t}$ and $S : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ (here, '×' denotes matrix-vector multiplication). The cost of the attack can potentially be reduced by taking into account the fact that all monomials in the polynomial representation R depend only on a single variable x_i.

Further Generalization: Ciphers over $\mathbb{F}_p$. Finally, the attack strategy can be generalized to include ciphers over $(\mathbb{F}_p)^t$ for a prime p. This is of particular

[14] We note that we cannot adopt this strategy for MiMC since we are not able to predict the growth of the degree of MiMC^{-1}. With such an estimation, the strategy proposed here can potentially reduce the cost of the attack.

importance since many of the new applications named in the introduction (e.g., STARKs and MPC) natively work over $\mathbb{F}_p$, which means that many of the recently proposed primitives are natively constructed over $\mathbb{F}_p$. We remark that the strategy of the attack does not depend on the details of the field $\mathbb{F}$. Hence, the only thing that seems to preclude this possibility seems to be a lack of knowledge regarding efficient distinguishers over $(\mathbb{F}_p)^t$. Indeed, while it is well-known how to find a higher-order distinguisher over Boolean fields (e.g., by exploiting division property tools present in the literature [46,50,52]), the same is not yet true for prime fields.

Acknowledgements. The authors thank the anonymous reviewers for their valuable comments and suggestions. Lorenzo Grassi is supported by the European Research Council under the ERC advanced grant agreement under grant ERC-2017-ADG Nr. 788980 ESCADA. Qingju Wang is funded by the University of Luxembourg Internal Research Project (IRP) FDISC. This project has received funding from the European Union's Horizon 2020 research and innovation programme (H2020 ICT 825225: Safe-DEED) and the European Research Council (ERC 681402: SOPHIA).

References

1. Albrecht, M.R., Cid, C., Grassi, L., Khovratovich, D., Lüftenegger, R., Rechberger, C., Schofnegger, M.: Algebraic Cryptanalysis of STARK-Friendly Designs: Application to MARVELlous and MiMC. In: ASIACRYPT 2019. LNCS, vol. 11923, pp. 371–397 (2019)
2. Albrecht, M.R., Grassi, L., Perrin, L., Ramacher, S., Rechberger, C., Rotaru, D., Roy, A., Schofnegger, M.: Feistel Structures for MPC, and More. In: ESORICS 2019. LNCS, vol. 11736, pp. 151–171 (2019)
3. Albrecht, M.R., Grassi, L., Rechberger, C., Roy, A., Tiessen, T.: MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In: ASIACRYPT 2016. LNCS, vol. 10031, pp. 191–219 (2016)
4. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 430–454 (2015)
5. Aly, A., Ashur, T., Ben-Sasson, E., Dhooghe, S., Szepieniec, A.: Design of Symmetric-Key Primitives for Advanced Cryptographic Protocols. IACR Cryptology ePrint Archive, Report 2019/426 (2019)
6. Ashur, T., Dhooghe, S.: MARVELlous: a STARK-Friendly Family of Cryptographic Primitives. IACR Cryptology ePrint Archive, Report 2018/1098 (2018)
7. Aumasson, J.P., Meier, W.: Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi (2009), presented at the Rump Session of CHES 2009, `https://131002.net/data/papers/AM09.pdf`
8. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable, transparent, and post-quantum secure computational integrity. IACR Cryptology ePrint Archive **2018**, 46 (2018)
9. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: On the indifferentiability of the sponge construction. In: EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197 (2008)

10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Note on zero-sum distinguishers of Keccak-f, `http://keccak.noekeon.org/NoteZeroSum.pdf`
11. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: CRYPTO. LNCS, vol. 537, pp. 2–21. Springer (1990)
12. Black, J., Rogaway, P., Shrimpton, T.: Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In: CRYPTO 2002. LNCS, vol. 2442, pp. 320–335 (2002)
13. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique cryptanalysis of the full AES. In: ASIACRYPT 2011. LNCS, vol. 7073, pp. 344–371 (2011)
14. Bogdanov, A., Rijmen, V.: Linear hulls with correlation zero and linear cryptanalysis of block ciphers. Des. Codes Cryptogr. **70**(3), 369–383 (2014), see also: Cryptology ePrint Archive, Report 2011/123
15. Bogdanov, A., Wang, M.: Zero Correlation Linear Cryptanalysis with Reduced Data Complexity. In: FSE 2012. LNCS, vol. 7549, pp. 29–48 (2012)
16. Bonnetain, X.: Collisions on Feistel-MiMC and univariate GMiMC. IACR Cryptology ePrint Archive **2019**, 951 (2019)
17. Boura, C., Canteaut, A.: On the Influence of the Algebraic Degree of F^{-1} on the Algebraic Degree of G ∘ F. IEEE Trans. Information Theory **59**(1), 691–702 (2013)
18. Boura, C., Canteaut, A., De Cannière, C.: Higher-Order Differential Properties of Keccak and *Luffa*. In: FSE 2011. LNCS, vol. 6733, pp. 252–269 (2011)
19. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: IEEE Symposium on Security and Privacy. pp. 315–334. IEEE Computer Society (2018)
20. Canteaut, A., Videau, M.: Degree of Composition of Highly Nonlinear Functions and Applications to Higher Order Differential Cryptanalysis. In: EUROCRYPT 2002. LNCS, vol. 2332, pp. 518–533 (2002)
21. Cantor, D.G., Kaltofen, E.: On Fast Multiplication of Polynomials over Arbitrary Algebras. Acta Inf. **28**(7), 693–701 (1991)
22. Carlet, C., Charpin, P., Zinoviev, V.A.: Codes, bent functions and permutations suitable for DES-like cryptosystems. DCC **15**(2), 125–156 (1998)
23. Dinur, I., Liu, Y., Meier, W., Wang, Q.: Optimized Interpolation Attacks on LowMC. In: ASIACRYPT 2015. LNCS, vol. 9453, pp. 535–560 (2015)
24. Dinur, I., Shamir, A.: Cube Attacks on Tweakable Black Box Polynomials. In: EUROCRYPT 2009. LNCS, vol. 5479, pp. 278–299 (2009)
25. Dobraunig, C., Eichlseder, M., Mendel, F.: Higher-Order Cryptanalysis of LowMC. In: ICISC 2015. LNCS, vol. 9558, pp. 87–101 (2015)
26. Dobraunig, C., Eichlseder, M., Mendel, F., Schofnegger, M.: Algebraic cryptanalysis of variants of Frit. In: SAC 2019. LNCS, vol. 11959, pp. 149–170 (2019)
27. Dobraunig, C., Rotella, Y., Schoone, J.: Algebraic and Higher-Order Differential Cryptanalysis of Pyjamask-96. IACR Transactions on Symmetric Cryptology **2020**(1), 289–312 (2020)
28. von zur Gathen, J., Gerhard, J.: Modern Computer Algebra (3. ed.). Cambridge University Press (2013)
29. Grassi, L., Kales, D., Khovratovich, D., Roy, A., Rechberger, C., Schofnegger, M.: Starkad and Poseidon: New Hash Functions for Zero Knowledge Proof Systems. Cryptology ePrint Archive, Report 2019/458 (2019)
30. Grassi, L., Lüftenegger, R., Rechberger, C., Rotaru, D., Schofnegger, M.: On a Generalization of Substitution-Permutation Networks: The HADES Design Strategy. In: EUROCRYPT 2020. LNCS, vol. 12106, pp. 674–704 (2020)

31. Grassi, L., Rechberger, C., Rotaru, D., Scholl, P., Smart, N.P.: Mpc-friendly symmetric key primitives. In: ACM Conference on Computer and Communications Security. pp. 430–443. ACM (2016)
32. Guo, J., Liao, G., Liu, G., Liu, M., Qiao, K., Song, L.: Practical Collision Attacks against Round-Reduced SHA-3. Journal of Cryptology **33**(1), 228–270 (2020)
33. Hao, Y., Leander, G., Meier, W., Todo, Y., Wang, Q.: Modeling for three-subset division property without unknown subset - improved cube attacks against Trivium and Grain-128AEAD. In: EUROCRYPT 2020. LNCS, vol. 12105, pp. 466–495. Springer (2020)
34. Hu, K., Wang, M.: Automatic Search for a Variant of Division Property Using Three Subsets. In: CT-RSA 2019. LNCS, vol. 11405, pp. 412–432 (2019)
35. Jakobsen, T., Knudsen, L.R.: The interpolation attack on block ciphers. In: FSE. LNCS, vol. 1267, pp. 28–40 (1997)
36. Khovratovich, D., Leurent, G., Rechberger, C.: Narrow-bicliques: Cryptanalysis of full IDEA. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 392–410 (2012)
37. Knudsen, L.R.: Truncated and Higher Order Differentials. In: FSE 1994. LNCS, vol. 1008, pp. 196–211 (1994)
38. Knudsen, L.R., Rijmen, V.: Known-Key Distinguishers for Some Block Ciphers. In: ASIACRYPT 2007. LNCS, vol. 4833, pp. 315–324 (2007)
39. Lai, X.: Higher Order Derivatives and Differential Cryptanalysis, pp. 227–233 (1994)
40. Li, C., Preneel, B.: Improved Interpolation Attacks on Cryptographic Primitives of Low Algebraic Degree. In: SAC 2019. LNCS, vol. 11959, pp. 171–193 (2019)
41. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397 (1993)
42. Moriai, S., Shimoyama, T., Kaneko, T.: Higher order differential attack of a CAST cipher. In: FSE. LNCS, vol. 1372, pp. 17–31. Springer (1998)
43. Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: Nearly practical verifiable computation. In: IEEE Symposium on Security and Privacy. pp. 238–252. IEEE Computer Society (2013)
44. Rotaru, D., Smart, N.P., Stam, M.: Modes of Operation Suitable for Computing on Encrypted Data. IACR Trans. Symmetric Cryptol. **2017**(3), 294–324 (2017)
45. Todo, Y.: Structural Evaluation by Generalized Integral Property. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 287–314 (2015)
46. Todo, Y., Isobe, T., Hao, Y., Meier, W.: Cube Attacks on Non-Blackbox Polynomials Based on Division Property. In: CRYPTO 2017. LNCS, vol. 10403, pp. 250–279 (2017)
47. Todo, Y., Morii, M.: Bit-Based Division Property and Application to Simon Family. In: FSE 2016. LNCS, vol. 9783, pp. 357–377 (2016)
48. Vielhaber, M.: Breaking ONE.FIVIUM by AIDA an algebraic IV differential attack. IACR Cryptology ePrint Archive **2007**, 413 (2007)
49. Wang, Q., Grassi, L., Rechberger, C.: Zero-Sum Partitions of PHOTON Permutations. In: CT-RSA 2018. LNCS, vol. 10808, pp. 279–299 (2018)
50. Wang, Q., Hao, Y., Todo, Y., Li, C., Isobe, T., Meier, W.: Improved Division Property Based Cube Attacks Exploiting Algebraic Properties of Superpoly. In: CRYPTO 2018. LNCS, vol. 10991, pp. 275–305 (2018)
51. Wang, Q., Liu, Z., Varici, K., Sasaki, Y., Rijmen, V., Todo, Y.: Cryptanalysis of Reduced-Round SIMON32 and SIMON48. In: INDOCRYPT 2014. LNCS, vol. 8885, pp. 143–160 (2014)
52. Wang, S., Hu, B., Guan, J., Zhang, K., Shi, T.: MILP-aided Method of Searching Division Property Using Three Subsets and Applications. In: ASIACRYPT 2019. LNCS, vol. 11923, pp. 398–427 (2019)

SUPPLEMENTARY MATERIAL

Scripts and Implementations

The MAGMA script *Magma_Script_MiMC_Univariate_Attack* has two input parameters: N and *version*. N is an odd integer that decides the block size of MiMC, i.e., MiMC-N/N. The second parameter $version \in \{1, 2\}$ determines whether to use Algorithm 1 or Algorithm 2. The script creates an instance of MiMC-N/N, and runs a key-recovery attack using the chosen algorithm. It outputs the roots of $F(K)$, as well as the secret key k for comparison.

We also provide the file *zero_sum_tester.cpp*, which contains the code we used to find the zero sums for MiMC. It accepts three parameters: the field size, the number of rounds, and the dimension of the vector space.

A Key-Alternating Ciphers with $\delta = 2$: Lower Bound of $\mathcal{R}_{\text{gen}}$

In order to compare our theoretical results (namely, $\mathcal{R}_{\text{lin}}$) with the ones already known in the literature (namely, $\mathcal{R}_{\text{gen}}$), we first provide a lower bound of $\mathcal{R}_{\text{gen}}$ – similar to the one provided in Eq. (3) – for this specific case.

Lemma 3. *Let $n \geq 3$. Under the assumption of Proposition 1, let S be an S-box on $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ of algebraic degree 2, and let γ be defined as in Proposition 1. First of all, $\gamma \leq \frac{n+1}{2}$. Moreover, in the case in which the function $F : \mathbb{F}_2^N \to \mathbb{F}_2^N$ for $N = n \cdot t$ is the concatenation of t S-boxes just defined, then for any function G from $\mathbb{F}_2^N$ to $\mathbb{F}_2^N$*

$$\deg(G \circ F) \leq \min\left\{\deg(G) \cdot \deg(F), N - 2 \times \frac{N - \deg(G)}{n+1}\right\}. \tag{14}$$

Proof. By definition, note that $\delta_i \leq 2i$ and that $\delta_i \leq n - 1$ for each i. Since $2i \leq n - 1$ if $i \leq (n-1)/2$, it follows that

$$\gamma = \max_{1 \leq i \leq n-1} \binom{n-i}{n-\delta_i} \leq \max\left\{\max_{1 \leq i \leq \frac{n-1}{2}} \binom{n-i}{n-2i}; n - \frac{n-1}{2} - 1\right\} = \frac{n+1}{2},$$

where $\max_{\frac{n-1}{2}+1 \leq i \leq n-1} \binom{n-i}{n-\delta_i} = n - \frac{n-1}{2} - 1$. The bound given in Eq. (14) is obtained by replacing γ with $(n+1)/2$ in Eq. (3). □

By experiments and working on the cube S-box $S(x) = x^3$, we found that $\gamma = \frac{n+1}{2}$ for each odd $n \leq 33$. For this reason, we conjecture the following.

Conjecture 1. For the cube S-box $S(x) = x^3 : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$, we conjecture that γ is always equal to $\frac{n+1}{2}$ for every (odd) n.

Lastly, we have already noted in Eq. (8) that two rounds of MiMC have algebraic degree 2. Using this observation, along with Lemma 3, we derive an iterative upper bound, $\delta_{\text{gen}}(r)$, for r rounds of MiMC$-n/n$. Define $\delta_{\text{gen}}(1) = 2$, and for $r \geq 2$ the following holds:

$$\delta_{\text{gen}}(r) = \min\left\{ \varDelta(r), \left\lfloor n - 2 \times \frac{n - \delta_{\text{gen}}(r-1)}{n+1} \right\rfloor \right\},$$

$$\varDelta(r) = \begin{cases} 2\delta_{\text{gen}}(r-1) & \text{when } r \text{ is an odd integer,} \\ \delta_{\text{gen}}(r-1) & \text{when } r \text{ is an even integer.} \end{cases}$$

This bound is used in Fig. 2 and Table 2.

B Algebraic Degree Growth of MiMC^{-1}

While not needed for our attack, we also analyzed the degree growth of MiMC in the decryption direction. The results of the tests we applied and the size of the vector space dimensions necessary for higher-order distinguishers are shown in Table 3.

As we can see, the algebraic degree does not increase in the second round for the instances we tested, and after that it starts growing slowly. Moreover, it seems to remain consistent after roughly half the number of rounds, until it finally reaches its maximum in the final round.

r' n	2	3	4	5	6	7	8	9	10	11	12
7	5	6	6	–	–	–	–	–	–	–	–
9	6	7	8	8	–	–	–	–	–	–	–
11	7	9	9	10	10	–	–	–	–	–	–
13	8	10	11	11	12	12	12	–	–	–	–
15	9	11	12	13	14	14	14	14	–	–	–
17	10	13	14	15	16	16	16	16	16	–	–
19	11	14	15	16	17	18	18	18	18	18	–

Table 3: Data complexities necessary for zero sums when evaluating MiMC in the decryption direction for various block sizes and round numbers.

C Division Property and Automatic Tools

The division property [45] – proposed as a generalization of integral and higher-order differential distinguishers at Eurocrypt 2015 – has already been applied to find new generic distinguishers for both SPN and Feistel constructions. However, the original division property treats the round function at word level, therefore by its nature some propagation information cannot be captured. In this paper, we evaluate some of our practical cases by mixed integer linear programming (MILP) based on the bit-based division property, namely, two-subset bit-based division property and three-subset bit-based division property.

We first introduce some notations for bit vectors. For any n-bit vector $\vec{x}$ and $0 \leq i \leq n-1$, we denote x_i as its i-th bit. Given two n-bit vectors $\vec{u}$ and $\vec{x}$, we define $\pi_{\vec{u}}(\vec{x}) = \mathit{\Pi}_{i=0}^{n-1} x_i^{u_i}$. Moreover, $\vec{u} \succeq \vec{k}$ denotes $u_i \geqslant k_i$ for all i.

C.1 Two-Subset Bit-Based Division Property

Todo and Morii [47] first introduced the bit-based division property to investigate integral characteristics at a bit level for block ciphers. In the bit-based division property, two cases are considered where $\vec{u}$ can be classified into two sets, which is therefore called the two-subset bit-based division property (`2-Subset-BDP`), according to which the parity of $\pi_{\vec{u}}(\vec{x})$ is even or unknown. The definition of the two-subset bit-based division property is as follows.

Definition 2 (Two-Subset Bit-Based Division Property [47]). *Let $\mathbb{X}$ be a multiset of n-bit vectors, and $\mathbb{K}$ be a set of n-bit vectors. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}_{\mathbb{K}}^{1^n}$, it fulfills the following conditions:*

$$\bigoplus_{\vec{x} \in \mathbb{X}} \pi_{\vec{u}}(\vec{x}) = \begin{cases} \mathit{unknown}, & \text{if } \exists\ \vec{k} \in \mathbb{K}\ \mathit{s.t.}\ \vec{u} \succeq \vec{k}, \\ 0, & \text{otherwise.} \end{cases}$$

Our Practical Results with `2-Subset-BDP`.

“Small” S-Box. We model the S-boxes of size 5 and 7 by a set of linear inequalities as in [64]. Given an S-box, we first compute a set of vectors A (also called division trail table) that is composed of all pairs of division property input and output of the S-box, and then calculate the H-Representation of the convex hull of A, which is a set of linear inequalities that can describe the vectors of the S-box accurately, by using the `inequality_generator()` in SageMath,[15].A greedy algorithm is usually applied to reduce the number of inequalities in order to speedup the process of the MILP.

Based on this method, we add 21 and 1216 inequalities respectively for the 5-bit and 7-bit cube S-boxes in Table 2 to the MILP model. After calling MILP solvers, we find a 2-round zero-sum property for them, i.e., this MILP-aided evaluation only provides us a lower bound of 3 rounds that are necessary to prevent higher-order distinguishing attacks. However, the practical results for $n = 5, 7$ are $4, 5$ rounds, which refutes the commonly believed fact that one can always find the best integral distinguisher using `2-Subset-BDP` for block ciphers, even when not taking the secret keys into consideration.

[15] `https://www.sagemath.org`

"Big" S-Box. Generating linear inequalities for the H-representation of the convex hull, often by using SageMath, requires an exponential complexity in the number of input and output pairs. In the case of our 9-bit cube S-box, in order to build the MILP model, 15612 inequalities are added. Such a large number of linear equations make the whole MILP quite heavy for the off-shell optimization solvers, which might eventually result in out-of-memory errors. In [60], an exhaustive list of compact representations in logical condition modeling against 4-bit S-boxes was proposed, and in [53], the Quine-McCluskey and the Espresso algorithm was applied to generate inequalities for 8-bit S-boxes. Unfortunately neither method is helpful for applications with S-boxes much larger than 8 bits (e.g., 129 bits).

We choose to model larger an S-box by modeling `COPY`, `AND`, and `XOR` operations in the ANF following rules as given in [64]. The bounds obtained by this automatic tool are far worse than the practical results, actually they are even worse than the bounds retrieved by `2-Subset-BDP`. Besides the hereditary inaccuracy of `2-Subset-BDP`, this unpleasant gap originates from the way we model the S-box that easily inserts a large amount of invalid division trails to the solution pool, which leads to a quicker loss of the zero-sum property than the cipher would.

C.2 Three-Subset Bit-Based Division Property

Three-subset bit-based division property (`3-Subset-BDP`) [47], where $\vec{u}$ is divided into three sets: "0", "1" and "unknown", is proposed to enhance `2-Subset-BDP` for detecting more features. [34,52]. A formal definition is given as follows:

Definition 3. *(Three-Subset Bit-Based Division Property [47]). Let $\mathbb{X}$ be a multiset of n-bit vectors. Let $\mathbb{K}$ and $\mathbb{L}$ be two sets of n-bit vectors. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}^{1^n}_{\mathbb{K},\mathbb{L}}$, it fulfills the following conditions:*

$$\bigoplus_{\vec{x}\in\mathbb{X}} \pi_{\vec{u}}(\vec{x}) = \begin{cases} \textit{unknown}, & \text{if } \exists\, \vec{k} \in \mathbb{K} \ s.t.\ \vec{u} \succeq \vec{k}, \\ 1, & \text{else if } \exists\, \vec{\ell} \in \mathbb{L} \ s.t.\ \vec{u} = \vec{\ell}, \\ 0, & \text{otherwise.} \end{cases}$$

Algorithms that can model this more precise version of the division property efficiently should enable us to achieve better results compared with the one obtained by the `2-Subset-BDP`. However, when applied to block ciphers where secret keys are added, it raises phenomenons that counter the intuitive.

Influence of Involving Secret Keys. In the key-independent setting, $\mathbb{K}$ and $\mathbb{L}$ can be processed independently. However, when secret round keys are added to the intermediates, vectors in $\mathbb{L}$ will affect the propagation of vectors in $\mathbb{K}$. A rule in [47] was proposed to handle the propagation: Assuming a round key is xored with the i-th bit, then for all $\vec{\ell} \in \mathbb{L}$ satisfying $\ell_i = 0$, a new vector $(\ell_1, \ell_2, \ldots, \ell_i \vee 1, \ldots, \ell_n)$ is appended to $\mathbb{K}$. This propagation rule evokes the problem which is called *unknown-producing problem* in [33].

Influence of Focusing on Single Trail. Another important propagation rule is XOR rules for calculating vectors in $\mathbb{L}'$ from $\mathbb{L}$. If $\vec{\ell}$ is not included in $\mathbb{L}$ before, then it is inserted to $\mathbb{L}'$; otherwise it is removed from $\mathbb{L}'$. This XOR rule results in the problem which is called *cancellation problem* in [33].

C.3 Three-Subset Bit-Based Division Property without the Unknown Subset

According to the propagation rules for the 3-Subset-BDP, the unknown-producing problem implies that all the vectors in $\mathbb{L}_i$ should be determined when the secret key is xored, and the cancellation problem implies that focusing only on one single trail is not enough. Furthermore, after iterating i rounds, the amount of vectors in $\mathbb{K}_i$ and $\mathbb{L}_i$ explodes, which makes it harder to trace the propagation of 3-Subset-BDP.

Motivated by modeling 3-Subset-BDP efficiently, the *variant three-subset division property* [34] was proposed to handle the unknown-producing problem. However, the cancellation problem is ignored in their model, which made their results worse than the ones by 3-Subset-BDP. In [52], the breadth-first search algorithm and the pruning technique were combined to model the 3-Subset-BDP. As a result, it guarantees that the sizes of $\mathbb{K}_i$ and $\mathbb{L}_i$ decrease dramatically. However, the pruning technique is useful only when the size of $\mathbb{L}_i$ is reasonably small, which heavily limits its applications.

To overcome these problems and model the 3-Subset-BDP efficiently, recently, [33] proposed a model formulating the 3-Subset-BDP without the unknown subset[16].

Definition 4. *(Modified Three-Subset Bit-Based Division Property [33]). Let $\mathbb{X}$ and $\mathbb{L}$ be multisets of n-bit vectors. When $\mathbb{X}$ has the division property $\mathcal{D}_{\mathbb{L}}^{1^n}$, it fulfills the following conditions:*

$$\bigoplus_{\vec{x}\in\mathbb{X}} \pi_{\vec{u}}(\vec{x}) = \begin{cases} 1, & \text{if there are odd number of } \vec{u}\text{'s in } \mathbb{L}, \\ 0, & \text{otherwise.} \end{cases}$$

In this new model, $\mathbb{L}$ is a multiset. When undertaking the propagation of bit vectors, we count the number of bit vectors in $\mathbb{L}$. Accordingly, the propagation rules are slightly modified to guarantee the propagation of vectors for the multiset. More details can be found in [33].

Our Practical Results with 3-Subset-BDP. We build MILP models for the modified 3-Subset-BDP for our practical experiments for cases of MiMC with an n-bit S-box, where $n \in \{5, 7, 9\}$. We obtain exactly the same results with the practical ones in Table 2. Therefore, we conclude that by modeling the modified 3-Subset-BDP with the help of MILP automatic tools, we can evaluate

[16] The idea of handling the cancellation problem is mentioned in [52], but it is not utilized in their MILP models.

an accurate bound resistant to higher-order distinguishing attacks for MiMC with “small” S-boxes.

However, as far as we know, there are no efficient methods to model a larger S-box with the (modified) `3-Subset-BDP`. Thus, our $\mathcal{R}_{\text{lin}}$ bound derived in this paper can evaluate S-boxes of any size, and give a bound very close to the practical result by experiments.

D Multivariate Attack Approach for MiMC

In this section, we consider attacking MiMC by solving a system of equations over $\mathbb{F}_2$. We will thus have n key variables. While this approach leads to a less efficient attack on MiMC when compared to our main approach described in Section 5, it may be useful for other cryptographic constructions which work only over $\mathbb{F}_2$.

D.1 Generating Low-Degree Equations in the Key Bits

Our goal is to find the key bits by solving a system of n key variables in n polynomials over $\mathbb{F}_2$. Only for simplicity, we focus on the instances where we can choose $r_{\text{KR}} = 1$ rounds of encryption. In order to build this system, we evaluate MiMC in the encryption direction over one single round symbolically, where we keep the key bits as variables and where we use the concrete values obtained by the oracle for the input bits.

This step results in n sums of 2^{n-1} values, where each sum is a degree-1 polynomial over $\mathbb{F}_2$ in the variables $k_1, k_2, \ldots, k_n$. This is the case because all monomials p_i, $p_i \cdot p_j$ for $i \neq j$, and $k_i \cdot k_j$ for $i \neq j$, where $i \in [1, n], j \in [1, n]$, are removed after substitution and summation. The remaining monomials $p_i \cdot k_j$, where $i \in [1, n], j \in [1, n]$, are linear in the key bits after substitution.

D.2 Solving a System of n Linear Equations in n Variables

Since we know that the sum in each bit after one single round is 0 due to the number of chosen ciphertexts, our equation system has the following structure:

$$\begin{cases} f_1(k_1, k_2, \ldots, k_n) = 0 \\ f_2(k_1, k_2, \ldots, k_n) = 0 \\ \vdots \\ f_n(k_1, k_2, \ldots, k_n) = 0 \end{cases}$$

where each $f_i : (\mathbb{F}_2)^n \to \mathbb{F}_2$ is a degree-1 polynomial. As shown in the following, the complexity of solving such a system of n linear equations in n variables can be given as the complexity of Gaussian elimination, which is

$$T_3 \in \mathcal{O}(n^3)$$

bit operations, and thus well within the allowed time frame for the attack.

n	Time	Data
33	$2^{28.61}$	2^{n-1}
63	$2^{57.68}$	2^{n-1}
193	$2^{186.07}$	2^{n-1}
255	$2^{247.67}$	2^{n-1}
513	$2^{504.66}$	2^{n-1}

Table 4: Attack complexities when using the multivariate approach.

Low-Degree Polynomial. Here we briefly analyze the cost of solving a polynomial system over $\mathbb{F}_2^n$ of algebraic degree d. For $d = 1$, this system is linear and can be solved in a number of bit operations in $\mathcal{O}(n^3)$ with Gaussian elimination. If $d > 2$, the best strategy may be to solve the system using a dedicated brute-force algorithm, as presented in [56]. For optimal choices of algorithm parameters[17], this is expected to require $4d \cdot \log(n) \cdot 2^n$ bit operations. In many instances, it may therefore be less costly to brute-force the polynomial system in this way than brute-forcing the encryption system directly. Lastly, techniques of solving quadratic polynomial systems (i.e., $d = 2$) have received extensive study from the cryptographic community. Under some assumptions on the polynomial system, [55] estimates the asymptotic time complexity of this problem to be in $\mathcal{O}\left(2^{0.841n}\right)$.

D.3 Summary of the Attack

In total, following steps are necessary.

1. *(Online)* Request the decryptions of 2^{n-1} chosen ciphertexts.
2. *(Offline)* For each of the obtained plaintexts, evaluate a single round of MiMC in the encryption direction and keep the key bits as variables.
3. *(Offline)* Solve the resulting system of n linear equations in n unknown key variables.

D.4 Attack Complexity

Note that since the algebraic degree of one round is only 2, we can obtain at most n different monomials for each bit position (namely, degree-1 monomials in the key bits) if we directly substitute the plaintext bits with the concrete values obtained from our oracle. Since we can therefore omit the computation of all monomials of the form $k_i \cdot k_j$, where $i \neq j$ and $i \in [1, n], j \in [1, n]$, the symbolic evaluation of a single round of MiMC is similarly expensive as the direct evaluation, and we approximate this complexity by n^2. Building the sums adds

[17] Here, we mean optimality with respect to the time complexity. In practice, the authors note that the optimal choice depends on the available hardware (see [56, Sect. 5]).

an additional $\leq n^2$ bit operations, and due to the number of input vectors we thus arrive at a total complexity of

$$\mathcal{C}_{\mathcal{A}} \leq 2^{n-1}\left(2n^2\right)$$

bit operations. Optimistically assuming[18] that we need only n^2 bit operations for a direct evaluation of $f(x) = x^3$, the cost of exhaustively searching for the correct key is around

$$\mathcal{C}_{\mathcal{E}} = 2^n \cdot \left(n^2 \cdot \left\lceil \frac{n}{\log_2(3)} \right\rceil\right)$$

bit operations, and $\mathcal{C}_{\mathcal{A}} < \mathcal{C}_{\mathcal{E}}$.

Finally, the number of chosen ciphertexts required for the zero sum results in a data complexity of 2^{n-1}, and the memory complexity is negligible at n^2, both for the symbolic evaluations and for the final solving step involving an $n \times n$ matrix over $\mathbb{F}_2$. The final complexities are shown in Table 4.

E (Rough) Complexity Estimations of the Attack Proposed in Section 6

For our complexity estimations of the attack proposed in Section 6, we count finite field operations over $\mathbb{F}_{2^n}$. We consider multiplications and squarings separately, since the squaring operation is an $\mathbb{F}_2$-linear operation in fields of characteristic 2.

As is the case for the attack on MiMC in Section 5, the generic attack strategy is composed of two steps. First, we construct the system of equations $F_i(k_1, \ldots, k_t) = 0$ for $1 \leq i \leq t$, and then we solve this system over $\mathbb{F}_{2^n}$ for $k_1, k_2, \ldots, k_t$. We recall that the cost of the first step grows with the size of $\mathcal{X}$, the subset needed for a zero sum. Since estimating the complexity for these steps more precisely would require a thorough analysis of the particular polynomial system in question, in the following we briefly describe these two steps without going into all the details an attacker could potentially exploit.

E.1 Setting Up the Equation System

For the equation system, we first need to symbolically evaluate r_{KR} encryption rounds, which results in t polynomials

$$E^{r_{\text{KR}}}_{(K_1,\ldots,K_t),i}(Y_1, \ldots, Y_t), \quad 1 \leq i \leq t,$$

of degree $D = D(r_{\text{KR}})$ over $\mathbb{F}_{2^n}$ in variables $K_1, \ldots, K_t$ and $Y_1, \ldots, Y_t$. Every monomial $Y_1^{i_1} \cdots Y_t^{i_t}$ in any polynomial $E^{r_{\text{KR}}}_{(K_1,\ldots,K_t),i}(Y_1, \ldots, Y_t)$ needs to be replaced by

$$\mathscr{P}_{i_1,\ldots,i_t} := \bigoplus_{y=(y_1,\ldots,y_t) \in E_k^{-r}(\mathcal{X})} y_1^{i_1} \cdot \cdots \cdot y_t^{i_t},$$

[18] We ignore the cost for key additions and constant additions, as well as the memory (or additional computation time) needed to store (or compute) the round constants.

leaving us with t polynomials in the key variables $K_1, \ldots, K_t$ as indeterminates. Here we need an estimation for computing all $\mathscr{P}_{i_1,\ldots,i_t}$, or equivalently to write down a system of equations of the form as in Eq. (13).

For $t = 1$, the number of multiplications and squarings needed was stated in Lemma 2. The situation is more complicated for $t \geq 2$, since several strategies can be used to compute the monomials and minimize the number of multiplications, the number of squarings, or the memory cost. Since this depends on the details of the considered primitives, in the following we limit ourselves to present a high-level analysis of two extreme cases, namely $n = 1$ (which corresponds e.g. to LowMC) and $n \geq 3$ and $D \leq 2^n - 1$ (which corresponds e.g. to HadesMiMC).

The Number of Needed Multiplications for Specific Cases. Here we limit ourselves to analyze two extreme cases, namely

(1) $n = 1$;
(2) $n \geq 3$ and $D \leq 2^n - 1$.

In the first case, the number of multiplications can be upper-bounded by the number of different monomials, namely $\sum_{i=1}^{D} \binom{t}{i}$ where $D < t$. This is done through successive multiplications by degree, i.e., every monomial of degree d can be computed by combining a monomial of degree $d-1$ with a single multiplication.

In the second case, we propose the following Lemma.

Lemma 4. *Let D be an integer with $1 \leq D \leq 2^n - 1$. The number of non-squaring based multiplications needed to compute all monomials of total degree at most D in t variables over $\mathbb{F}_{2^n}$ is upper-bounded by*

$$\left(\sum_{i=2}^{D} \binom{i+t-1}{t-1} \right) - t \cdot \frac{D-1}{2}$$

Proof. It is a well-known fact that the number of different monomials of degree d in t variables is $M_d = \binom{d+t-1}{t-1}$. As above we use a successive multiplication, where every monomial of degree d can be computed by combining a monomial of degree $d-1$ with a single multiplication. It follows that the number of multiplications needed to compute all monomials in at least two variables up to degree D is upper-bounded by

$$M = \sum_{i=2}^{D} (M_i - t) = \left(\sum_{i=2}^{D} \binom{i+t-1}{t-1} \right) - (D-1)t.$$

Lastly, we add the t univariate monomials to M, which by Lemma 2 amounts to at most $t \cdot \frac{D-1}{2}$ multiplications. □

We note that a monomial with all univariate degrees being even can be generated by squaring a lower-degree monomial. This fact is not considered in Lemma 4 since such a squaring is counted as a non-squaring-based multiplication.

Hence, there may be a different tradeoff between squarings and non-squaring-based multiplications when counting the number of multiplications for computing all monomials of total degree at most D in t variables. Potentially, the tradeoff may be improved in favour of squarings when dealing with a concrete cipher.

E.2 Complexity Estimation for Solving the Equation System

For $t > 1$, the resulting equation system is a multivariate polynomial system. If we additionally have $n > 1$, the standard strategy for finding the solutions of such systems[19] is through a Gröbner basis [57]. Such an attack essentially consists of first computing a Gröbner basis in *degrevlex* order, then converting it to the *lex* order, and finally factorizing a univariate polynomial in this basis and back-substituting its roots. It is in general a hard problem to estimate the complexity needed for these steps. As largely done in the literature, we assume that the most expensive step is the first one (i.e., computing a Gröbner basis in *degrevlex* order). For generic systems, the complexity of this step for a system of $\mathfrak{N}$ polynomials f_i in $\mathfrak{V}$ variables is $\mathcal{O}\left(\binom{\mathfrak{V}+D_{\text{reg}}}{D_{\text{reg}}}^{\omega}\right)$ operations over the base field $\mathbb{F}$, where D_{reg} is the *degree of regularity* [54] and $2 \leq \omega < 3$ is the linear algebra constant. The degree of regularity depends on the number of polynomials $\mathfrak{N}$, their degrees d_i, as well as the algebraic structure of the system. Closed-form formulas for D_{reg} are only known for some special cases. For example, if $\mathfrak{V} = \mathfrak{N}$ (namely, the case considered in this attack), a simple closed form is given by $D_{\text{reg}} = 1 + \sum_{i=0}^{\mathfrak{N}-1}(d_i - 1)$. We remark that this is a pessimistic upper bound. Indeed, the algebraically simple ciphers we are considering may exhibit more algebraic structure than what is the case for generic systems.

Supplementary Material References

53. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics. IACR Trans. Symmetric Cryptol. **2017**(4), 99–129 (2017)
54. Bardet, M., Faugere, J., Salvy, B., Yang, B.: Asymptotic behaviour of the index of regularity of quadratic semi-regular polynomial systems. In: The Effective Methods in Algebraic Geometry Conference (MEGA). pp. 1–14 (2005)
55. Bardet, M., Faugère, J., Salvy, B., Spaenlehauer, P.: On the complexity of solving quadratic boolean systems. J. Complexity **29**(1), 53–75 (2013)
56. Bouillaguet, C., Chen, H., Cheng, C., Chou, T., Niederhagen, R., Shamir, A., Yang, B.: Fast Exhaustive Search for Polynomial Systems in F_2. In: CHES 2010. LNCS, vol. 6225, pp. 203–218 (2010)
57. Cox, D.A., Little, J., O'Shea, D.: Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.). Undergraduate texts in mathematics, Springer (1997)
58. Hao, Y., Leander, G., Meier, W., Todo, Y., Wang, Q.: Modeling for three-subset division property without unknown subset - improved cube attacks against Trivium

[19] Strategies that involve guessing (parts of) the variables may be more viable over $\mathbb{F}_2$.

and Grain-128AEAD. In: EUROCRYPT 2020. LNCS, vol. 12105, pp. 466–495. Springer (2020)

59. Hu, K., Wang, M.: Automatic Search for a Variant of Division Property Using Three Subsets. In: CT-RSA 2019. LNCS, vol. 11405, pp. 412–432 (2019)
60. Sasaki, Y., Todo, Y.: New Impossible Differential Search Tool from Design and Cryptanalysis Aspects - Revealing Structural Properties of Several Ciphers. In: EUROCRYPT 2017. LNCS, vol. 10212, pp. 185–215 (2017)
61. Todo, Y.: Structural Evaluation by Generalized Integral Property. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 287–314 (2015)
62. Todo, Y., Morii, M.: Bit-Based Division Property and Application to Simon Family. In: FSE 2016. LNCS, vol. 9783, pp. 357–377 (2016)
63. Wang, S., Hu, B., Guan, J., Zhang, K., Shi, T.: MILP-aided Method of Searching Division Property Using Three Subsets and Applications. In: ASIACRYPT 2019. LNCS, vol. 11923, pp. 398–427 (2019)
64. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers. In: ASIACRYPT 2016. LNCS, vol. 10031, pp. 648–678 (2016)

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

uib.no

ISBN: 9788230859629 (print)
9788230868416 (PDF)