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 $X \mapsto X^{1+q^\Theta}$ over $\mathbb{F}_{q^d}$.

These mappings are combined together into P' as follows

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

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

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

and

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

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.

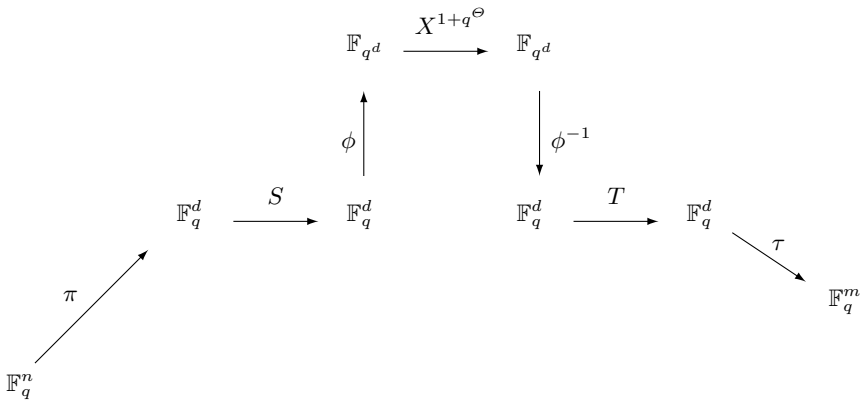

Fig. 1: Diagram of EFLASH mappings.

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:

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

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* [1]. 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 [2]:

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

From experiments it seems to be the case that randomly generated polynomial systems will behave as semi–regular sequences [1], 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 often been 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

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

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 heavily rely 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, 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 $\Theta = 0, 1, \ldots, d-1$, 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 $\sum X^{2^{i_1}+...+2^{i_k}} Q^{2^j}$ 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:

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

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 $1 \leq j \leq a$. 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:

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

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 $\{\#\mathrm{P_{df}}\}_D = dim\big(ker(\psi_{D-2})/T(\psi_{D-2})\big)$ for its dimension, and derive the following estimate for $\{\#\mathrm{P_{df}}\}_3$:

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

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 $\{\#\mathrm{P_{df}}\}_3$. 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*

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

can be written on the form:

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

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^Θ} (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:

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

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 $\sum_{i\neq j} b_{i,j}p_ip_j$, 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 generated at step degree $D = 3$, and not step degree $D = 4$.
- $X^{2^\Theta}\beta_1 = X^{1+2^\Theta}Q = X\beta_2$, 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 $(3n - 9)d$.

It may initially seem like there are $a\binom{n}{2}$ removed polynomials of degree 4, 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} (predicted)	D_{ff} (Magma)	$\{\#\mathrm{P}_{\mathrm{df}}\}_{D_{ff}}$ (Magma)	D_{solv}	$a+3$	D_{reg}
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).

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

ω \ D_{solv}	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. 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. 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.
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. C. Bouillaguet, P.-A. Fouque, and G. Macario-Rat. Practical Key-recovery For All Possible Parameters of SFLASH. In *Advances in Cryptology - ASIACRYPT 2011-17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 667–685. Springer, 2011.
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. Faugère. 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. 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. 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.