Algebraic Attacks in Post–Quantum Cryptography

Morten Øygarden
SimulaUiB

Jan 2020

Overview

Table of Content

Polynomial Systems

Consider a system composed of m quadratic polynomials in n variables over a finite field $\mathbb{F}$. By *solving* we mean finding one common root to these m polynomials.

$$\begin{aligned} p_1(x_1, \ldots, x_n) &= \sum \alpha_{i,j}^{(1)} x_i x_j + \sum \beta_{i,j}^{(1)} x_i + \gamma^{(1)} = 0 \\ \vdots & \qquad\qquad \vdots \\ p_m(x_1, \ldots, x_n) &= \sum \alpha_{i,j}^{(m)} x_i x_j + \sum \beta_{i,j}^{(m)} x_i + \gamma^{(m)} = 0 \end{aligned}$$

Example: Multivariate Public Key Cryptosystems

- **public key** is typically a system of m quadratic polynomial equations in n variables over a finite field (often of characteristic 2 in practice).

Example: Multivariate Public Key Cryptosystems

- **public key** is typically a system of m quadratic polynomial equations in n variables over a finite field (often of characteristic 2 in practice).
- **encryption**: evaluate the polynomials $p_1(x_1, \ldots, x_n), \ldots, p_m(x_1, \ldots, x_n)$ on plaintext $(m_1, \ldots, m_n)$ to obtain $(c_1, \ldots, c_m)$.

Example: Multivariate Public Key Cryptosystems

- **public key** is typically a system of m quadratic polynomial equations in n variables over a finite field (often of characteristic 2 in practice).
- **encryption**: evaluate the polynomials $p_1(x_1, \ldots, x_n), \ldots, p_m(x_1, \ldots, x_n)$ on plaintext $(m_1, \ldots, m_n)$ to obtain $(c_1, \ldots, c_m)$.
- **decryption**: solve the system

$$\begin{aligned} p_1(x_1, \ldots, x_n) &= c_1 \\ \ldots & \\ p_m(x_1, \ldots, x_n) &= c_m \end{aligned}$$

  to recover plaintext $(m_1, \ldots, m_n)$.

Example: Multivariate Public Key Cryptosystems

- **public key** is typically a system of m quadratic polynomial equations in n variables over a finite field (often of characteristic 2 in practice).
- **encryption**: evaluate the polynomials $p_1(x_1, \ldots, x_n), \ldots, p_m(x_1, \ldots, x_n)$ on plaintext $(m_1, \ldots, m_n)$ to obtain $(c_1, \ldots, c_m)$.
- **decryption**: solve the system

$$\begin{aligned} p_1(x_1, \ldots, x_n) &= c_1 \\ \ldots & \\ p_m(x_1, \ldots, x_n) &= c_m \end{aligned}$$

  to recover plaintext $(m_1, \ldots, m_n)$.
- **features**: efficient encryption, compact ciphertext, reasonably large public keys, decryption efficiency depends on how easy is to solve the system above (with knowledge of the secret key), post-quantum security.

Linearisation

The most efficient Gröbner basis algorithms solves $\{p_i(x_1, ..., x_n) = 0\}_i$ by step–wise reducing certain Macaulay matrices according to degree. Idea is as follows:

Linearisation

The most efficient Gröbner basis algorithms solves $\{p_i(x_1, ..., x_n) = 0\}_i$ by step–wise reducing certain Macaulay matrices according to degree. Idea is as follows:

D = 3:

$$\begin{array}{c} \\ x_i p_j \end{array} \begin{array}{c} \begin{array}{cccccc} x_i x_j x_k & \dots & x_i x_j & \dots & x_i & \dots \end{array} \\ \begin{pmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix} \end{array}$$

Linearisation

The most efficient Gröbner basis algorithms solves $\{p_i(x_1, ..., x_n) = 0\}_i$ by step–wise reducing certain Macaulay matrices according to degree. Idea is as follows:

D = 3:

$$\begin{array}{c} \\ x_i p_j \end{array} \begin{array}{c} \begin{array}{cccccc} x_i x_j x_k & \dots & x_i x_j & \dots & x_i & \dots \end{array} \\ \begin{pmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix} \end{array}$$

D = 4:

$$\begin{array}{c} \\ x_i x_j p_k \end{array} \begin{array}{c} \begin{array}{cccccccc} x_i x_j x_k x_l & \dots & x_i x_j x_k & \dots & x_i x_j & \dots & x_i & \dots \end{array} \\ \begin{pmatrix} * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{pmatrix} \end{array}$$

Big Field Multivariate Schemes

Idea: let $\mathbb{F}_{2^n}$ be an extension field of $\mathbb{F}_2$ of degree n; then use a polynomial $F \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} + X\rangle$ which is efficiently computable, and efficiently invertible, but try to *hide* its structure by representing the encryption by polynomials $p_1, \ldots, p_n$ in $\mathbb{F}_2[x_1, x_2, \ldots, x_n]$ (using the public isomorphism ϕ).

Big Field Multivariate Schemes

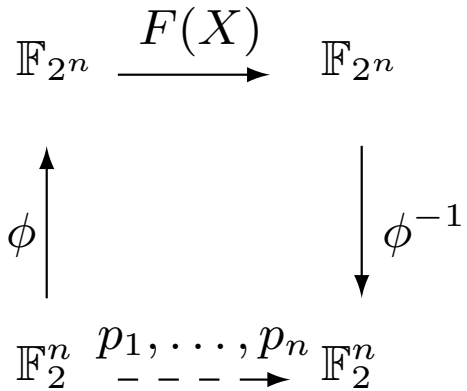
Idea: let $\mathbb{F}_{2^n}$ be an extension field of $\mathbb{F}_2$ of degree n; then use a polynomial $F \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} + X\rangle$ which is efficiently computable, and efficiently invertible, but try to *hide* its structure by representing the encryption by polynomials $p_1, \ldots, p_n$ in $\mathbb{F}_2[x_1, x_2, \ldots, x_n]$ (using the public isomorphism ϕ).

Fact: The degree of the polynomials $p_1, \ldots, p_n$ is equal to the maximal Hamming weight of the univariate polynomial. Example $F(X) = X^3 = X^{2^0+2^1} \Rightarrow p_1, \ldots, p_n$ are quadratic.

Big Field Multivariate Schemes

Idea: let $\mathbb{F}_{2^n}$ be an extension field of $\mathbb{F}_2$ of degree n; then use a polynomial $F \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} + X\rangle$ which is efficiently computable, and efficiently invertible, but try to *hide* its structure by representing the encryption by polynomials $p_1, \ldots, p_n$ in $\mathbb{F}_2[x_1, x_2, \ldots, x_n]$ (using the public isomorphism ϕ).

Fact: The degree of the polynomials $p_1, \ldots, p_n$ is equal to the maximal Hamming weight of the univariate polynomial. Example $F(X) = X^3 = X^{2^0+2^1} \Rightarrow p_1, \ldots, p_n$ are quadratic.



Example: C^*

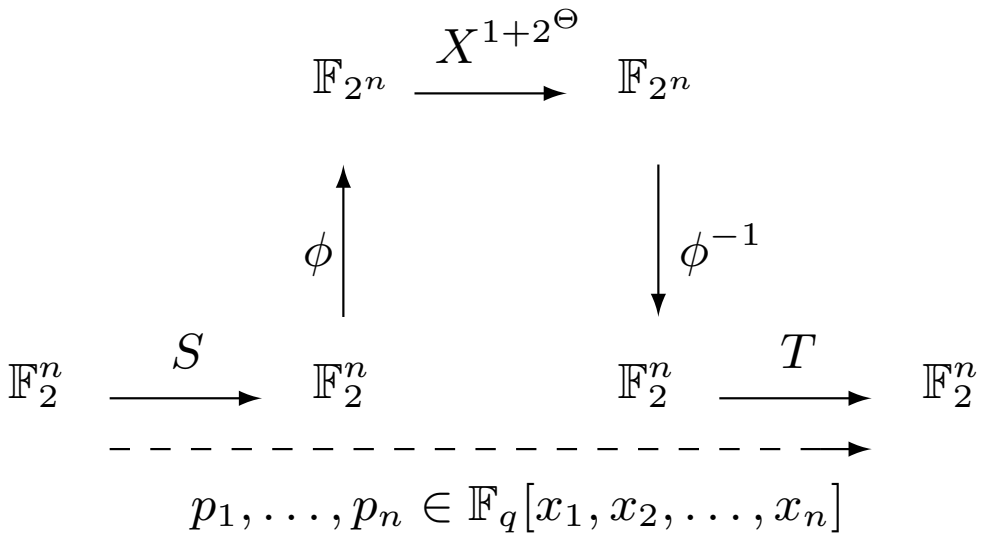
- **Efficiency**: we may use $F(X) = X^{1+2^\Theta}$. Then inversion can be done efficiently; moreover polynomials $p_1, \ldots, p_n$ are quadratic (since X^{2^Θ} is linear over $\mathbb{F}_2$).

Example: C^*

- **Efficiency**: we may use $F(X) = X^{1+2^\Theta}$. Then inversion can be done efficiently; moreover polynomials $p_1, \ldots, p_n$ are quadratic (since X^{2^Θ} is linear over $\mathbb{F}_2$).
- **Security**: the construction does **not** hide the structure of F, but we may do that by taking linear combinations of the variables and polynomials (using *secret* matrices S and T), to generate the *public* polynomials $p_1, \ldots, p_n$

Example: C^*

- **Efficiency**: we may use $F(X) = X^{1+2^\Theta}$. Then inversion can be done efficiently; moreover polynomials $p_1, \ldots, p_n$ are quadratic (since X^{2^Θ} is linear over $\mathbb{F}_2$).
- **Security**: the construction does **not** hide the structure of F, but we may do that by taking linear combinations of the variables and polynomials (using *secret* matrices S and T), to generate the *public* polynomials $p_1, \ldots, p_n$



A Weakness of Big–Field Schemes

- Big–Field schemes tend to carry a lot of algebraic structure (hailing from the extension field). These can be exploited by Gröbner basis algorithms.

[a] In fact, for C^* we can find linear polynomials through similar means, which broke the system.

A Weakness of Big–Field Schemes

- Big–Field schemes tend to carry a lot of algebraic structure (hailing from the extension field). These can be exploited by Gröbner basis algorithms.
- Example in C^*: $X \cdot F(X) = X \cdot X^{1+2^\Theta} = X^{2+2^\Theta}$. This combination will correspond to "new" quadratic polynomials, that can be exploited by an attacker[a]. (Recall that the linear transformations does not alter the degree).

[a] In fact, for C^* we can find linear polynomials through similar means, which broke the system.

A Weakness of Big–Field Schemes

- Big–Field schemes tend to carry a lot of algebraic structure (hailing from the extension field). These can be exploited by Gröbner basis algorithms.
- Example in C^*: $X \cdot F(X) = X \cdot X^{1+2^\Theta} = X^{2+2^\Theta}$. This combination will correspond to "new" quadratic polynomials, that can be exploited by an attacker[a]. (Recall that the linear transformations does not alter the degree).
- Current iterations of Big–Field Schemes rely on the use of certain modifiers to hide the algebraic structure. Examples include removing public polynomials, or adding extra information (variables or whole polynomials).

[a] In fact, for C^* we can find linear polynomials through similar means, which broke the system.

Overview, Big–Field Schemes

There are several variants of secure Big–Field multivariate *signature* schemes:

- GeMMS (variant of HFE$v-$) (Casanova et al., Round 2 NIST Submission, 2017)
- PFLASH (Chen et al., 2015)

[b] Currently analysing security in a joint work with H.Raddum, P. Felke, and C.Cid.

Overview, Big–Field Schemes

There are several variants of secure Big–Field multivariate *signature* schemes:

- GeMMS (variant of HFE$v-$) (Casanova et al., Round 2 NIST Submission, 2017)
- PFLASH (Chen et al., 2015)

On the other hand, designing a *secure* and *efficient encryption* scheme is **very hard**:

- C^* (Imai-Matsumoto, Eurocrypt'88): **broken** (Patarin, Crypto'95)
- HFE (Patarin, Eurocrypt'96): **broken** (Faugère–Joux, Crypto'03)

[b] Currently analysing security in a joint work with H.Raddum, P. Felke, and C.Cid.

Overview, Big–Field Schemes

There are several variants of secure Big–Field multivariate *signature* schemes:

- GeMMS (variant of HFE$v-$) (Casanova et al., Round 2 NIST Submission, 2017)
- PFLASH (Chen et al., 2015)

On the other hand, designing a *secure* and *efficient encryption* scheme is **very hard**:

- C^* (Imai-Matsumoto, Eurocrypt'88): **broken** (Patarin, Crypto'95)
- HFE (Patarin, Eurocrypt'96): **broken** (Faugère–Joux, Crypto'03)
- EFC (Szepieniec et al., PQC'16): **one variant broken** (Wang et al., 2019)[b]
- HFERP (Ikematsu et al., PQC'18)
- Two–Face (Macario–Rat and Patarin, Africacrypt'18)[b]
- EFLASH (Cartor and Smith–Tone, SAC'18): **broken** (Øygarden et al., CT–RSA'20)

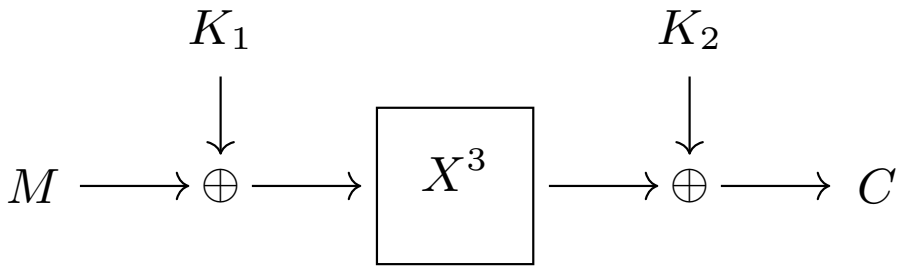
[b] Currently analysing security in a joint work with H.Raddum, P. Felke, and C.Cid.

Algebraically Simple Block Ciphers

- Recent years have seen the design of symmetric ciphers that are optimized for advanced cryptographic protocols such as Multi–Party Computation, Zero–Knowledge proofs and Fully Homomorphic Encryption. A design goal is often to minimize the multiplicative complexity (MC).

Algebraically Simple Block Ciphers

- Recent years have seen the design of symmetric ciphers that are optimized for advanced cryptographic protocols such as Multi–Party Computation, Zero–Knowledge proofs and Fully Homomorphic Encryption. A design goal is often to minimize the multiplicative complexity (MC).
- Example: One round of the MiMC–cipher:



(Number of rounds needs to be large in order to be resistant against interpolation attacks).

Overview, Algebraically Simple Block Ciphers

Several new algebraically simple symmetric ciphers have been designed in recent years. This has in particular been accelerated by the ongoing *STARK-Friendly Hash Challenge* competition[c].

[c] `https://starkware.co/hash-challenge/`

Overview, Algebraically Simple Block Ciphers

Several new algebraically simple symmetric ciphers have been designed in recent years. This has in particular been accelerated by the ongoing *STARK-Friendly Hash Challenge* competition[c].

- MiMC (Albrecht et al., Asiacrypt'16)
- Jarvis and Friday (Ashur and Dhoogle, 2018) **broken**
- Starkad and Poseidon (Grassi et al., 2019)
- Vision and Rescue (Aly et al., 2019)
- GMiMC (Albrecht et al., 2019)

[c] `https://starkware.co/hash-challenge/`

Overview, Algebraically Simple Block Ciphers

Several new algebraically simple symmetric ciphers have been designed in recent years. This has in particular been accelerated by the ongoing *STARK-Friendly Hash Challenge* competition[c].

- MiMC (Albrecht et al., Asiacrypt'16)
- Jarvis and Friday (Ashur and Dhoogle, 2018) **broken**
- Starkad and Poseidon (Grassi et al., 2019)
- Vision and Rescue (Aly et al., 2019)
- GMiMC (Albrecht et al., 2019)

Security of these ciphers are not well understood. Some recent relevant analysis:

- Improved interpolation on round–reduced MiMC (Li–Preneel, SAC'19)
- Cryptanalysis of Jarvis and Friday with Gröbner bases (Albrecht et al., Asiacrypt'19)
- Attack on full–round MiMC (Strong Attack Model). Current joint work with L. Grassi, C. Rechberger, and M. Schofnegger.

[c] `https://starkware.co/hash-challenge/`

Bilinear Polynomial Systems

- Let $x_1, ..., x_{n_1}$ and $y_1, ..., y_{n_2}$ be two sets of variables. Then a bilinear polynomial in these two sets can be written on the form:

$$p(x_1, ..., y_{n_2}) = \sum a_{i,j} x_i y_j + \sum b_i x_i + \sum c_j y_j + d, \quad \text{for } a, b, c, d \in \mathbb{F}.$$

Bilinear Polynomial Systems

- Let $x_1, ..., x_{n_1}$ and $y_1, ..., y_{n_2}$ be two sets of variables. Then a bilinear polynomial in these two sets can be written on the form:

$$p(x_1, ..., y_{n_2}) = \sum a_{i,j} x_i y_j + \sum b_i x_i + \sum c_j y_j + d, \quad \text{for } a, b, c, d \in \mathbb{F}.$$

- Bilinear systems may for example arise in cryptography through matrix multiplication:

$$0 = XY = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \begin{bmatrix} y_1 & y_2 \\ y_3 & y_4 \end{bmatrix} = \begin{bmatrix} x_1y_1 + x_2y_3 & x_1y_2 + x_2y_4 \\ x_3y_1 + x_4y_3 & x_3y_2 + x_4y_4 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix}.$$

Solving Bilinear Systems

- (Faugère et al., 2011) showed that the left kernel of the jacobian matrix of the polynomial system (w.r.t the x– or y–variables) will be in the ideal of the system.

$$Jac_X(\mathcal{P}) = \begin{bmatrix} \frac{\partial p_1}{\partial x_1} & \frac{\partial p_1}{\partial x_2} & \cdots & \frac{\partial p_1}{\partial x_{n_x}} \\ \frac{\partial p_2}{\partial x_1} & \frac{\partial p_2}{\partial x_2} & \cdots & \frac{\partial p_2}{\partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial p_m}{\partial x_1} & \frac{\partial p_m}{\partial x_2} & \cdots & \frac{\partial p_m}{\partial x_{n_x}} \end{bmatrix}$$

Solving Bilinear Systems

- (Faugère et al., 2011) showed that the left kernel of the jacobian matrix of the polynomial system (w.r.t the x– or y–variables) will be in the ideal of the system.

$$Jac_X(\mathcal{P}) = \begin{bmatrix} \frac{\partial p_1}{\partial x_1} & \frac{\partial p_1}{\partial x_2} & \cdots & \frac{\partial p_1}{\partial x_{n_x}} \\ \frac{\partial p_2}{\partial x_1} & \frac{\partial p_2}{\partial x_2} & \cdots & \frac{\partial p_2}{\partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial p_m}{\partial x_1} & \frac{\partial p_m}{\partial x_2} & \cdots & \frac{\partial p_m}{\partial x_{n_x}} \end{bmatrix}$$

- Furthermore, (Verbel, et al., 2019) and (Bardet et al., 2019) showed that this left kernel has additional structure when the polynomial system is generated by matrix multiplication.

Solving Bilinear Systems

- (Faugère et al., 2011) showed that the left kernel of the jacobian matrix of the polynomial system (w.r.t the x– or y–variables) will be in the ideal of the system.

$$Jac_X(\mathcal{P}) = \begin{bmatrix} \frac{\partial p_1}{\partial x_1} & \frac{\partial p_1}{\partial x_2} & \cdots & \frac{\partial p_1}{\partial x_{n_x}} \\ \frac{\partial p_2}{\partial x_1} & \frac{\partial p_2}{\partial x_2} & \cdots & \frac{\partial p_2}{\partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial p_m}{\partial x_1} & \frac{\partial p_m}{\partial x_2} & \cdots & \frac{\partial p_m}{\partial x_{n_x}} \end{bmatrix}$$

- Furthermore, (Verbel, et al., 2019) and (Bardet et al., 2019) showed that this left kernel has additional structure when the polynomial system is generated by matrix multiplication.
- Combining these works implies that an attacker can pre–calculate this left kernel, which will be polynomials of (relatively) low degree in one of the sets of variables. This avoids reducing huge Macaulay matrices (as e.g. in a straightforward Gröbner basis calculation).

Effects on Rank–Metric Code Based Cryptosystems

- Currently there are two rank–metric code based encryption systems in the NIST competition, ROLLO and RQC.

Effects on Rank–Metric Code Based Cryptosystems

- Currently there are two rank–metric code based encryption systems in the NIST competition, ROLLO and RQC.

- (Bardet et al., 2019) showed that these cryptosystems can be broken by solving certain bilinear polynomial systems.

Effects on Rank–Metric Code Based Cryptosystems

- Currently there are two rank–metric code based encryption systems in the NIST competition, ROLLO and RQC.
- (Bardet et al., 2019) showed that these cryptosystems can be broken by solving certain bilinear polynomial systems.
- In particular, their attack breaks 11 out of the 12 suggested parameter sets of ROLLO and RQC.

Summary

Big–Field Multivariate Schemes.

- *Signature* schemes have been studied for a long time, and their security is understood (to some extent). E.g. HFEv−.

Summary

Big–Field Multivariate Schemes.

- *Signature* schemes have been studied for a long time, and their security is understood (to some extent). E.g. HFEv−.
- *Encryption* schemes are a different story. While much work has been done towards design, many of these have subsequently been broken. Efficient and secure big–field multivariate encryption schemes still seems far away.

Summary

Big–Field Multivariate Schemes.

- *Signature* schemes have been studied for a long time, and their security is understood (to some extent). E.g. HFEv−.
- *Encryption* schemes are a different story. While much work has been done towards design, many of these have subsequently been broken. Efficient and secure big–field multivariate encryption schemes still seems far away.

Algebraically Simple Symmetric Ciphers

- Several new designs in recent years, aimed to be effective for ZK, MPC and/or FHE.

Summary

Big–Field Multivariate Schemes.

- *Signature* schemes have been studied for a long time, and their security is understood (to some extent). E.g. HFEv−.
- *Encryption* schemes are a different story. While much work has been done towards design, many of these have subsequently been broken. Efficient and secure big–field multivariate encryption schemes still seems far away.

Algebraically Simple Symmetric Ciphers

- Several new designs in recent years, aimed to be effective for ZK, MPC and/or FHE.
- Still a lot of work to be done to fully understand their security.

Summary

Big–Field Multivariate Schemes.

- *Signature* schemes have been studied for a long time, and their security is understood (to some extent). E.g. HFEv−.
- *Encryption* schemes are a different story. While much work has been done towards design, many of these have subsequently been broken. Efficient and secure big–field multivariate encryption schemes still seems far away.

Algebraically Simple Symmetric Ciphers

- Several new designs in recent years, aimed to be effective for ZK, MPC and/or FHE.
- Still a lot of work to be done to fully understand their security.

Rank–Metric Code Based Encryption Schemes

- Recent breakthrough in analysis, based on Gröbner basis methods.

Summary

Big–Field Multivariate Schemes.

- *Signature* schemes have been studied for a long time, and their security is understood (to some extent). E.g. HFEv−.
- *Encryption* schemes are a different story. While much work has been done towards design, many of these have subsequently been broken. Efficient and secure big–field multivariate encryption schemes still seems far away.

Algebraically Simple Symmetric Ciphers

- Several new designs in recent years, aimed to be effective for ZK, MPC and/or FHE.
- Still a lot of work to be done to fully understand their security.

Rank–Metric Code Based Encryption Schemes

- Recent breakthrough in analysis, based on Gröbner basis methods.
- Remains to be seen how much these attacks can be improved, and whether there at the end of the day is possible to choose secure and efficient parameters.

Thank you!