Introduction to Multivariate Cryptography

Carlo Sanna

Cryp**TO**

Group of Cryptography and Number Theory of Politecnico di Torino

Structure of the Talk

- Motivation
  - Public Key Cryptography
  - RSA Cryptosystem
  - The Threat of Quantum Computers
  - Post-Quantum Cryptography
- Multivariate Cryptography
  - The MQ Problem
  - Basic Design (Bipolar Systems)
  - Unbalanced Oil and Vinegars (UOV)
  - Hidden Field Equations (HFE)
  - Types of Attacks (brief mentions)

Public Key Cryptography (or Asymmetric Cryptography)

In a Public Key Cryptosystem:

- Each user has two keys: a **public key**, which everybody knows, and a **secret key**, which only the user knows.
- Encryption is done using an encryption function $E : \mathcal{P} \times \mathcal{M} \to \mathcal{M}'$, which takes as input a public key and a plain message, and returns as output an encrypted message.
- Decryption is done using a decryption function $D : \mathcal{S} \times \mathcal{M}' \to \mathcal{M}$, which takes as input a (corresponding) secret key and an encrypted message, and returns as output the original plain message.

Depending on the cryptosystem, some mathematical theory guarantees that retrieving the plain message from the encrypted message without knowing the secret key, or retrieving the secret key from the public key, is extremely difficult.

Public Key Cryptography: Encryption / Decryption

Example: Alice wants to send a message m to Bob.

1. Alice computes the encrypted message $m' = E(\text{pk}_{Bob}, m)$ using the public key of Bob and her plain message m.
2. Alice sends the encrypted message m' to Bob.
3. Bob decrypts m' using his secret key and read the original plain message $m = D(\text{sk}_{Bob}, m')$.

Even if the channel used by Alice to transmit the encrypted message is insecure, and the evil Charles obtain a copy of the encrypted message m', he cannot decrypt it, because he needs Bob's secret key.

Public Key Cryptography: Signature

Public Key Cryptography can also be used to **sign** a message (encrypted or not), that is, to guarantee that you are the author.

Example: Alice wants to send a signed message m to Bob.

1. Alice computes the "decrypted" message $m^* = D(\text{sk}_{Alice}, m)$ using her secret key of and her message m.
2. Alice sends m^* to Bob.
3. Bob "encrypt" m^* using Alice public key and read the message $m = E(\text{pk}_{Alice}, m^*)$.

Only Alice can have produced m^*, because her secret key was necessary. If Bob sees that m is a bunch of meaningless numbers, instead of a meaningful message, then it means that m^* was not produced by Alice.

RSA Public Key Cryptography

The first and most widely used Public Key Cryptosystem is the RSA cryptosystem. It was invented by Rivest, Shamir, and Adleman in 1977. An equivalent cryptosystem was developed in secret by Clifford Cocks in 1973 for the British signals intelligence agency (declassified in 1997).

RSA is based on Number Theory (Modular Arithmetic). In RSA, each user has a secret key $\mathrm{sk} = (p, q, d)$ consisting of two large prime numbers p, q and an integer d (computed in terms of p, q); and a public key $\mathrm{pk} = (n, e)$ consisting of $n = pq$ and an integer e (computed also in terms of p, q).

Recovering the secret key of a user from its public key, requires to factorize n into its prime factors p, q. This is the **Factorization Problem** and no efficient (classical) algorithm to solve it for large n is known. Note that humans are studying prime factors at least since Euclid ($\sim$300 BC).

Who is Afraid of Quantum Computers?

In 1994, Peter Shor invented a **quantum algorithm**, now known as **Shor's algorithm**, to efficiently solve the Factorization Problem. This algorithm needs to run on a quantum computer, and at the time it was "only" an important theoretical result.

Due to the recent progress into quantum computing, the advent of quantum computers, and the actual implementation of Shor's algorithm, seems more imminent than ever and constitutes a real threat to RSA.

Consequently, a lot of effort is put into **Post-Quantum Cryptography**, that is, asymmetric cryptography designed to be resistant even against attacks using quantum computers. Post-Quantum Cryptography includes:

- Multivariate Cryptography
- Code-based Cryptography
- Hash-based Cryptography
- Lattice-based Cryptography
- Supersingular Elliptic Curve Isogeny Cryptography

Multivariate Cryptography

Multivariate Cryptography is based on the following problem:

Multivariate Quadratic Problem (MQ Problem)

Data:

- A finite field of q elements $\mathbb{F}_q$;
- m quadratic polynomials $p_1, \ldots, p_m \in \mathbb{F}_q[X_1, \ldots, X_n]$ in n variables.

Question: Find a solution $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$ of the system of equations

$$p_i(X_1, \ldots, X_n) = 0, \quad i = 1, \ldots, m.$$

The MQ problem is proved to be NP-complete, and it seems that using quantum computers does not provide an advantage to solve it.
However, NP-completeness does not exclude the possibility that for certain polynomials $p_1, \ldots, p_m$ we can efficiently find a solution. Actually, we will see that in Multivariate Cryptography one **needs** to efficiently solve the system in order to decrypt a message.

Multivariate Public Key Cryptosystem

In a Multivariate Public Key Cryptosystem (MPKC)

- The public key $\mathrm{pk} = (p_1, \ldots, p_m)$ consists of a m-tuple of quadratic polynomials $p_1, \ldots, p_m \in \mathbb{F}_q[X_1, \ldots, X_n]$ in n variables.
- The encryption func. is the polynomial map $E_{\mathrm{pk}} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ defined by

$$E_{\mathrm{pk}}(X_1, \ldots, X_n) = \big(p_1(X_1, \ldots, X_n), \ldots, p_m(X_1, \ldots, X_n)\big).$$

- The secret key consists of data on how $p_1, \ldots, p_m$ have been generated (depends on the cryptosystem) and makes possible to easily invert E_{pk} using the decryption function.

Decrypting an encrypted message without knowing the secret key is an instance of the MQ Problem, and so it should be difficult even for a quantum computer. However, the special structure of $p_1, \ldots, p_m$, which is necessary to the decryption, can lead to vulnerabilities of the cryptosystem.

Bipolar Systems

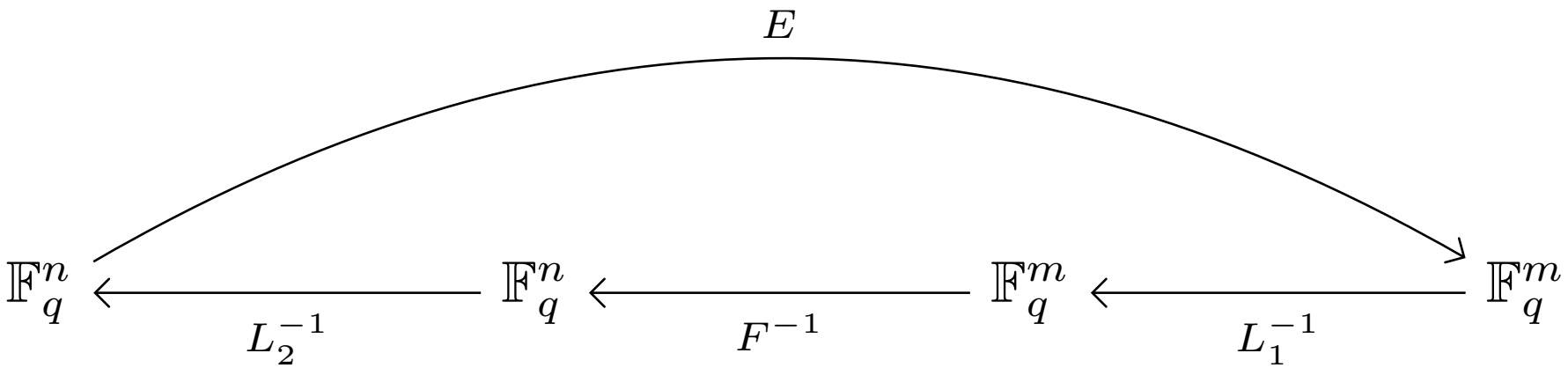
Most of MPKCs are bipolar systems, which means that:

- The secret key $\mathrm{sk} = (F, L_1, L_2)$ consists of a quadratic polynomial map $F : \mathbb{F}_q^n \to \mathbb{F}_q^m$, called **central map** and whose special structure makes easy to invert it, and two random affine maps $L_1 : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $L_2 : \mathbb{F}_q^n \to \mathbb{F}_q^n$.
- The public key $\mathrm{pk} = (p_1, \ldots, p_m)$ consists of the m components of the encryption function defined by $E_{\mathrm{pk}} = L_1 \circ F \circ L_2$.

The purpose of L_1 and L_2 is to mask the special structure of F.

Bipolar Systems: Diagram

Encryption / Signature Verification



Decryption / Signature Generation

NIST Multivariate Candidates: Rainbow and GeMSS

In 2016, the National Institute of Standards and Technology (NIST) started a competition to select the Post-Quantum cryptosystems more fit for standardization.

Currently, two MPKCs are in the run:

- **Rainbow**, a signature scheme based on a multi-layer version of the UOV scheme, is a Round 3 Finalist;
- **GeMSS** (**Gre**at **M**ultivariate **S**hort **S**ignature) a signature scheme based on HFEv–, is an Alternate Candidate.

Rainbow: Sizes

NIST security category	public key size (kB)	secret key size (kB)	signature (bits)
I	157.8	101.2	528
III	861.4	611.3	1312
V	1885.4	1375.7	1632

For comparison, RSA keys are usually of 2048 or 4096 bits.

GeMSS: Sizes

Scheme	public key size (kB)	secret key size (bits)	signature size (bits)
GeMSS128	352.19	128	258
BlueGeMSS128	363.61	128	270
RedGeMSS128	375.21	128	282
GeMSS192	1237.96	192	411
BlueGeMSS192	1264.12	192	423
RedGeMSS192	1290.54	192	435
GeMSS256	3040.70	256	576
BlueGeMSS256	3087.96	256	588
RedGeMSS256	3135.59	256	600

Note: The secret key is compressed as a seed of a pseudorandom number generator. The decompressed secret key has size between 10 and 80 kB.

Unbalanced Oil and Vinegar (UOV)

Unbalanced Oil and Vinegar (UOV) is a multivariate signature scheme introduced by Kipnis, Patarin, and Goubin in 1999, which generalizes the previous Oil and Vinegar scheme of Patarin.

The variables $X_1, \ldots, X_n$ are divided into two classes:

- The **oil variables** $X_1, \ldots, X_o = O_1, \ldots, O_o$; and
- The **vinegar variables** $X_{o+1}, \ldots, X_{o+v} = V_1, \ldots, V_v$.

The central map has the form $F = (f_1, \ldots, f_m)$ where $m = o$ and

$$f_k(X_1, \ldots, X_n) = \sum_{i,j} a_{i,j,k} O_i V_j + \sum_{i,j} b_{i,j,k} V_i V_j + \sum_i c_{i,k} O_i + \sum_i d_{i,k} V_i + e_k.$$

That is, vinegar variables "mix" quadratically with all other variables and oil variables never "mix" with themselves (the terms $O_i O_j$ are missing).

UOV: Inverting F

Knowing the secret data $a_{i,j,k}, b_{i,j,k}, c_{i,k}, d_{i,k}, e_k$ of the central map F, we can easily invert F.

Indeed, given $(y_1, \ldots, y_m) \in \mathbb{F}_q^m$, we can find $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$ such that $F(x_1, \ldots, x_n) = (y_1, \ldots, y_m)$ in the following way:

1. Set the vinegar variables $x_{o+1}, \ldots, x_{o+v}$ to random values;
2. Solve the **linear system** in $X_1, \ldots, X_o$

$$f_k(X_1, \ldots, X_o, x_{o+1}, \ldots, x_{o+v}) = y_k, \quad k = 1, \ldots, m,$$

to find the oil variables $x_1, \ldots, x_o$. (Note that $m = o$, so the system is expected to have one solution.)

UOV: Security

The security of UOV depends on the relative sizes of v and o (the number of equations):

- For $v = o$ (the original Oil and Vinegar) or v slightly larger than o, the scheme was broken by Kipnis and Shamir (structural attack).
- For $v \gtrsim o^2$ the scheme is not secure because there are efficient algorithm to find solutions of very underdetermined quadratic systems.

A recommended choice is $v \simeq 3o$.

Hidden Field Equations (HFE)

Hidden Field Equations (HFE) is a family of MPKCs of the bipolar type where the central map is defined using an univariate polynomial over an extension field. It was introduced by Patarin in 1996 as a generalization of a previous scheme of Matsumoto and Imai.

Let $\mathbb{F}_{q^n}$ be an extension of $\mathbb{F}_q$ of degree n. Assume that $\mathbb{F}_{q^n}$ is constructed as the quotient ring $\mathbb{F}_q[T]/(f)$, where $f \in \mathbb{F}_q[T]$ is irreducible of degree n. Thus we have an isomorphism of $\mathbb{F}_q$-linear spaces $\varphi : \mathbb{F}_{q^n} \to \mathbb{F}_q^n$ given by

$$\varphi\left(x_1 + x_2 T + \cdots + x_n T^{n-1}\right) := (x_1, x_2, \ldots, x_n).$$

The secret key in HFE is a triple (P, L_1, L_2), where L_1, L_2 are affine maps $\mathbb{F}_q^n \to \mathbb{F}_q^n$, and P is a polynomial of degree d with coefficients in $\mathbb{F}_{q^n}$ of the special form

$$P(X) = \sum_{q^i + q^j \leq d} a_{i,j} X^{q^i + q^j} + \sum_{q^i \leq d} b_i X^{q^i}.$$

The central map $F : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is then defined by $F = \varphi \circ P \circ \varphi^{-1}$, and the encryption by $E = L_1 \circ F \circ L_2$.

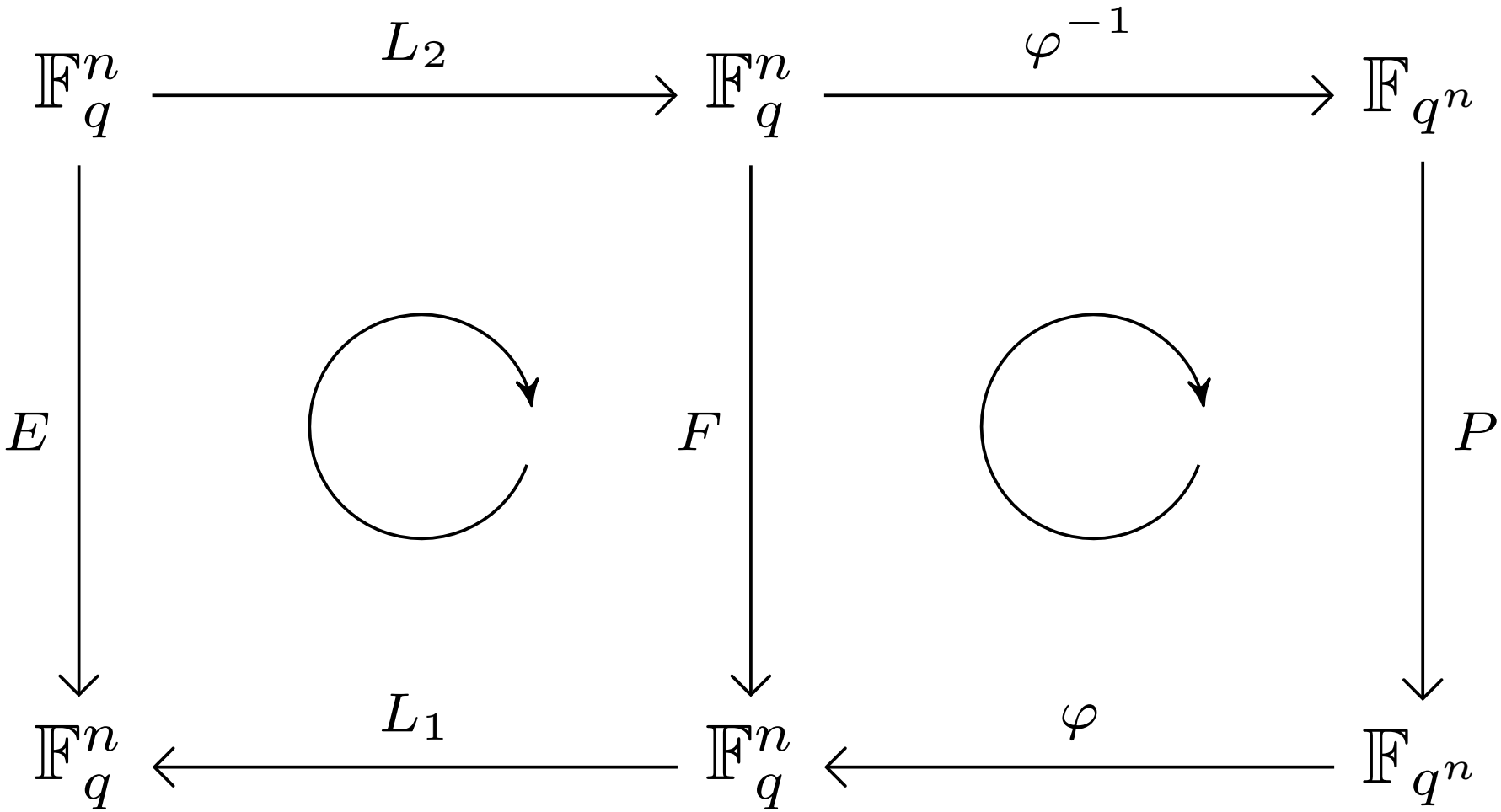
HFE: Public Key

The public key consists of the polynomials $p_1, \ldots, p_n \in \mathbb{F}_q[X_1, \ldots, X_n]$ such that

$$E(X_1, \ldots, X_n) = \big(p_1(X_1, \ldots, X_n), \ldots, p_n(X_1, \ldots, X_n)\big).$$

The purpose of the special form of P is to guarantees that the polynomials $p_1, \ldots, p_n$ are quadratic.

HFE: Diagram


$\mathbb{F}_q^n$
L_2
$\mathbb{F}_q^n$
φ^{-1}
$\mathbb{F}_{q^n}$
E
F
P
$\mathbb{F}_q^n$
L_1
$\mathbb{F}_q^n$
φ
$\mathbb{F}_{q^n}$

HFE: Encryption / Decryption

Encryption: Given a plaintext message $m = (m_1, \ldots, m_n) \in \mathbb{F}_q^n$, the corresponding ciphertext $e = (e_1, \ldots, e_n) \in \mathbb{F}_q^n$ is computed by $e = E(m)$. In addition, a redundancy $r := H(m)$ is computed from the plaintext message m, where H is a hash function. The final output is (e, r).

Decryption: Given a ciphertext $e = (e_1, \ldots, e_n) \in \mathbb{F}^n$, decryption consists of the following steps:

1. Compute $y := (\varphi^{-1} \circ L_1^{-1})(e) \in \mathbb{F}_{q^n}$.
2. Solve the polynomial equation $P(X) = y$ over $\mathbb{F}_{q^n}$. This can be done efficiently using a version of the Berlekamp algorithm.
3. Compute $m = (L_2^{-1} \circ \varphi)(x)$ for each solution x of the equation $P(X) = y$. Since the equation $P(X) = y$ might have multiple solutions (but no more than d), a method to distinguish the correct plaintext message m is necessary. This is done with the aid of the redundancy r, by checking that $H(m) = r$.

HFE: Attacks and Variations

"Pure" HFE is not secure, since many efficient attacks have been found. For example, Faugére and Joux (2003) gave a fast attack using Gröbner bases.

However, there exist unbroken variations of HFE:

- **HFE–** uses as public key only the polynomials $p_1, \ldots, p_{n-u}$, keeping secret the last u polynomials. As long as we transmit enough redundancy, it is still possible to recover the plaintext message.

  Roughly speaking, by keeping u polynomials secret, we loose $\log_2(q^u)$ bits of information, which have to be compensated by the redundancy, and we have to try up to q^u different possible encrypted messages, slowing down the decryption process by a factor q^u.

HFE: Variations

- **HFEv** changes the structure of the secret polynomial P by adding some vinegar variables $z_1, \ldots, z_v \in \mathbb{F}_q$. Thus HFEv works with a family of secret polynomials of the form

$$P_{z_1,\ldots,z_v}(X) = \sum_{q^i+q^j \leq d} a_{i,j} X^{q^i+q^j} + \sum_{q^i \leq d} b_i(z_1, \ldots, z_v) X^{q^i} + c(z_1, \ldots, z_v),$$

  where $a_{i,j} \in \mathbb{F}_{q^n}$ and $b_i, c \in \mathbb{F}_{q^n}[Z_1, \ldots, Z_v]$ with b_i linear and c quadratic (this is required for the public polynomials to be quadratic).

  The vinegar variables $z_1, \ldots, z_v$ are randomly initialized. Once the polynomial $P_{z_1,\ldots,z_v}$ is determined, the encryption proceeds unchanged, while for the decryption it is necessary to check up to q^v different equations $P_{z_1,\ldots,z_v}(X) = y$, trying the possible $z_1, \ldots, z_v$.

- **HFEv–** is a combination of HFE– and HFEv.

Attacks to MPKCs

There are two types of attacks to MPKCs:

- **Direct Attacks** try to invert the encryption map by solving the corresponding instance of the MQ Problem.
- **Structural Attacks** exploit in some way the special structure of the central map.

Direct Attacks

Direct attacks mainly employ Gröbner bases and/or exhaustive search.

To solve the polynomial system

$$p_i(X_1, \ldots, X_n) - y_i = 0, \quad i = 1, \ldots, m,$$

the *Gröbner basis* of the ideal $\mathcal{I}$ of $\mathbb{F}_q[X_1, \ldots, X_n]$ generated by $p_1 - y_1, \ldots, p_m - y_m$ is computed. This is a special set $\{g_1, \ldots, g_k\}$ of generators of $\mathcal{I}$ that makes easier to solve the polynomial system.

For example, a (lexicographical order-) Gröbner basis reduces the problem to solving a polynomial system of the form

$$g_1(X_1) = 0,$$
$$g_2(X_1, X_2) = 0, \quad g_3(X_1, X_2) = 0, \quad \ldots, \quad g_{k_1}(X_1, X_2) = 0,$$
$$g_{k_1+1}(X_1, X_2, X_3) = 0, \quad g_{k_1+2}(X_1, X_2, X_3) = 0, \quad \ldots, \quad g_{k_2}(X_1, X_2, X_3) = 0,$$
$$\ldots \quad g_k(X_1, \ldots, X_n) = 0,$$

which can be done easily by successive elimination of variables.

Computation of Gröbner Bases

The historical method to compute Gröbner basis is Buchberger's algorithm, which however is not efficient.

Many improved methods have been proposed, in particular the F_4 and F_5 algorithms, due to Faugére, and their many variations.

These are implemented in many mathematical softwares, including Magma, Maple, Singular, FGb...

Structural Attacks

Structural Attacks mostly fall into the categories of:

- **MinRank Attacks**, which search for a small-rank linear combinations of some matrices; and
- **Differential Attacks**, which search for symmetries or invariants of the **differential**

$$DF(x, a) := F(x + a) - F(x) - F(a) + F(0)$$

where F is the central map.

MinRank Problem

MinRank attacks reduces the cryptanalysis to an instance of the:

MinRank Problem

Data:

- Positive integers m, n, k, r;
- $m \times n$ matrices $M_0; M_1, \ldots, M_k$ with entries in $\mathbb{F}_q$.

Question: Determine a k-tuple $(\lambda_1, \ldots, \lambda_k) \in \mathbb{F}_q^k$ such that

$$\operatorname{rank}\left(M_0 - \sum_{i=1}^{k} \lambda_i M_i \right) \leq r.$$

The MinRank problem is NP-complete. The main methods to solve it are: Kernel attack, Kipnis–Shamir modeling, Minors modeling, and Support Minors modeling. The modeling methods work by reducing the MinRank problem to an instance of the MQ problem.

References


Bernstein, Buchmann, and Dahmen,
Post-Quantum Cryptography, Springer-Verlag, Berlin, 2009.


Ding, Gower, and Schmidt,
Multivariate Public Key Cryptosystems, Advances in Information Security, vol. 25, Springer, New York, 2006.


Kipnis, Patarin, and Goubin,
Unbalanced Oil and Vinegar Signature Schemes. In J. Stern, editor, Advances in Cryptology — EUROCRYPT '99, pages 206–222, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.


Patarin,
Hidden Field Equations (HFE) and Isomorphism of Polynomials (IP): Two New Families of Asymmetric Algorithms. In U. Maurer, editor, LNCS EUROCRYPT '96, volume 1070, pages 33–48. Springer, 1996.