Multivariate Public Key Cryptography

or

Why is there a rainbow hidden behind fields full of oil and vinegar?

Christian Eder, Jean-Charles Faugère and Ludovic Perret

Seminar on Fundamental Algorithms, University of Kaiserslautern

June 25, 2015
Introduction to public key cryptography

Construction of MPKC

Examples of MPKC schemes: Hidden fields

Examples of MPKC schemes: Oil & Vinegar
General idea of public key cryptography

- **complete key** (set of data)
- Public key $P$
- Private key $Q$
- Message $M$
- Ciphertext $C$
- Original message $M$
General idea of public key cryptography

- Public key $P$: subset of complete key
- Private key $Q$: complete key \ public key
- Complete key: set of data
- Message $M$: original message
- Ciphertext $C$: encrypted message
General idea of public key cryptography

- **message $M$**
- **public key $P$** (subset of complete key)
- **private key $Q$** (complete key \ public key)
- **ciphertext $C$**

complete key (set of data)
General idea of public key cryptography

- **Message** $M$
- **Ciphertext** $C$
- **Public key** $P$
- **Private key** $Q$
- **Complete key** (set of data)
- **Original message** $M$
General idea of signature schemes

original message $M$

hash function $h$:

$H = h(M)$

signature $S = Q(H)$ with private key $Q$

send tuple $(M, S)$

compute $H' = P(S)$ with public key $P$

compute hash $H = h(M)$ of message $M$

verify sender via testing $H = H'$
General idea of signature schemes

Original message $M$

Hash function $h$:

$H = h(M)$

Signature $S = Q(H)$ with private key $Q$

Send tuple $(M, S)$

Compute $H' = P(S)$ with public key $P$

Compute hash $H = h(M)$ of message $M$

Verify sender by testing $H = H'$
General idea of signature schemes

- Original message $M$
- Hash function $h$: $H = h(M)$
- Signature $S = Q(H)$ with private key $Q$
- Send tuple $(M, S)$

Compute $H' = P(S)$ with public key $P$.

Verify sender by testing $H = H'$. 
General idea of signature schemes

- **original message** $M$
- **hash function** $h$:
  
  $H = h(M)$
- **signature** $S = Q(H)$ with private key $Q$
- **send tuple** $(M, S)$

- Compute hash $H = h(M)$ of message $M$
- Compute $H' = P(S)$ with public key $P$
General idea of signature schemes

- **Original message**: $M$
- **Hash function**: $H = h(M)$
- **Signature**: $S = Q(H)$ with private key $Q$
- **Send tuple**: $(M, S)$
- **Compute hash**: $H = h(M)$ of message $M$
- **Verify sender via testing**: $H = H'$
- **Compute $H' = P(S)$ with public key $P$**
Introduction to public key cryptography

Construction of MPKC

Examples of MPKC schemes: Hidden fields

Examples of MPKC schemes: Oil & Vinegar
Trapdoors of MPKC

PKC depends on the existence of a class of trapdoor one-way functions.
PKC depends on the existence of a class of trapdoor one-way functions.

**Example**

- Elliptic curve crypto depends on the elliptic curve group.
- NTRU depends on the structure of an integral lattice.
PKC depends on the existence of a class of trapdoor one-way functions

**Example**
- Elliptic curve crypto depends on the elliptic curve group.
- NTRU depends on the structure of an integral lattice.

In MPKC the trapdoor one-way function is of the form of a multivariate non-linear polynomial map over a finite field.
PKC depends on the existence of a class of **trapdoor one-way functions**

**Example**

- Elliptic curve crypto depends on the elliptic curve group.
- NTRU depends on the structure of an integral lattice.

In **MPKC** the trapdoor one-way function is of the form of a **multivariate non-linear polynomial map over a finite field**.

**Note**

Usually *non-linear* means *quadratic*. Thus people often speak about **MQ systems** referring to *multivariate quadratic*. 
MQ Problem
Given \( m \) quadratic polynomials

\[ p_1(x_1, \ldots, x_n), \ldots, p_m(x_1, \ldots, x_n) \]

in \( n \) variables \( \mathbf{x} = x_1, \ldots, x_n \) over a finite field \( \mathbb{F}_q \), find a vector \( \mathbf{x}' \) such that

\[ p_1(\mathbf{x}') = \ldots = p_m(\mathbf{x}') = 0. \]
Basis of security of MPKC

**MQ Problem**

Given $m$ quadratic polynomials

\[ p_1(x_1, \ldots, x_n), \ldots, p_m(x_1, \ldots, x_n) \]

in $n$ variables $\mathbf{x} = x_1, \ldots, x_n$ over a finite field $\mathbb{F}_q$, find a vector $\mathbf{x}'$ such that

\[ p_1(\mathbf{x}') = \ldots = p_m(\mathbf{x}') = 0. \]

Solving MQ polynomial systems is worst case NP-hard and in general doubly exponential over any finite field.
Construction of the MPKC

1. Get a trapdoor, e.g. non-linear function $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$, easily invertible.
2. Represent $\mathcal{F}$ as multivariate polynomials $\mathcal{F}$ over $\mathbb{F}_q$. 
Construction of the MPKC

1. Get a trapdoor, e.g. non-linear function $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$, easily invertible.
2. Represent $\mathcal{F}$ as multivariate polynomials $\mathcal{P}$ over $\mathbb{F}_q$.
3. Take invertible secret matrices $S$ and $T$. 
Construction of the MPKC

1. Get a trapdoor, e.g. non-linear function $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$, easily invertible.
2. Represent $\mathcal{F}$ as multivariate polynomials $\mathcal{F}$ over $\mathbb{F}_q$.
3. Take invertible secret matrices $S$ and $T$.
4. Distribute your public key $\mathcal{P} = T \circ \mathcal{F} \circ S$ as polynomials $\mathcal{P}$. 
Public keys of MPKC

We use a MQ polynomial map over $\mathbb{F}_q$: $\mathcal{P}: \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$

$$\mathcal{P} = (p_1(x), \ldots, p_m(x))$$

with

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{ijk} x_i x_j + \sum_i \beta_{ik} x_i + \gamma_k,$$

for $x = (x_1, \ldots, x_n)$, $\alpha_{ijk}, \beta_{ik}, \gamma_k \in \mathbb{F}_q$. 
Public keys of MPKC

We use a MQ polynomial map over $\mathbb{F}_q$: $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$

$$\mathcal{P} = (p_1(x), \ldots, p_m(x))$$

with

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{ijk} x_i x_j + \sum_i \beta_{ik} x_i + \gamma_k,$$

for $x = (x_1, \ldots, x_n)$, $\alpha_{ijk}, \beta_{ik}, \gamma_k \in \mathbb{F}_q$.

**Note**

The constant and linear terms of the $p_k$ do not provide any further security, so we can neglect them in our discussion:
Public keys of MPKC

We use a MQ polynomial map over $\mathbb{F}_q$: $\mathcal{P}: \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$

$$\mathcal{P} = (p_1(x), \ldots, p_m(x))$$

with

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{ijk} x_i x_j + \sum_i \beta_{ik} x_i + \gamma_k,$$

for $x = (x_1, \ldots, x_n)$, $\alpha_{ijk}, \beta_{ik}, \gamma_k \in \mathbb{F}_q$.

**Note**

The constant and linear terms of the $p_k$ do not provide any further security, so we can neglect them in our discussion:

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{ijk} x_i x_j,$$
Public keys of MPKC

In other words: $p_k \longleftrightarrow (n \times n)$ matrix $P_k$

$\left( x_1, \ldots, x_n \right) \cdot P_k \cdot \left( x_1, \ldots, x_n \right)^T$
Public keys of MPKC

In other words: $p_k \leftrightarrow (n \times n)$ matrix $\mathbb{P}_k$

$$(x_1, \ldots, x_n) \cdot \mathbb{P}_k \cdot (x_1, \ldots, x_n)^T$$

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{ijk} x_i x_j = x \mathbb{P}_k x^T$$

Clearly, $\mathbb{P}$ should be a random (mostly dense) system of MQ polynomial equations. But is it really?
Public keys of MPKC

In other words: $p_k \leftrightarrow (n \times n)$ matrix $\Psi_k$

$$\begin{align*}
(x_1, \ldots, x_n) \cdot \Psi_k \cdot (x_1, \ldots, x_n)^T
\end{align*}$$

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{ijk} x_i x_j = x \Psi_k x^T$$

Clearly, $\Psi$ should be a random (mostly dense) system of MQ polynomial equations. But is it really?
Main ideas for attacking MPKC

▶ Try to retrieve secret $S$ and $T$ in order to get $\mathcal{F}$.

▶ The used **MPKC** variant is known, thus try to exploit knowledge of general trapdoor / shape of private map.
Main ideas for attacking MPKC

► Try to retrieve secret $S$ and $T$ in order to get $\Phi$.

► The used **MPKC** variant is known, thus try to exploit knowledge of general trapdoor / shape of private map.

What are possible instantiations of **MPKC**?

or

What settings for $\Phi$, $S$ and $T$ are used?
Introduction to public key cryptography

Construction of MPKC

Examples of MPKC schemes: Hidden fields

Examples of MPKC schemes: Oil & Vinegar
Matsumoto-Imai scheme (MI or $C^*$) – 1988

In general:

- Utilize vector space and hidden field structure of $(\mathbb{F}_q)^n$.

- Instead of searching for invertible maps over the vector space $(\mathbb{F}_q)^n$
  - Look for invertible maps on the extension field $\mathbb{F}_{q^n}$ ($\mathcal{F}$).
  - Transform to an invertible map over $(\mathbb{F}_q)^n$ applying secret $S$ and $T$ ($\mathcal{P}$).

- A single univariate polynomial $\mathcal{F}$ over $\mathbb{F}_{q^n}$ is represented by $n$ multivariate polynomials $\mathcal{P} = (p_i(x_1, \ldots, x_n))_{1 \leq i \leq n}$ over $\mathbb{F}_q$. 

Note

For $C^*$ let us assume $q = 2$ or $q = 2^k$ for some $k$. Makes the following discussion easier, generalization is rather trivial.
Matsumoto-Imai scheme (MI or $C^*$) – 1988

In general:

▶ Utilize vector space and hidden field structure of $(\mathbb{F}_q)^n$.

▶ Instead of searching for invertible maps over the vector space $(\mathbb{F}_q)^n$
  ▷ Look for invertible maps on the extension field $\mathbb{F}_{q^n}$ ($\mathcal{F}$).
  ▷ Transform to an invertible map over $(\mathbb{F}_q)^n$ applying secret $S$ and $T$ ($\mathcal{P}$).

▶ A single univariate polynomial $\mathcal{F}$ over $\mathbb{F}_{q^n}$ is represented by $n$
  multivariate polynomials $\mathcal{P} = (p_i(x_1, \ldots, x_n))_{1 \leq i \leq n}$ over $\mathbb{F}_q$.

Note

For $C^*$ let us assume $q = 2$ or $q = 2^k$ for some $k$. Makes the following discussion easier, generalization is rather trivial.
Matsumoto-Imai scheme (MI or $C^*$) – 1988

In particular:

$\mathbb{E} = \mathbb{F}_q[x]/g(x)$ for an irreducible polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n$. 
Matsumoto-Imai scheme (MI or $C^*$) – 1988

In particular:

- $E = \mathbb{F}_q[x]/g(x)$ for an irreducible polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n$.
- $\phi : E \to (\mathbb{F}_q)^n$ an $\mathbb{F}_q$-linear isomorphism given by

  \[
  \phi(a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}) = (a_0, \ldots, a_{n-1}).
  \]
Matsumoto-Imai scheme (MI or $C^*$) – 1988

In particular:

- $\mathbb{E} = \mathbb{F}_q[x]/g(x)$ for an irreducible polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n$.
- $\phi : \mathbb{E} \to (\mathbb{F}_q)^n$ an $\mathbb{F}_q$-linear isomorphism given by
  $$\phi(a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}) = (a_0, \ldots, a_{n-1}).$$

- Choose $0 < \theta < n$ such that $\gcd(q^\theta + 1, q^n - 1) = 1$. Define map $\mathcal{F}$ in $\mathbb{E}[X]$ via
  $$\mathcal{F}(X) = X^{1+q^\theta}.$$
Matsumoto-Imai scheme (MI or $C^*$) – 1988

In particular:

- $\mathbb{E} = \mathbb{F}_q[x]/g(x)$ for an irreducible polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n$.
- $\phi : \mathbb{E} \to (\mathbb{F}_q)^n$ an $\mathbb{F}_q$-linear isomorphism given by
  \[
  \phi(a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}) = (a_0, \ldots, a_{n-1}).
  \]

- Choose $0 < \theta < n$ such that $\gcd(q^\theta + 1, q^n - 1) = 1$. Define map $\mathcal{F}$ in $\mathbb{E}[X]$ via
  \[
  \mathcal{F}(X) = X^{1+q^\theta}.
  \]

- Choice of $\theta$ ensures $\mathcal{F}$ being invertible:
  \[
  \xi(1 + q^\theta) \equiv 1 \mod (q^n - 1) \implies \mathcal{F}^{-1}(X) = X^\xi.
  \]
Matsumoto-Imai scheme (MI or $C^*$) – 1988

In particular:

- $\mathbb{E} = \mathbb{F}_q[x]/g(x)$ for an irreducible polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n$.
- $\phi : \mathbb{E} \to (\mathbb{F}_q)^n$ an $\mathbb{F}_q$-linear isomorphism given by
  \[ \phi(a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}) = (a_0, \ldots, a_{n-1}). \]

- Choose $0 < \theta < n$ such that $\gcd(q^\theta + 1, q^n - 1) = 1$. Define map $\mathcal{F}$ in
  $\mathbb{E}[X]$ via
  \[ \mathcal{F}(X) = X^{1+q^\theta}. \]

- Choice of $\theta$ ensures $\mathcal{F}$ being invertible:
  \[ \xi(1+q^\theta) \equiv 1 \mod (q^n - 1) \implies \mathcal{F}^{-1}(X) = X^{\xi}. \]

- Go back to $(\mathbb{F}_q)^n$: $\mathcal{P}' = \phi \circ \mathcal{F} \circ \phi^{-1}(x_1, \ldots, x_n) = (p'_1(x), \ldots, p'_n(x))$.
- Apply secret transformations $S$ and $T$:
  \[ \mathcal{P} = T \circ \mathcal{P}' \circ S = T \circ \phi \circ \mathcal{F} \circ \phi^{-1} \circ S. \]
Note 1

Raising $X$ to a power of the form $q^i$ is linear in $E$.

$\Rightarrow$ $\mathcal{P}$ is a system of MQ polynomials over $\mathbb{F}_q$. 
Matsumoto-Imai scheme (MI or $C^*$) – 1988

**Note 1**
Raising $X$ to a power of the form $q^i$ is linear in $E$.
\[ \Rightarrow \mathcal{P} \text{ is a system of MQ polynomials over } \mathbb{F}_q. \]

**Note 2**
There are not so many choices for $\theta$ thus we can assume $\theta$ to be publicly known.
Matsumoto-Imai scheme (MI or $C^*$) – 1988

Broken by Patarin (1995):

\[
Y = X^{q\theta} + 1 \Rightarrow XY^{q\theta} = X^{q\theta^2}Y.
\]

▶ From this one receives a system of equations of type

\[
\sum \alpha_{ij} x_i y_j + \sum \beta_i x_i + \sum \gamma_i y_i + \delta = 0.
\]

▶ Taking enough cipher texts from the original system one can determine the coefficients.

▶ For a given $y$ (cipher text) we can then solve the linear equations to get $x$ (plain text).
Matsumoto-Imai scheme (MI or $C^*$) – 1988

Broken by Patarin (1995):

\[
\text{If } Y = X^{q^\theta + 1} \implies XY^{q^\theta} = X^{q^{2\theta}} Y.
\]
Matsumoto-Imai scheme (MI or $C^*$) – 1988

Broken by Patarin (1995):

If $Y = X^{q^\theta + 1} \Rightarrow XY^{q^\theta} = X^{q^{2\theta}} Y$.

▶ From this one receives a system of equations of type

$$
\sum \alpha_{ij} x_i y_j + \sum \beta_i x_i + \sum \gamma_i y_i + \delta = 0.
$$
Matsumoto-Imai scheme (MI or $C^*$) – 1988

Broken by Patarin (1995):

If $Y = X^{q^\theta} + 1 \Rightarrow XY^{q^\theta} = X^{q^{2\theta}} Y$.

▶ From this one receives a system of equations of type

$$\sum \alpha_{ij} x_i y_j + \sum \beta_i x_i + \sum \gamma_i y_i + \delta = 0.$$ 

▶ Taking enough cipher texts from the original system one can determine the coefficients.
Matsumoto-Imai scheme (MI or $C^*$) – 1988

Broken by Patarin (1995):

\[
\text{If } Y = X^{q^\theta + 1} \Rightarrow XY^{q^\theta} = X^{q^{2\theta}} Y.
\]

- From this one receives a system of equations of type

\[
\sum \alpha_{ij} x_i y_j + \sum \beta_i x_i + \sum \gamma_i y_i + \delta = 0.
\]

- Taking enough cipher texts from the original system one can determine the coefficients.

- For a given $y$ (cipher text) we can then solve the linear equations to get $x$ (plain text).
\((C^*)^{-}\) or SFLASH

**Construction**

Remove \(r\) public polys: 
\[ \mathcal{P} = (p_1(x), \ldots, p_{n-r}(x)) = R \circ T \circ \mathcal{P}' \circ S. \]
(C*) or SFLASH

**Construction**
Remove $r$ public polys: $\mathcal{P} = (p_1(x), \ldots, p_{n-r}(x)) = R \circ T \circ \mathcal{P}' \circ S$.

How to attack this? (Dubious, Fouque, Shamir, Stern; 2007)

- $T$ randomly samples linear space $V$ spanned by the $n$ quadratic equations generated via $\mathcal{P}' \circ S$.

- Only $n - r$ of the $n$ samples $\Rightarrow$ Recover the missing ones?
Construction
Remove $r$ public polys: $\mathcal{P} = (p_1(x), \ldots, p_{n-r}(x)) = R \circ T \circ \mathcal{P}' \circ S$.

How to attack this? (Dubious, Fouque, Shamir, Stern; 2007)

- $T$ randomly samples linear space $V$ spanned by the $n$ quadratic equations generated via $\mathcal{P}' \circ S$.

- Only $n - r$ of the $n$ samples $\Rightarrow$ Recover the missing ones?

- Transform $T$: $T' = T \circ A$ applying matrices $A$.
  $\Rightarrow$ Several samples of $V$, several different linearly independent equations!
(C∗)- or SFLASH

Construction
Remove \( r \) public polys: \( \mathcal{P} = (p_1(x), \ldots, p_{n-r}(x)) = R \circ T \circ \mathcal{P}' \circ S. \)

How to attack this? (Dubious, Fouque, Shamir, Stern; 2007)

- \( T \) randomly samples linear space \( V \) spanned by the \( n \) quadratic equations generated via \( \mathcal{P}' \circ S. \)

- Only \( n - r \) of the \( n \) samples \( \Rightarrow \) Recover the missing ones?

- Transform \( T: T' = T \circ A \) applying matrices \( A. \)
  \( \Rightarrow \) Several samples of \( V, \text{several different linearly independent equations!} \)

- Only have the public polynomials \( \Rightarrow \) How to compute \( T \circ A? \)
Construction
Remove $r$ public polys: $\mathcal{P} = (p_1(x), \ldots, p_{n-r}(x)) = R \circ T \circ \mathcal{P}' \circ S$.

How to attack this? (Dubious, Fouque, Shamir, Stern; 2007)

- $T$ randomly samples linear space $V$ spanned by the $n$ quadratic equations generated via $\mathcal{P}' \circ S$.

- Only $n - r$ of the $n$ samples $\Rightarrow$ Recover the missing ones?

- Transform $T$: $T' = T \circ A$ applying matrices $A$.
  $\Rightarrow$ Several samples of $V$, several different linearly independent equations!

- Only have the public polynomials $\Rightarrow$ How to compute $T \circ A$?

- Multiply input of $\mathcal{F}$ by $\alpha \Rightarrow$ multiply output by $\beta := \mathcal{F}(\alpha) = \alpha^{q^\theta} + 1$. 

$(C^*)^-$ or SFLASH
Construction
Remove $r$ public polys: $\mathcal{P} = (p_1(x), \ldots, p_{n-r}(x)) = R \circ T \circ \mathcal{P}' \circ S$.

How to attack this? (Dubious, Fouque, Shamir, Stern; 2007)

- $T$ randomly samples linear space $V$ spanned by the $n$ quadratic equations generated via $\mathcal{P}' \circ S$.

- Only $n - r$ of the $n$ samples $\Rightarrow$ Recover the missing ones?

- Transform $T$: $T' = T \circ A$ applying matrices $A$.
  $\Rightarrow$ Several samples of $V$, several different linearly independent equations!

- Only have the public polynomials $\Rightarrow$ How to compute $T \circ A$?

- Multiply input of $\mathcal{P}$ by $\alpha$ $\Rightarrow$ multiply output by $\beta := \mathcal{F}(\alpha) = \alpha^{q^\theta} + 1$.

- Find matrix $B$ such that $R \circ T \circ A \circ \mathcal{P}' \circ S = R \circ T \circ \mathcal{P}' \circ S \circ B$. 

($C^*$) or SFLASH
How to get $B$?

\[ W = \text{linear space spanned by all possible quadratic expressions} \]
\[ V = \text{linear space spanned by quadratic expressions of } T \circ \mathcal{P}^I \circ S \]
\[ V_R = \text{linear space spanned by quadratic expressions of } R \circ T \circ \mathcal{P}^I \circ S \]
How to get \( B \)?

\[
\begin{align*}
W &= \text{linear space spanned by all possible quadratic expressions} \\
V &= \text{linear space spanned by quadratic expressions of } T \circ \mathcal{D}' \circ S \\
V_R &= \text{linear space spanned by quadratic expressions of } R \circ T \circ \mathcal{D}' \circ S
\end{align*}
\]

**Example: SFLASHv3**

\[
n = 67, \ r = 11 \Rightarrow \dim(V_R) = 56, \ dim(V) = 67, \ dim(W) = 2278.
\]
How to get $B$?

$W = \text{linear space spanned by all possible quadratic expressions}$

$V = \text{linear space spanned by quadratic expressions of } T \circ \mathcal{D}' \circ S$

$V_R = \text{linear space spanned by quadratic expressions of } R \circ T \circ \mathcal{D}' \circ S$

**Example: SFLASHv3**

$n = 67, r = 11 \Rightarrow \dim(V_R) = 56, \dim(V) = 67, \dim(W) = 2278.$

$\Rightarrow \text{Not so many good choices for } B.$
How to get $B$?

$W =$ linear space spanned by all possible quadratic expressions

$V =$ linear space spanned by quadratic expressions of $T \circ P' \circ S$

$V_R =$ linear space spanned by quadratic expressions of $R \circ T \circ P' \circ S$

**Example: SFLASHv3**

$n = 67, r = 11 \Rightarrow \dim(V_R) = 56, \dim(V) = 67, \dim(W) = 2278.$

$\Rightarrow$ Not so many good choices for $B$.

**In particular**

$\exists q^{n^2}$ possible matrices over $\mathbb{F}_q$ but only $q^n$ elements in $\mathbb{F}$.

$\Rightarrow$ “good matrices” corresponding to extension field multiplication form tiny linear subspace.
(C*)− or SFLASH

**Last step**
Find “good matrices” using the fact that they preserve the membership of the output quadratic equations in $V$. (“Bad matrices” are extremely unlikely to have this property as $V$ is very sparse in $W$."

Either solve quadratic equations in $n^2$ variables, not so efficient.

Or use the differential operator to get bivariate bilinear equations from univariate quadratic ones (working since $F(X) = X^{q^θ} + 1$):

$$DF(a, X) = F(a + X) - F(a) - F(X) + F(0) = aX^{q^θ} + a^{q^θ}X.$$
Last step
Find “good matrices” using the fact that they preserve the membership of the output quadratic equations in \( V \). (“Bad matrices” are extremely unlikely to have this property as \( V \) is very sparse in \( W \).)

- Either solve quadratic equations in \( n^2 \) variables, not so efficient.
- Or use the differential operator to get bivariate bilinear equations from univariate quadratic ones (working since \( \mathcal{F}(X) = X^{q\theta + 1} \)):

\[
D\mathcal{F}(a, X) = \mathcal{F}(a + X) - \mathcal{F}(a) - \mathcal{F}(X) + \mathcal{F}(0)
= aX^{q\theta} + a^{q\theta} X.
\]
Other generalizations of $C^*$

- Generalize $F$ to a univariate polynomial of a given degree $d$ (HFE).

$$
F(X) = \sum \alpha_i X^{q^{\theta_i} + q^{\eta_i}}
$$

\[i, q^{\theta_i} + q^{\eta_i} \leq d\]

$\Rightarrow$ Broken by Faugère and Joux in 2002.

- HFE with removing equations (HFE-).

- Use more than one polynomial for $F$ (multi-$C^*$, multi-HFE).

- Use intermediate field equations (IFS).

- Perturb polynomials or add some auxiliary variable $Y$ (vinegar variable).

- ...
Other generalizations of $C^*$

- Generalize $F$ to a univariate polynomial of a given degree $d$ (HFE).

$$F(X) = \sum_{i, q^{\theta_i} + q^{\eta_i} \leq d} \alpha_i X^{q^{\theta_i} + q^{\eta_i}}$$

$\Rightarrow$ Broken by Faugère and Joux in 2002.

- HFE with removing equations (HFE-).

- Use more than one polynomial for $F$ (multi-$C^*$, multi-HFE).

- Use intermediate field equations (IFS).

- Perturb polynomials or add some auxiliary variable $Y$ (vinegar variable).

- ...
Introduction to public key cryptography

Construction of MPKC

Examples of MPKC schemes: Hidden fields

Examples of MPKC schemes: Oil & Vinegar
Oil and Vinegar scheme (OV)

In general:

- Trapdoor achieved by special structure of private polynomials, not by field extensions.

- Structure given by distinguishing set of variables: \( n = v + o \) variables, \( V := \{u_1, \ldots, u_v\} \) (vinegar) and \( O := \{u_{v+1}, \ldots, u_n\} \) (oil)

- \( V \) and \( O \) are balanced: \( v = o \), \( n = 2v = 2o \).

- There are \( v = o \) private polynomials in \( F \).

- Structure of private polynomials
  \[
  f_k(u) = \sum_{i \in V, j \in V, i \leq j} \alpha_{ijk} u_i u_j + \sum_{i \in V, j \in O} \beta_{ijk} u_i u_j.
  \]

- There are no quadratic terms in two oil variables.
Oil and Vinegar scheme (OV)

In general:

- Trapdoor achieved by special structure of private polynomials, not by field extensions.

- Structure given by distinguishing set of variables: $n = v + o$ variables, $\mathcal{V} := \{u_1, \ldots, u_v\}$ (vinegar) and $\mathcal{O} := \{u_{v+1}, \ldots, u_n\}$ (oil)

- $\mathcal{V}$ and $\mathcal{O}$ are balanced: $v = o$, $n = 2v = 2o$.

- There are $v = o$ private polynomials in $\mathbb{F}$.
Oil and Vinegar scheme (OV)

In general:

▶ Trapdoor achieved by special structure of private polynomials, not by field extensions.

▶ Structure given by distinguishing set of variables: \( n = v + o \) variables, \( \mathcal{V} := \{u_1, \ldots, u_v\} \) (vinegar) and \( \mathcal{O} := \{u_{v+1}, \ldots, u_n\} \) (oil)

▶ \( \mathcal{V} \) and \( \mathcal{O} \) are balanced: \( v = o \), \( n = 2v = 2o \).

▶ There are \( v = o \) private polynomials in \( \mathbb{F} \).

Structure of private polynomials

\[
f_k(u) = \sum_{i \in \mathcal{V}, j \in \mathcal{V}, i \leq j} \alpha_{ijk} u_i u_j + \sum_{i \in \mathcal{V}, j \in \mathcal{O}} \beta_{ijk} u_i u_j.
\]

There are no quadratic terms in two oil variables.
Oil and Vinegar scheme (OV)

In matrix representation the private polynomials look like the following:

\[ \tilde{f}_k = \begin{bmatrix} \mathcal{V} \times \mathcal{V} & \mathcal{V} \times \mathcal{O} \\ \mathcal{O} \times \mathcal{V} & O_{m \times m} \end{bmatrix} \quad \text{for } 1 \leq k \leq o \]
Oil and Vinegar scheme (OV)

In matrix representation the private polynomials look like the following:

\[ \tilde{\delta}_k = \begin{pmatrix} V \times V & V \times O \\ O \times V & O_{m \times m} \end{pmatrix} \text{ for } 1 \leq k \leq o \]

Having a message of length \( v \) we fix variables \( u_1, \ldots, u_v \) in \( \mathbb{F} \) and receive linear equations in the remaining \( o \) variables.
Oil and Vinegar scheme (OV)

In matrix representation the private polynomials look like the following:

\[
\begin{array}{c}
\mathcal{V} \times \mathcal{V} \\
\mathcal{O} \times \mathcal{V} \\
\mathcal{V} \times \mathcal{O} \\
\mathcal{O} \times \mathcal{O}
\end{array}
\]

\[\mathcal{F}_k = \begin{array}{c}
\mathcal{V} \times \mathcal{V} \\
\mathcal{O} \times \mathcal{V} \\
\mathcal{V} \times \mathcal{O} \\
\mathcal{O} \times \mathcal{O}
\end{array} \quad \text{for } 1 \leq k \leq o\]

Having a message of length \( v \) we fix variables \( u_1, \ldots, u_v \) in \( \mathcal{F} \) and receive linear equations in the remaining \( o \) variables.

Those linear equations are invertible with high probability.
Oil and Vinegar scheme (OV)

In matrix representation the private polynomials look like the following:

\[
\begin{array}{ccc}
\mathcal{V} \times \mathcal{V} & \mathcal{V} \times \mathcal{O} \\
\mathcal{O} \times \mathcal{V} & O_{m \times m}
\end{array}
\]

\[\mathbb{3}_k = \text{ for } 1 \leq k \leq o\]

Having a message of length \(v\) we fix variables \(u_1, \ldots, u_v\) in \(\mathcal{F}\) and receive linear equations in the remaining \(o\) variables.

Those linear equations are invertible with high probability.

**Public polynomials**

Apply secret transformations \(S\) and \(T\) to receive \(\mathcal{P}\) as “random” MQ polynomials. In this setting \(T\) does not add any further security, so we can assume the identity.
Attacking OV (Kipnis, Shamir; 1998)

- Try to separate vinegar and oil variables in the public polynomials.
- Search for an equivalent representation $\mathcal{F}' \circ S' = \mathcal{P} = \mathcal{F} \circ S$. 

Efficient algorithms for computing eigenspaces by Kipnis and Shamir
Attacking OV (Kipnis, Shamir; 1998)

- Try to separate vinegar and oil variables in the public polynomials.
- Search for an equivalent representation $F' \circ S' = P = F \circ S$.
- Exploit balance between $v$ and $o$: $v = o$.
- Due to this each $F_i$ maps the oil subspace $u_1 = \ldots = u_v = 0$ to the vinegar subspace $u_{v+1} = \ldots = u_n = 0$. 
Attacking OV (Kipnis, Shamir; 1998)

- Try to separate vinegar and oil variables in the public polynomials.
- Search for an equivalent representation $F' \circ S' = P = F \circ S$.
- Exploit balance between $v$ and $o$: $v = o$.
- Due to this each $F_i$ maps the oil subspace $u_1 = \ldots = u_v = 0$ to the vinegar subspace $u_{v+1} = \ldots = u_n = 0$.
- If $F_j$ is invertible (high probability) then $F_i F_j^{-1}$ maps the vinegar space on itself.
Attacking OV (Kipnis, Shamir; 1998)

- Try to separate vinegar and oil variables in the public polynomials.

- Search for an equivalent representation $F' \circ S' = P = F \circ S$.

- Exploit balance between $v$ and $o$: $v = o$.

- Due to this each $\mathcal{F}_i$ maps the oil subspace $u_1 = \ldots = u_v = 0$ to the vinegar subspace $u_{v+1} = \ldots = u_n = 0$.

- If $\mathcal{F}_j$ is invertible (high probability) then $\mathcal{F}_i \mathcal{F}_j^{-1}$ maps the vinegar space on itself.

- Thus the image $V$ of the vinegar subspace under $S$ is a common eigenspace for each $P_i P_j^{-1}$, $1 \leq i < j \leq o$. 
Attacking OV (Kipnis, Shamir; 1998)

- Try to separate vinegar and oil variables in the public polynomials.
- Search for an equivalent representation $F' \circ S' = P = F \circ S$.
- Exploit balance between $v$ and $o$: $v = o$.
- Due to this each $F_i$ maps the oil subspace $u_1 = \ldots = u_v = 0$ to the vinegar subspace $u_{v+1} = \ldots = u_n = 0$.
- If $F_j$ is invertible (high probability) then $F_i F_j^{-1}$ maps the vinegar space on itself.
- Thus the image $V$ of the vinegar subspace under $S$ is a common eigenspace for each $P_i P_j^{-1}$, $1 \leq i < j \leq o$.
- Efficient algorithms for computing eigenspaces by Kipnis and Shamir
  $\Rightarrow$ Get $V$, find $O$ such that $O + V = \mathbb{F}_q^n$ (separating vinegar and oil).
  $\Rightarrow$ Get $(F', S')$ isomorphic to $(F, S)$. 
► This attack works due to $v = o \Rightarrow \text{“unbalance” } v \text{ and } o: v > o$.
► In general: $n = v + o$, $m = n - v$, $v > m$, $m$ polynomials.
Unbalancing OV – UOV

- This attack works due to \( v = o \) \( \Rightarrow \) “unbalance” \( v \) and \( o \): \( v > o \).

- In general: \( n = v + o, m = n - v, v > m, m \) polynomials.

**Be careful**

The previous attack works in a probabilistic fashion also for \( v > o \).

It has complexity \( O(q^{v-m-1}m^4) = O(q^{n-2m-1}m^4) \). Thus one should take at least \( v \geq 3m \).
Unbalancing OV – UOV

- This attack works due to \( v = o \Rightarrow \text{“unbalance” } v \) and \( o: v > o \).
- In general: \( n = v + o, m = n - v, v > m, m \) polynomials.

Be careful
The previous attack works in a probabilistic fashion also for \( v > o \).
It has complexity \( O(q^{v-m-1}m^4) = O(q^{n-2m-1}m^4) \).
Thus one should take at least \( v \geq 3m \).

Size problem
Enlarging \( v \) and \( o \) the key sizes get, at some point, too big for practical applications.
Why do we need big values for $v$ and $o$?

Due to another attack by Kipnis and Shamir that we can interpret as a Minrank problem (NP-complete).

**Minrank $(n, k, r)$ problem**

Given $M_0, \ldots, M_k \in \mathcal{M}_{n \times n}(\mathbb{F}_q)$, find $(\lambda_1, \ldots, \lambda_k) \in \mathbb{F}_q^k$ such that

$$\text{rank} \left( \sum_{i=1}^{k} \lambda_i M_i - M_0 \right) \leq r.$$
Why do we need big values for $v$ and $o$?

Due to another attack by Kipnis and Shamir that we can interpret as a Minrank problem (NP-complete).

**Minrank $(n, k, r)$ problem**

Given $M_0, \ldots, M_k \in \mathcal{M}_{n \times n}(\mathbb{F}_q)$, find $(\lambda_1, \ldots, \lambda_k) \in \mathbb{F}_q^k$ such that

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

Often the rank of the matrices corresponding to $\mathcal{F}$ is restricted (e.g. in HFE via degree, in UOV by the choice of $v$ and $o$.)
Why do we need big values for $v$ and $o$?

Due to another attack by Kipnis and Shamir that we can interpret as a **Minrank problem** (NP-complete).

**Minrank $(n, k, r)$ problem**

Given $M_0, \ldots, M_k \in \mathcal{M}_{n \times n}(\mathbb{F}_q)$, find $(\lambda_1, \ldots, \lambda_k) \in \mathbb{F}_q^k$ such that

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

Often the rank of the matrices corresponding to $\mathcal{F}$ is restricted (e.g. in HFE via degree, in UOV by the choice of $v$ and $o$.)

$$\Rightarrow \text{rank} \left( \sum_{i=1}^{m} \lambda_i \mathcal{P}_i \right) \leq r.$$
Somewhere over the Rainbow . . .

Try to improve security by using several layers of UOV
Somewhere over the Rainbow . . .

Try to improve security by using several layers of UOV

Rainbow scheme with $L$ layers

- Defined by a tuple $(q, v_1, \ldots, v_{L+1})$

- Preset $0 < v_1 < v_2 < \ldots < v_{L+1} = n$, # vinegar variables of the different layers.

- # oil variables $o_\ell = v_{\ell+1} - v_\ell$ for $1 \leq \ell \leq L$.

- Each layer $\ell$ consists of $o_\ell$ polynomials $f_{v_\ell - v_1 + 1}, \ldots, f_{v_{\ell+1} - v_1}$ where we set $m := \sum_{\ell=1}^{L} o_\ell$.

- For each such level we have vinegar variables $V_\ell = \{u_1, \ldots, u_{v_\ell}\}$ and oil variables $O_\ell = \{u_{v_\ell+1}, \ldots, u_{v_{\ell+1}}\}$.
Rainbow \((q, 6, 12, 17, 22, 33)\)

4-layered Rainbow with
\[
m = (12 - 6) + (17 - 12) + (22 - 17) + (33 - 22) = 27 \text{ polynomials.}
\]

**Classical Minrank attack**

6 polynomials in first layer of rank \(r = 12\).

- \(\text{Prob} (\text{random vector in } \ker (\sum_{i=1}^{m} \lambda_i \mathcal{P}_i)) = \frac{1}{q^r}.\)
- For such a vector \(w\) we have \((\sum_{i=1}^{m} \lambda_i \mathcal{P}_i) w = 0.\) Linear in \(\lambda_1, \ldots, \lambda_m.\) Since \(n > m\) we can just use linear algebra.
- Complexity \(O(q^{r} m^3) = O(q^{12} 27^3).\)
Rainbow \((q, 6, 12, 17, 22, 33)\)

**Improved attack (Billet, Gilbert; 2006)**

For \(1 \leq k \leq 6\):

\[
F_k = w = 6 \implies w_1 = F_1 w_T, \ldots, w_6 = F_6 w_T^\hat{=} = 6
\]

▶ \(\text{Prob}(\text{linearly independent}) = \prod_{i=0}^{5} (1 - q^i q^6) < 1 - q^6\).

▶ \(\text{Prob}(\sum_{i=1}^{6} \lambda_i F_i w = 0) > 1 q^6\).

▶ \(\text{Prob}(\text{random vector of type} w) = 1 q^6\).

Afterwards linear algebra as in standard Minrank attack:

\[O(27^{3^6})\]

\[O(q^{727^{3^6}})\]
Rainbow \((q, 6, 12, 17, 22, 33)\)

**Improved attack (Billet, Gilbert; 2006)**

For \(1 \leq k \leq 6\):

\[
\mathcal{F}_k =
\]

\[
\begin{align*}
1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 & \quad 6 \\
6 & \quad 1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 \\
5 & \quad 6 & \quad 1 & \quad 2 & \quad 3 & \quad 4 \\
4 & \quad 5 & \quad 6 & \quad 1 & \quad 2 & \quad 3 \\
3 & \quad 4 & \quad 5 & \quad 6 & \quad 1 & \quad 2 \\
2 & \quad 3 & \quad 4 & \quad 5 & \quad 6 & \quad 1 \\
1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 & \quad 6
\end{align*}
\]

\[w = \begin{bmatrix} 6 \quad 27 \end{bmatrix}\]

- Probability \((w_i\) linearly independent\) = \(\prod_{i=0}^{5} (1 - q_i q_{6}) < 1 - q\)
- Probability \((\sum_{i=1}^{6} \lambda_i F_i w = 0)\) > \(1 - q\)
- Probability \(\) (random vector of type \(w\)) = \(1 - q\)

Afterwards linear algebra as in standard Minrank attack:

\(O(27^3)\)
Rainbow \((q, 6, 12, 17, 22, 33)\)

Improved attack (Billet, Gilbert; 2006)

For \(1 \leq k \leq 6\) : 
\[
\mathcal{F}_k = \begin{pmatrix}
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\end{pmatrix}
\]

\[
w_1 = \mathcal{F}_1 w^T, \ldots, w_6 = \mathcal{F}_6 w^T
\]

\[
w = \begin{pmatrix}
6 \\
27 \\
\end{pmatrix}
\]

\[
\text{Prob}(\text{random vector of type } w) = \frac{1}{q^6}
\]

\[
\text{Prob}(\sum_{i=1}^{6} \lambda_i F_i w = 0) < 1 - \frac{1}{q^6}
\]

\[
\text{Prob}(w \text{ linearly independent}) = \prod_{i=0}^{5} \left(1 - q^i q^6\right) < 1 - \frac{1}{q^6}
\]

Afterwards linear algebra as in standard Minrank attack:

\[
O \left(27^{3} \right)
\]

\[
O \left(q^{7/27} \right)
\]
Rainbow \((q, 6, 12, 17, 22, 33)\)

**Improved attack (Billet, Gilbert; 2006)**

For \(1 \leq k \leq 6\):

\[
\mathcal{F}_k = w = \begin{bmatrix}
6 \\
27
\end{bmatrix}
\]

\[
\Rightarrow w_1 = \mathcal{F}_1 w^T, \ldots, w_6 = \mathcal{F}_6 w^T \Rightarrow \begin{bmatrix}
6 \\
27
\end{bmatrix}^T
\]

\[
\text{Prob}(w_i \text{ linearly independent}) = \prod_{i=0}^{5} \left( 1 - \frac{q^i}{q^6} \right) < 1 - \frac{1}{q}.
\]
Improved attack (Billet, Gilbert; 2006)

For $1 \leq k \leq 6$:

$$F_k = w = 6^{27} \Rightarrow w_1 = F_1 w^T, \ldots, w_6 = F_6 w^T \iff T^{6^{27}}$$

- $\text{Prob}(w_i \text{ linearly independent}) = \prod_{i=0}^{5} \left( 1 - \frac{q^i}{q^6} \right) < 1 - \frac{1}{q}$.
- $\text{Prob}(\left( \sum_{i=1}^{6} \lambda_i \mathcal{F}_i \right) w = 0) > \frac{1}{q}$. 

Rainbow $(q, 6, 12, 17, 22, 33)$
Rainbow \((q, 6, 12, 17, 22, 33)\)

**Improved attack (Billet, Gilbert; 2006)**

For \(1 \leq k \leq 6\) : \(\mathcal{F}_k = \)

\[\Rightarrow w_1 = \mathcal{F}_1 w^T, \ldots, w_6 = \mathcal{F}_6 w^T \Rightarrow w = 6 \quad 27\]

- \(\text{Prob}(w_i \text{ linearly independent}) = \prod_{i=0}^{5} \left(1 - \frac{q^i}{q^6}\right) < 1 - \frac{1}{q}\).
- \(\text{Prob}\left((\sum_{i=1}^{6} \lambda_i \mathcal{F}_i) w = 0\right) > \frac{1}{q}\).
- \(\text{Prob}(\text{random vector of type } w) = \frac{1}{q^6}\).
- Afterwards linear algebra as in standard Minrank attack: \(O\left(27^3\right)\)
Rainbow \((q, 6, 12, 17, 22, 33)\)

**Improved attack (Billet, Gilbert; 2006)**

For \(1 \leq k \leq 6 : \mathcal{F}_k = \)

\[ w_1 = \mathcal{F}_1 w^T, \ldots, w_6 = \mathcal{F}_6 w^T \]

\[ w = \begin{bmatrix} 6 \\ 27 \end{bmatrix} \]

\(\Rightarrow\)

\[ \operatorname{Prob}(w_i \text{ linearly independent}) = \prod_{i=0}^{5} \left( 1 - \frac{q^i}{q^6} \right) < 1 - \frac{1}{q}. \]

\[ \operatorname{Prob}\left( \left( \sum_{i=1}^{6} \lambda_i \mathcal{F}_i \right) w = 0 \right) > \frac{1}{q}. \]

\[ \operatorname{Prob}(\text{random vector of type } w) = \frac{1}{q^6} \]

\[ \text{Afterwards linear algebra as in standard Minrank attack: } O\left(27^3\right) \]

\[ O\left(q^7 27^3\right) \]
Rainbow (2^8, 18, 30, 42)

\[\tilde{\delta}_k = \begin{cases} \mathbb{O}_{18 \times 12} & \text{for } 1 \leq k \leq 12 \\ \mathbb{O}_{12 \times 12} & \text{for } 13 \leq k \leq 24 \end{cases}\]

(First layer)

\[\tilde{\delta}_k = \begin{cases} \mathbb{O}_{12 \times 12} & \text{for } 1 \leq k \leq 12 \\ \mathbb{O}_{12 \times 12} & \text{for } 13 \leq k \leq 24 \end{cases}\]

(Second layer)
Direct key recovery attack

- 24 “random” public polynomials, need to find $S \in \text{Mat}(42 \times 42, \mathbb{F}_q)$ and $T \in \text{Mat}(24 \times 24, \mathbb{F}_q)$.

- Structure of $\mathcal{F} \implies$ System of polynomial equations in entries of $S$ and $T$ for corresponding zero coefficients.

- $(v_1 + o_1 + o_2)^2 + (o_1 + o_2)^2 = 42^2 + 24^2 = 2340$ variables.

- $(o_1 + o_2) \cdot |\mathcal{O}_2 \times \mathcal{O}_2| + o_1 \cdot (|\mathcal{V}_1 \cup \mathcal{O}_1 \times \mathcal{O}_2|) + o_1 \cdot (|\mathcal{O}_1 \times \mathcal{O}_1|) = 7128$ (non-linear) equations.

- Complexity: $O(2^{3608})$ (or: forget it!)
Rainbow $(2^8, 18, 30, 42)$

Again idea of equivalent keys
Rainbow \((2^8, 18, 30, 42)\)

Again idea of **equivalent keys**

\[
S' = \begin{pmatrix}
1_{18 \times 18} & O_{12 \times 12} & O_{12 \times 18} & O_{12 \times 12} \\
O_{12 \times 12} & 1_{12 \times 12} & O_{12 \times 12} & O_{12 \times 12} \\
O_{12 \times 18} & O_{12 \times 12} & 1_{12 \times 12} & O_{12 \times 12} \\
O_{12 \times 12} & O_{12 \times 12} & O_{12 \times 12} & 1_{12 \times 12}
\end{pmatrix}
\]
Rainbow \((2^8, 18, 30, 42)\)

Again idea of equivalent keys

\[
S' = 
\begin{array}{ccc}
1_{18 \times 18} & & \\
O_{12 \times 12} & 1_{12 \times 12} & \\
O_{12 \times 18} & O_{12 \times 12} & 1_{12 \times 12}
\end{array}
\]

\[
T' = 
\begin{array}{ccc}
1_{12 \times 12} & & \\
O_{12 \times 12} & 1_{12 \times 12} & \\
O_{12 \times 12} & O_{12 \times 12} & 1_{12 \times 12}
\end{array}
\]

**Equivalent key attack** \((T \circ F \circ S = P = T' \circ F' \circ S')\)

- 24 “random” public polynomials, need to find \(S' \in \text{Mat}(42 \times 42, \mathbb{F}_q)\) and \(T' \in \text{Mat}(24 \times 24, \mathbb{F}_q)\).

- \(v_1 \cdot o_1 + v_1 \cdot o_2 + o_1 \cdot o_2 + o_1 \cdot o_2 = 720\) variables.

- # equations stays the same, but \(o_1 \times |V_1 \times O_2|\) are no longer cubic, but quadratic (first \(v_1\) variables in \(S'\) are set!)

- 2592 quadratic (bihomogeneous in \(s_{ij}\) and \(t_{kl}\)) and 4536 cubic equations.

- Complexity: \(O(2^{374})\) (or: still, forget it!)
Rainbow \((2^8, 18, 30, 42)\)

Remove information: good keys resp. \textbf{Rainbow} band separation attack
Rainbow \((2^8, 18, 30, 42)\)

Remove information: good keys resp. Rainbow band separation attack

\[ S_n'' = \]

\[ T_1'' = \]

\[ S_n'' = \]

\[ T_1'' = \]
Rainbow $(2^8, 18, 30, 42)$

Remove information: good keys resp. **Rainbow band separation attack**

We can only recover zero coefficients for $x_n^2$ terms (for all private polynomials) and for $x_k x_n$ terms (for $1 \leq k \leq n - 1$ and only for first private polynomial).
Rainbow \((2^8, 18, 30, 42)\)

Remove information: good keys resp. Rainbow band separation attack

![Diagram]

We can only recover zero coefficients for \(x_n^2\) terms (for all private polynomials) and for \(x_kx_n\) terms (for \(1 \leq k \leq n - 1\) and only for first private polynomial).

- \((v_1 + o_1) + o_2 = 42\) variables
- \((n, n, 1)\) 1 cubic equation.
- \((n, n, k)\) \(o_1 + o_2 - 1\) quadratic equations for \(2 \leq k \leq o_1 + o_2\)
- \((k, n, 1)\) \(v_1 + o_1 + o_2 - 1\) quadratic equations for \(1 \leq k \leq n - 1\)
Rainbow \((2^8, 18, 30, 42)\)

Remove information: **good keys resp. Rainbow band separation attack**

\[
S''_n = \begin{array}{ccc}
1_{18\times18} & O_{18\times12} & O_{18\times11} \\
O_{12\times18} & 1_{12\times12} & O_{12\times11} \\
O_{12\times12} & O_{12\times12} & 1_{12\times12}
\end{array}
\quad T''_1 = \begin{array}{c}
1_{12\times12} \\
O_{12\times12} \\
O_{12\times12} \\
1_{12\times12}
\end{array}
\]

We can only recover zero coefficients for \(x_n^2\) terms (for all private polynomials) and for \(x_k x_n\) terms (for \(1 \leq k \leq n - 1\) and only for first private polynomial).

- \((v_1 + o_1) + o_2 = 42\) variables
- \((n, n, 1)\) 1 cubic equation.
- \((n, n, k)\) \(o_1 + o_2 - 1\) quadratic equations for \(2 \leq k \leq o_1 + o_2\)
- \((k, n, 1)\) \(v_1 + o_1 + o_2 - 1\) quadratic equations for \(1 \leq k \leq n - 1\)

\[\implies 42\] variables in \(65\) equations ?
Rainbow \((2^8, 18, 30, 42)\)

**Metacryptography**
Not breaking a system, but breaking an attack :)
Rainbow \((2^8, 18, 30, 42)\)

**Metacryptography**

Not breaking a system, but breaking an attack :)

We can recover zero coefficients for \(x_n^2\) terms only for the last \(o_2\) private polynomials

and for \(x_k x_n\) terms only for \(1 \leq k \leq v_1\) and only for the first private polynomial.
Rainbow \((2^8, 18, 30, 42)\)

**Metacryptography**
Not breaking a system, but breaking an attack :)

We can recover zero coefficients for \(x_n^2\) terms only for the last \(o_2\) private polynomials and for \(x_k x_n\) terms only for \(1 \leq k \leq v_1\) and only for the first private polynomial.

- \((v_1 + o_1) + o_2 = 42\) variables
- \((n, n, 1)\) 1 cubic equation.
- \((n, n, k)\) \(o_2\) quadratic equations for \(o_1 + 1 \leq k \leq o_1 + o_2\)
- \((k, n, 1)\) \(v_1\) quadratic equations for \(1 \leq k \leq v_1\)

\[\Rightarrow\] 42 variables in 31 equations.
Rainbow \((2^8, 18, 30, 42)\)

Fix attack by adding more rows to \(T''\):
Fix attack by adding more rows to $T''$:

- Minimal system received with 3 rows in $T''$ and $S''_n$.
- $v_1 + o_1 + 3 \cdot o_2 = 66$ variables.
- 3 cubic equations, 66 quadratic equations.
- **Complexity**: Still $O(2^{231})$ (or: well, you know)
Rainbow \((2^8, 18, 30, 42)\)

**Metametacryptography**
Not breaking a system, but breaking the attack on an attack :)

\[v_1 + o_1 = 42 \text{ variables}\]

\[(n, n, 1) \text{ cubic equation.}\]

\[(n, n, k) o_1 + o_2 - 1 \text{ quadratic equations for } 2 \leq k \leq o_1 + o_2\]

\[(k, n, 1) v_1 + o_1 + o_2 - 1 \text{ quadratic equations for } 1 \leq k \leq n - 1\]

\[65 \text{ equations in } 42 \text{ variables}\]

**Complexity:** \(O(2^{95})\)

Still not feasible, but now we have new ideas.
Rainbow \((2^8, 18, 30, 42)\)

**Metametacryptography**

Not breaking a system, but breaking the attack on an attack :)  

The Rainbow band separation attack is really working out!
Rainbow \((2^8, 18, 30, 42)\)

**Metametacryptography**
Not breaking a system, but breaking the attack on an attack :)

The **Rainbow** band separation attack is really working out!

- \((v_1 + o_1) + o_2 = 42\) variables
- \((n, n, 1)\) 1 cubic equation.
- \((n, n, k)\) \(o_1 + o_2 - 1\) quadratic equations for \(2 \leq k \leq o_1 + o_2\)
- \((k, n, 1)\) \(v_1 + o_1 + o_2 - 1\) quadratic equations for \(1 \leq k \leq n - 1\)
- 65 equations in 42 variables
- Complexity: \(O(2^{95})\)
**Rainbow** \((2^8, 18, 30, 42)\)

**Metametacryptography**
Not breaking a system, but breaking the attack on an attack :)

The **Rainbow** band separation attack is really working out!

- \((v_1 + o_1) + o_2 = 42\) variables
- \((n, n, 1)\) \(1\) cubic equation.
- \((n, n, k)\) \(o_1 + o_2 - 1\) quadratic equations for \(2 \leq k \leq o_1 + o_2\)
- \((k, n, 1)\) \(v_1 + o_1 + o_2 - 1\) quadratic equations for \(1 \leq k \leq n - 1\)
- 65 equations in 42 variables
- **Complexity:** \(O(2^{95})\)

Still not feasible, but now we have new ideas.
References


