Multivariate Quadratic Public-Key Cryptography In the NIST Competition

Bo-Yin Yang

Academia Sinica

Oxford PQCRYPTO Workshop 2019
Oxford Mathematical Institute
Tuesday, 19.03.2019
MPKC: Multivariate (Quadratic) Public Key Cryptosystem

Public Key: System of nonlinear multivariate equations

\[
p^{(1)}(w_1, \ldots, w_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(1)} \cdot w_i w_j + \sum_{i=1}^{n} p_i^{(1)} \cdot w_i (p_0^{(1)})
\]

\[
p^{(2)}(w_1, \ldots, w_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(2)} \cdot w_i w_j + \sum_{i=1}^{n} p_i^{(2)} \cdot w_i (p_0^{(2)})
\]

\[\vdots\]

\[
p^{(m)}(w_1, \ldots, w_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(m)} \cdot w_i w_j + \sum_{i=1}^{n} p_i^{(m)} \cdot w_i (p_0^{(m)})
\]
MPKC: Multivariate (Quadratic) Public Key Cryptosystem
Public Key: System of nonlinear multivariate equations

\[ p^{(1)}(w_1, \ldots, w_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(1)} \cdot w_i w_j + \sum_{i=1}^{n} p_i^{(1)} \cdot w_i \left( + p_0^{(1)} \right) \]

\[ p^{(2)}(w_1, \ldots, w_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(2)} \cdot w_i w_j + \sum_{i=1}^{n} p_i^{(2)} \cdot w_i \left( + p_0^{(2)} \right) \]

\[ \vdots \]

\[ p^{(m)}(w_1, \ldots, w_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(m)} \cdot w_i w_j + \sum_{i=1}^{n} p_i^{(m)} \cdot w_i \left( + p_0^{(m)} \right) \]

If degree \( d \) then Public Key size = \( m \binom{n + d}{d} \), hence usually \( d = 2 \).
Security

The security of multivariate schemes is based on the

**Problem MQ**: Given $m$ multivariate quadratic polynomials $p^{(1)}, \ldots, p^{(m)}$, find a vector $w = (w_1, \ldots, w_n)$ such that $p^{(1)}(w) = \ldots = p^{(m)}(w) = 0$.

- NP hard
- believed to be hard on average (even for quantum computers):
Security

The security of multivariate schemes is based on the

Problem MQ: Given $m$ multivariate quadratic polynomials $p^{(1)}, \ldots, p^{(m)}$, find a vector $w = (w_1, \ldots, w_n)$ such that $p^{(1)}(w) = \ldots = p^{(m)}(w) = 0$.

- NP hard
- believed to be hard on average (even for quantum computers):
  suppose we have a probabilistic Turing Machine $T$ and a subexponential function $\eta$, $T$ terminates with an answer to a random $MQ(n, m = an, \mathbb{F}_q)$ instance in time $\eta(n)$ with probability $\text{negl}(n)$.
- higher order versions (MP for Multivariate Polynomials or PoSSo for Polynomial System Solving) clearly no less hard

However usually no direct reduction to MQ !! There are exceptions:
## Identification Scheme of Sakumoto et al and MQDSS

### An example 5-pass ID scheme depending only on MQ

- \( \mathcal{P} \) be a set of random MQ polynomials
- Its “polar” form \( D\mathcal{P}(x, y) := \mathcal{P}(x + y) - \mathcal{P}(x) - \mathcal{P}(y) - \mathcal{P}(0) \)
- \( \mathcal{P}(s) = p \) is the public key, \( s \) is the secret.
- Peter picks and commits random \((r_0, t_0, e_0)\), sets \( r_1 = s - r_0 \) and commits \((r_1, D\mathcal{P}(t_0, r_1) + e_0)\).
- Vera sends random \( \alpha \),
- Peter sets and sends \( t_1 := \alpha r_0 - t_0, e_1 := \alpha \mathcal{P}(r_0) - e_0 \).
- Vera sends challenge \( Ch \), Peter sends \( r_{Ch} \).
- Vera checks the commit of either \((r_0, \alpha r_0 - t_1, \alpha \mathcal{P}(r_0) - e_1)\) or \((r_1, \alpha(p - \mathcal{P}(r_1)) - D\mathcal{P}(t_1, r_1) - e_1)\).

The Fiat-Shamir transform of this ID scheme is the MQDSS scheme.
Bipolar Construction

- Easily invertible quadratic map $Q : \mathbb{F}^n \to \mathbb{F}^m$
- Two invertible linear maps $\mathcal{T}(\mathbb{F}^m \to \mathbb{F}^m)$ and $\mathcal{S}(\mathbb{F}^n \to \mathbb{F}^n)$
- **Public key**: $\mathcal{P} = \mathcal{T} \circ Q \circ \mathcal{S}$ supposed to look random
- **Private key**: $\mathcal{S}$, $Q$, $\mathcal{T}$ allows to invert the public key
Bipolar Construction

- Easily invertible quadratic map $Q : \mathbb{F}^n \rightarrow \mathbb{F}^m$
- Two invertible linear maps $T : \mathbb{F}^m \rightarrow \mathbb{F}^m$ and $S : \mathbb{F}^n \rightarrow \mathbb{F}^n$
- Public key: $\mathcal{P} = T \circ Q \circ S$ supposed to look random
- Private key: $S, Q, T$ allows to invert the public key

Encryption Schemes ($m \geq n$)

- Triangular schemes, ZHFE (broken)
- PMI+, IPHFE+
- Simple Matrix (not highly thought of)
Bipolar Construction

- Easily invertible quadratic map $Q : \mathbb{F}^n \rightarrow \mathbb{F}^m$
- Two invertible linear maps $T : \mathbb{F}^m \rightarrow \mathbb{F}^m$ and $S : \mathbb{F}^n \rightarrow \mathbb{F}^n$
- Public key: $P = T \circ Q \circ S$ supposed to look random
- Private key: $S$, $Q$, $T$ allows to invert the public key

Encryption Schemes ($m \geq n$)

- Triangular schemes, ZHFE (broken)
- PMI+, IPHFE+
- Simple Matrix (not highly thought of)

Signature Schemes ($m \leq n$)

- Unbalanced Oil and Vinegar
  - Rainbow (TTS)
- HFEv- (QUARTZ/Gui)
- pFLASH
## NIST Candidates

### Digital Signature Schemes (4 into second round)
- Transformed Zero-Knowledge: MQDSS
- HFEv-: GUI, GeMSS, DualModeMS
- Small Field: Rainbow, L(ifted)UOV, HiMQ3 (a version of TTS)

### Encryption Schemes
- SRTP1 (broken)
- DME (dubious)
- CFPKM (Polly Cracker)
Workflow

Decryption / Signature Generation

\[ z \in \mathbb{F}^m \xrightarrow{\mathcal{T}^{-1}} y \in \mathbb{F}^m \xrightarrow{Q^{-1}} x \in \mathbb{F}^n \xrightarrow{S^{-1}} w \in \mathbb{F}^n \]

Encryption / Signature Verification

B.-Y. Yang (Academia Sinica)
Multivariate Cryptography
Oxford 19.03.19 7/41
Isomorphism of Polynomials

Due to the bipolar construction, the security of MPKCs is also based on the

**Problem EIP** (Extended Isomorphism of Polynomials): Given the public key $P$ of a multivariate public key cryptosystem, find affine maps $\tilde{S}$ and $\tilde{T}$ as well as quadratic map $\tilde{Q}$ in class $C$ such that $P = \tilde{T} \circ \tilde{Q} \circ \tilde{S}$.

$\Rightarrow$ Hardness of problem depends much on the structure of the central map

$\Rightarrow$ Often EIP is really (a not so hard) MinRank

$\Rightarrow$ In general, not much is known about the complexity

$\Rightarrow$ Security analysis of multivariate schemes is a hard task
Generic (Direct) Attacks

Try to solve the public equation \( P(w) = z \) as an instance of the MQ-Problem, all algorithms have exponential running time (for \( m \approx n \))

**Known Best Generic Algorithms**

- For larger \( q \), FXL (“Hybridized XL” can Groverize)
- For \( q = 2 \), smart enumerative methods

**Complexity of Direct Attacks**

<table>
<thead>
<tr>
<th>Security Level (bit)</th>
<th>Number of Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>( F_{2^8} )</td>
</tr>
<tr>
<td>88</td>
<td>( F_{2^9} )</td>
</tr>
<tr>
<td>30</td>
<td>( F_{2^{10}} )</td>
</tr>
<tr>
<td>28</td>
<td>( F_{2^{11}} )</td>
</tr>
<tr>
<td>26</td>
<td>( F_{2^{12}} )</td>
</tr>
<tr>
<td>100</td>
<td>( F_{2^{16}} )</td>
</tr>
<tr>
<td>110</td>
<td>( F_{2^{17}} )</td>
</tr>
<tr>
<td>39</td>
<td>( F_{2^{256}} )</td>
</tr>
<tr>
<td>36</td>
<td>( F_{2^{265}} )</td>
</tr>
<tr>
<td>33</td>
<td>( F_{2^{275}} )</td>
</tr>
<tr>
<td>128</td>
<td>( F_{2^{32}} )</td>
</tr>
<tr>
<td>140</td>
<td>( F_{2^{33}} )</td>
</tr>
<tr>
<td>51</td>
<td>( F_{2^{34}} )</td>
</tr>
<tr>
<td>48</td>
<td>( F_{2^{35}} )</td>
</tr>
<tr>
<td>43</td>
<td>( F_{2^{36}} )</td>
</tr>
<tr>
<td>256</td>
<td>( F_{2^{512}} )</td>
</tr>
<tr>
<td>280</td>
<td>( F_{2^{520}} )</td>
</tr>
<tr>
<td>110</td>
<td>( F_{2^{521}} )</td>
</tr>
<tr>
<td>103</td>
<td>( F_{2^{522}} )</td>
</tr>
<tr>
<td>93</td>
<td>( F_{2^{523}} )</td>
</tr>
</tbody>
</table>

* depending on how we model the Joux-Vitse algorithm
Generic (Direct) Attacks

Try to solve the public equation $P(w) = z$ as an instance of the MQ-Problem, all algorithms have exponential running time (for $m \approx n$)

Known Best Generic Algorithms

- For larger $q$, FXL (“Hybridized XL” can Groverize)
- For $q = 2$, Joux-Vitse’s XL-with-enumeration Variant.

Complexity of Direct Attacks

How many equations are needed to meet given levels of security?

<table>
<thead>
<tr>
<th>security level (bit)</th>
<th>$F_2$</th>
<th>$F_{16}$</th>
<th>$F_{31}$</th>
<th>$F_{256}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>88</td>
<td>30</td>
<td>28</td>
<td>26</td>
</tr>
<tr>
<td>100</td>
<td>110</td>
<td>39</td>
<td>36</td>
<td>33</td>
</tr>
<tr>
<td>128</td>
<td>140</td>
<td>51</td>
<td>48</td>
<td>43</td>
</tr>
<tr>
<td>192</td>
<td>208</td>
<td>80</td>
<td>75</td>
<td>68</td>
</tr>
<tr>
<td>256</td>
<td>280</td>
<td>110</td>
<td>103</td>
<td>93</td>
</tr>
</tbody>
</table>

* depending on how we model the Joux-Vitse algorithm
XL Algorithm (Lazard, 1983; CKPS, 1999)

Given: nonlinear polynomials \( f_1, \ldots, f_m \) of degree \( d \)

1. **eXtend** multiply each polynomial \( f_1, \ldots, f_m \) by every monomial of degree \( \leq D - d \)

2. **Linearize**: Apply (sparse) linear algebra to solve the extended system

\[
\text{Complexity} = 3 \cdot \binom{n + d_{\text{XL}}}{d_{\text{XL}}}^2 \cdot \binom{n}{d} \quad \text{(for larger } q)\]

or

3. or **Linearize and use an improved XL**: Many variants...
**XL Variants**

**FXL – XL with \( k \) variables guessed or “hybridized”**

if with \( k \) initial guesses / fixing / ”hybridization”:

\[
\text{Complexity} = \min_k 3q^k \cdot \left( n - k + d_{XL} \right)^2 \cdot \binom{n-k}{d}.
\]

[generic method with the best asymptotic multiplicative complexity].
XL Variants

FXL – XL with \( k \) variables guessed or “hybridized”

Joux-Vitse (“Hybridized XL-related method”)

1. **eXtend:** multiply each polynomial \( f_1, \ldots, f_m \) by monomials, up to total degree \( \leq D \)

2. **Linearize:** Apply linear algebra to eliminate all monomials of total degree \( \geq 2 \) in the first \( k \) variables (and get at least \( k \) such equations).

3. **Fix** \( n - k \) variables, solve for the initial \( k \) in linear equations.
XL Variants

FXL – XL with $k$ variables guessed or “hybridized”

Joux-Vitse (“Hybridized XL-related method”)

1. **eXtend**: multiply each polynomial $f_1, \ldots, f_m$ by monomials, up to total degree $\leq D$
2. **Linearize**: Apply linear algebra to eliminate all monomials of total degree $\geq 2$ in the first $k$ variables (and get at least $k$ such equations).
3. **Fix** $n - k$ variables, solve for the initial $k$ in linear equations.

XL2 – simplified $F_4$

1. **eXtend**: multiply each polynomial $f_1, \ldots, f_m$ by monomials, up to total degree $\leq D$
2. **Linearize**: Apply linear algebra to eliminate top level monomials
3. Multiply degree $D - 1$ equations by variables, **Eliminate Again**.
More Advanced Gröbner Bases Algorithms

- find a “nice” basis of the ideal \( \langle f_1, \ldots, f_m \rangle \)
- first studied by B. Buchberger
- later improved by Faugère et al. (\( F_4, F_5 \))
- With linear algebra constant \( 2 < \omega \leq 3 \).

\[
\text{Complexity}(q, m, n) = O\left(\left(\frac{n + d_{\text{reg}} - 1}{d_{\text{reg}}}\right)^\omega\right) \quad \text{(for larger } q)\]

- Can also be “Hybridized”:

\[
\text{Complexity}(q, m, n) = \min_k q^k \cdot O\left(\left(\frac{n - k + d_{\text{reg}} - 1}{d_{\text{reg}}}\right)^\omega\right)
\]

- Runs at the same degree as XL2.

**Do not blithely set \( \omega = 2 \) here**

Even if \( \omega \to 2 \), there is a huge constant factor which cannot be neglected.
Remarks

Every cryptosystem can be represented as a set of nonlinear multivariate equations

- Direct attacks can be used in the cryptanalysis of other cryptographic schemes (in particular block and stream ciphers)
- The MQ (or PoSSo) Problem can be seen as one of the central problems in cryptography

Post-Quantum-ness of MQ

- A Grover attack against $n$-bit-input MQ takes $2^{\frac{n}{2}+1} n^3$ time.
- A Hybridized XL with Grover for enumeration on $n$ boolean variables and as many equations still takes $2^{(0.471+o(1))n}$ in true (time-area) cost
## Features of Multivariate Cryptosystems

### Advantages
- resistant against attacks with quantum computers
- reasonably fast
- only simple arithmetic operations required
  - can be implemented on low cost devices
  - suitable for security solutions for the IoT
- many practical signature schemes (UOV, Rainbow, HFEv-, ...)
- short signatures (e.g. 120 bit signatures for 80 bit security)

### Disadvantages
- large key sizes (public key size \( \sim 10 - 100 \) kB)
- no security proofs
- mainly restricted to digital signatures
Big Field Schemes

Decryption / Signature Generation

Encryption / Signature Verification
Extension Fields

- $\mathbb{F}_q$: finite field with $q$ elements
- $g(X)$ irreducible polynomial in $\mathbb{F}[X]$ of degree $n$
  $\Rightarrow \mathbb{F}_{q^n} \cong \mathbb{F}[X]/\langle g(X) \rangle$ finite field with $q^n$ elements
- isomorphism $\phi : \mathbb{F}_q^n \rightarrow \mathbb{F}_{q^n}$, $(a_1, \ldots, a_n) \mapsto \sum_{i=1}^n a_i \cdot X^{i-1}$
- Addition in $\mathbb{F}_{q^n}$: Addition in $\mathbb{F}_q[X]$
- Multiplication in $\mathbb{F}_{q^n}$: Multiplication in $\mathbb{F}_q[X]$ modulo $g(X)$
The Matsumoto-Imai Cryptosystem (1988) or $C^*$

- $F_q$: finite field of characteristic 2
- degree $n$ extension field $E = F_{q^n}$
- isomorphism $\phi : F_q^n \rightarrow E$
- $C^*$ parameter $\theta \in \mathbb{N}$ with

$$\gcd(q^\theta + 1, q^n - 1) = 1.$$ 

**Key Generation**

- *central map* $Q : E \rightarrow E$, $X \mapsto X^{q^\theta+1}$ \Rightarrow $Q$ is bijective
- choose 2 invertible linear or affine maps $S, T : F^n \rightarrow F^n$
- *public key*: $P = T \circ \phi^{-1} \circ Q \circ \phi \circ S : F^n \rightarrow F^n$ quadratic multivariate map
- use the extended Euclidian algorithm to compute $h \in \mathbb{N}$ with

$$h \cdot \theta \equiv 1 \mod q^n - 1$$ 

- *private key*: $S, T$
Linearization Attack against $C^*$

Given public key $P$, $z^* \in \mathbb{F}^n$, find plaintext $w^* \in \mathbb{F}^n$, s.t. $P(w^*) = z^*$

Proposed by J. Patarin in 1995

Taking the $q^\theta - 1$st power of $Y = X^{q^\theta+1}$ and multiplying with $XY$ yields

$$X \cdot Y^{q^\theta} = X^{q^2\theta} \cdot Y$$

$\Rightarrow$ bilinear equation in $X$ and $Y$, hence, same in $w$ and $z$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{ij} w_i z_j + \sum_{i=1}^{n} \beta_i w_i + \sum_{j=1}^{n} \gamma_j z_j + \delta = 0. \quad (\star)$$

1. Compute $N \geq \frac{(n+1) \cdot (n+2)}{2}$ pairs $(z^{(k)}/w^{(k)})$ and substitute into $(\star)$.
2. Solve the resulting linear system for the coefficients $\alpha_{ij}$, $\beta_i$, $\gamma_j$ and $\delta$.

$\Rightarrow$ $n$ bilinear equations in $w_1, \ldots, w_n, z_1, \ldots, z_n$

3. Substitute $z^*$ into these bilinear equations and solve for $w^*$. 
pFLASH: Prefixed $C^*$ signature scheme

Natural restriction of $Q$ to hyperplane $=$ set coordinate to 0

Start from a $C^*$ scheme with $Q(x) = x^{1+q^\theta}$ with secret linear maps $S$ and $T$. Let $r$ and $s$ be two integers between 0 and $n$. Let $T^-$ be the projection of $T$ on the last $r$ coordinates and $S^-$ be the restriction of $S$ to the first $n-s$ coordinates. $P = T^− \circ Q \circ S^−$ is the public key and $S^−1$ and $T^−1$ are the secret key. This is pFLASH$(\mathbb{F}_q, n-s, n-r)$.

Inversion

To find $P^−1(m)$ for $m \in \mathbb{F}_q^{n-r}$, the legitimate user first pads $m$ randomly into vector $m' \in (\mathbb{F})^n$ and compute $T^−1 \circ Q^−1 \circ S^−1(m')$. Repeat until this element has its last $s$ coordinates to 0. Its $n-s$ first coordinates are a valid signature for $m$. When $r > s$, the process ends with probability 1 and costs on average $q^s$ inversions of $Q$.

pFLASH Parameters at NIST Cat. I-II

Suggested pFLASH$(\mathbb{F}_{16},96-1,64)$ (146 kB pubkey, 6 kB prvkey).
The HFE Cryptosystem

- “Hidden Field Equations”, proposed by Patarin in 1995
- BigField Scheme, can be used both for encryption and signatures
- finite field $\mathbb{F}$, extension field $\mathbb{E}$ of degree $n$, isomorphism $\phi: \mathbb{F}^n \rightarrow \mathbb{E}$

Original HFE

- central map $Q: \mathbb{E} \rightarrow \mathbb{E}$ (not bijective, invert using Berlekamp Algorithm).

$$Q(X) = \sum_{0 \leq i \leq j}^{q_i + q_j \leq D} \alpha_{ij} X^{q_i+q_j} + \sum_{i=0}^{q_i \leq D} \beta_i \cdot X^{q_i} + \gamma$$

$$\Rightarrow \bar{Q} = \phi^{-1} \circ Q \circ \phi: \mathbb{F}^n \rightarrow \mathbb{F}^n \text{ quadratic}$$

- degree bound $D$ needed for efficient decryption / signature generation
- linear maps $S, T: \mathbb{F}^n \rightarrow \mathbb{F}^n$
- public key: $\mathcal{P} = T \circ \bar{Q} \circ S: \mathbb{F}^n \rightarrow \mathbb{F}^n$
- private key: $S, Q, T$
MinRank Attack against HFE

Look in extension field $\mathbb{E}$ (Kipnis and Shamir [KS99])

- the linear maps $S$ and $T$ relate to univariate maps $S^*(X) = \sum_{i=1}^{n-1} s_i \cdot X^{q^i}$ and $T^*(X) = \sum_{i=1}^{n-1} t_i \cdot X^{q^i}$, with $s_i, t_i \in \mathbb{E}$.
- the public key $\mathcal{P}^*$ can be expressed as $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p^*_{ij} X^{q^i+q^j} = X \cdot \mathcal{P}^* \cdot X^T$,
- Components of $\mathcal{P}^*$ can be found by polynomial interpolation.
- Solve MinRank problem over $\mathbb{E}$.

No need to look in $\mathbb{E}$ (Bettale et al)

Perform the MinRank attack without recovering $\mathcal{P}^* \Rightarrow$ HFE can be broken by using a MinRank problem over the base field $\mathbb{F}$.

$$\text{Complexity}_{\text{MinRank}} = \binom{n + r}{r}^\omega$$

with $2 < \omega \leq 3$ and $r = \lceil \log_q(D - 1) \rceil + 1$. 
Direct Attacks

- J-C Faugère solved HFE Challenge 1 (HFE over GF2, \( d = 96 \)) in 2002
- Empirically HFE systems can be solved much faster than random
- Ding-Hodges Upper bound for \( d_{reg} \)

\[
\begin{align*}
d_{reg} & \leq \begin{cases} 
\frac{(q-1) \cdot (r-1)}{2} + 2 & q \text{ even and } r \text{ odd,} \\
\frac{(q-1) \cdot r}{2} + 2 & \text{otherwise.}
\end{cases}
\end{align*}
\]

with \( r = \left\lceil \log_q (D - 1) \right\rceil + 1 \).

\( \Rightarrow \) Basic version of HFE is not secure

Variant Schemes

- Encryption Schemes IPHFE+ (inefficient), ZHFE (broken).
- Signature Schemes HFEv- (QUARTZ/GUI), MHFEv- (broken)
HFEv-

- finite field $\mathbb{F}$, extension field $\mathbb{E}$ of degree $n$, isomorphism $\phi : \mathbb{F}^n \rightarrow \mathbb{E}$
- central map $Q : \mathbb{F}^v \times \mathbb{E} \rightarrow \mathbb{E}$, where the $\beta_i$ and $\gamma$ are affine.

$$Q(X) = \sum_{0 \leq i \leq j} \alpha_{ij} X^{q^i+q^j} + \sum_{i=0}^{q^i \leq D} \beta_i(v_1, \ldots, v_v) \cdot X^{q^i} + \gamma(v_1, \ldots, v_v)$$

$\Rightarrow \bar{Q} = \phi^{-1} \circ Q \circ (\phi \times \text{id}_v)$ quadratic map: $\mathbb{F}^{n+v} \rightarrow \mathbb{F}^n$

- linear maps $T : \mathbb{F}^n \rightarrow \mathbb{F}^{n-a}$ and $S : \mathbb{F}^{n+v} \rightarrow \mathbb{F}^{n+v}$ of maximal rank
- public key: $P = T \circ \bar{Q} \circ S : \mathbb{F}^{n+v} \rightarrow \mathbb{F}^{n-a}$
- private key: $S, Q, T$

**Signing Message digest $z$**

1. Compute $y = T^{-1}(z) \in \mathbb{F}^n$ and $Y = \phi(y) \in \mathbb{E}$
2. Choose random values for the vinegar variables $v_1, \ldots, v_v$
   Solve $Q_{v_1,\ldots,v_v}(X) = Y$ over $\mathbb{E}$
   Can Repeat first step of Berlekamp until there is a unique solution.
3. Compute $x = \phi^{-1}(X) \in \mathbb{F}^n$ and signature $w = S^{-1}(x||v_1||\ldots||v_v)$. 

B.-Y. Yang (Academia Sinica)
Security vs. Efficiency

Main Attacks

- **MinRank Attack** \( \text{Rank}(F) = r + a + v \)

  \[ \Rightarrow \text{Compl}_{\text{MinRank}} = \left( \begin{array}{c} n + r + a + v \\ r + a + v \end{array} \right)^\omega \]

- **Direct attack** [DY13]

  \[ d_{reg} \leq \begin{cases} \frac{(q-1) \cdot (r+a+v-1)}{2} + 2 & \text{if } q \text{ even and } r + a \text{ odd,} \\ \frac{(q-1) \cdot 2}{2} \cdot (r+a+v) + 2 & \text{otherwise.} \end{cases} \]

  with \( r = \lfloor \log_q(D - 1) \rfloor + 1 \) and \( 2 < \omega \leq 3 \).

Efficiency

Rate determining step: solving \( X \) from a univariate equation of degree \( D \).

\[ \text{Complexity}_{\text{Berlekamp}} = \mathcal{O}(D^3 + n \cdot D^2) \]
How to define a HFEv- like scheme over $\mathbb{F}_2$ [PCY+15]? 

### Collision Resistance of the hash function

To cover a hash value of $k$ bit, the public key of a pure HFEv- scheme has to contain at least $k$ equations over $\mathbb{F}_2$. \( \Rightarrow \) public key $> k^3/2$ bits

<table>
<thead>
<tr>
<th>security level</th>
<th>80</th>
<th>100</th>
<th>128</th>
<th>192</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td># equations</td>
<td>100</td>
<td>200</td>
<td>256</td>
<td>384</td>
<td>512</td>
</tr>
<tr>
<td>pubkey size (kB)</td>
<td>$&gt;250$</td>
<td>$&gt;500$</td>
<td>$&gt;1000$</td>
<td>$&gt;3000$</td>
<td>$&gt;8000$</td>
</tr>
</tbody>
</table>

#### QUARTZ

- standardized by Courtois, Patarin in 2002
- HFEv$^-$ with $\mathbb{F} = \text{GF}(2)$, $n = 103$, $D = 129$, $a = 3$ and $v = 4$
- public key: quadratic map $\mathcal{P} = \mathcal{T} \circ \mathcal{Q} \circ \mathcal{S} : \text{GF}(2)^{107} \rightarrow \text{GF}(2)^{100}$
- Prevent birthday attacks \( \Rightarrow \) Generate four HFEv$^-$ signatures (for $w$, $\mathcal{H}(w|00)$, $\mathcal{H}(w|01)$ and $\mathcal{H}(w|11)$)
- Combine them to a single signature of length $(n - a) + 4 \cdot (a + v) = 128$ bit
**GeMSS, GUI (Generalized QUARTZ) Signature Generation**

**Input:** HFEv- private key \((S, Q, T)\) message \(d\), repetition factor \(k\)

**Output:** signature \(\sigma \in \mathbb{F}_2^{(n-a)+k(a+v)}\)

1. \(h \leftarrow \text{Hash}(d)\)
2. \(S_0 \leftarrow 0 \in \mathbb{GF}(2)^{n-a}\)
3. for \(i = 1\) to \(k\) do
   4. \(D_i \leftarrow \text{first } n - a \text{ bits of } h\)
   5. \((S_i, X_i) \leftarrow \text{HFEv}^{-1}(D_i \oplus S_{i-1})\)
   6. \(h \leftarrow \text{Hash}(h)\)
4. end for
5. \(\sigma \leftarrow (S_k||X_k||\ldots||X_1)\)
6. return \(\sigma\)

Note that if any equation has zero (or more than 2 solutions for Gui), then we discard those vinegars and try again.
Signature Verification

**Input:** HFEv- public key $\mathcal{P}$, message $\mathbf{d}$, repetition factor $k$, signature $\sigma \in \mathbb{F}_2^{(n-a)+k(a+v)}$

**Output:** TRUE or FALSE

1. $\mathbf{h} \leftarrow \text{Hash}(\mathbf{d})$
2. $(S_k, X_k, \ldots, X_1) \leftarrow \sigma$
3. for $i = 1$ to $k$ do
4. $D_i \leftarrow \text{first } n-a \text{ bits of } \mathbf{h}$
5. $\mathbf{h} \leftarrow \text{Hash}(\mathbf{h})$
6. end for
7. for $i = k-1$ to 0 do
8. $S_i \leftarrow \mathcal{P}(S_{i+1} || X_{i+1}) \oplus D_{i+1}$
9. end for
10. if $S_0 = 0$ then
11. return TRUE
12. else
13. return FALSE
14. end if
Parameters for HFEv- (GeMSS, GUI) over $\mathbb{F}_2$?

Parameters are set by the complexity of MinRank and direct attacks
- For the complexity of the MinRank attack we have a concrete formula
- For the direct attack, we only have an upper bound on $d_{\text{reg}}$.

$$d_{\text{reg}} \leq \begin{cases} \frac{(q-1)(r+a+v-1)}{2} + 2 & q \text{ even and } r + a \text{ odd,} \\ \frac{(q-1)(r+a+v)}{2} + 2 & \text{otherwise.} \end{cases} \quad (\star)$$

Experiments show that these estimate for $d_{\text{reg}}$ is reasonably tight.

---

**Parameter Choice of HFEv- over $\mathbb{F}_2$**

**Aggressive $\Rightarrow$ Choose $D$ as small as possible (GUI)**
- $D = 5 \Rightarrow r = \lceil \log_2(D - 1) \rceil + 1 = 3$
- $D = 9 \Rightarrow r = \lceil \log_2(D - 1) \rceil + 1 = 4$
- $D = 17 \Rightarrow r = \lceil \log_2(D - 1) \rceil + 1 = 5$

Increase $a$ and $v$ ($0 \leq v - a \leq 1$) to reach the required security level.
Conservate choice: choose $D = 513$ and $n$ as needed (GeMSS).
Quantum Attacks and Impact

A determined multivariate system of $m$ equations over $\mathbb{F}_2$ can be solved using $2^{m/2} \cdot 2 \cdot m^3$ operations using a quantum computer.

- This does not affect signatures in general because the hashes are typically twice as wide as the design security.
- Alas, this wipes out much of GUI’s gains.

⇒ very large public key size

<table>
<thead>
<tr>
<th>security level</th>
<th>80</th>
<th>100</th>
<th>128</th>
<th>192</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>min # equations</td>
<td>117</td>
<td>155</td>
<td>208</td>
<td>332</td>
<td>457</td>
</tr>
</tbody>
</table>

Proposed Parameters (Signature includes 128-bit salt)

<table>
<thead>
<tr>
<th>NIST Category level (bit)</th>
<th>Parameters $\mathbb{F}_q, n, D, a, v, k$</th>
<th>public key size (kB)</th>
<th>private key size (kB)</th>
<th>signature size (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Gui ($\mathbb{F}_2, 184, 33, 16, 16, 2$)</td>
<td>416.3</td>
<td>19.1</td>
<td>360</td>
</tr>
<tr>
<td>III</td>
<td>Gui ($\mathbb{F}_2, 312, 129, 24, 20, 2$)</td>
<td>1,955.1</td>
<td>59.3</td>
<td>504</td>
</tr>
<tr>
<td>V</td>
<td>Gui ($\mathbb{F}_2, 448, 513, 32, 28, 2$)</td>
<td>5,789.2</td>
<td>155.9</td>
<td>664</td>
</tr>
</tbody>
</table>
Quantum Attacks and Impact

A determined multivariate system of \( m \) equations over \( \mathbb{F}_2 \) can be solved using \( 2^{m/2} \cdot 2 \cdot m^3 \) operations using a quantum computer.

- This does not affect signatures in general because the hashes are typically twice as wide as the design security.
- Alas, this wipes out much of GUI’s gains.

\[ \Rightarrow \text{very large public key size} \]

<table>
<thead>
<tr>
<th>security level</th>
<th>80</th>
<th>100</th>
<th>128</th>
<th>192</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>min # equations</td>
<td>117</td>
<td>155</td>
<td>208</td>
<td>332</td>
<td>457</td>
</tr>
</tbody>
</table>

Proposed Parameters (Signature includes 128-bit salt)

<table>
<thead>
<tr>
<th>NIST Category level (bit)</th>
<th>Parameters ( \mathbb{F}_q, n, D, \Delta, v, nb_ite )</th>
<th>public key size (kB)</th>
<th>private key size (kB)</th>
<th>signature size (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>GeMSS ( (\mathbb{F}_2, 174, 513, 12, 12, 4) )</td>
<td>417</td>
<td>14.5</td>
<td>384</td>
</tr>
<tr>
<td>III</td>
<td>GeMSS ( (\mathbb{F}_2, 265, 513, 22, 20, 4) )</td>
<td>1,304</td>
<td>40.3</td>
<td>704</td>
</tr>
<tr>
<td>V</td>
<td>GeMSS ( (\mathbb{F}_2, 354, 513, 30, 33, 4) )</td>
<td>3,604</td>
<td>83.7</td>
<td>832</td>
</tr>
</tbody>
</table>
HFEv- - Summary

- short signatures
- security well respected
- conflict between security and efficiency
- restricted to very small fields, hence very large keys
- 109M cycles keygen, 676M cycles signing, about 107k cycles verifying at NIST Cat. 1.
Oil-Vinegar Polynomials [Patarin 1997]

Let $\mathbb{F}$ be a (finite) field. For $o, v \in \mathbb{N}$ set $n = o + v$ and define

$$
p(x_1, \ldots, x_n) = \sum_{i=1}^{v} \sum_{j=i}^{v} \alpha_{ij} \cdot x_i \cdot x_j + \sum_{i=1}^{v} \sum_{j=v+1}^{n} \beta_{ij} \cdot x_i \cdot x_j + \sum_{i=1}^{n} \gamma_i \cdot x_i + \delta
$$

$x_1, \ldots, x_v$: Vinegar variables $x_{v+1}, \ldots, x_n$: Oil variables, no $o \times o$ terms.

If we randomly set $x_1, \ldots, x_v$, result is linear in $x_{v+1}, \ldots, x_n$

(Unbalanced) Oil-Vinegar matrix

$\tilde{p}$ the homogeneous quadratic part of $p(x_1, \ldots, x_n)$ can be written as quadratic form $\tilde{p}(x) = x^T \cdot M \cdot x$ with

$$
M = \begin{pmatrix}
*_{v \times v} & *_{o \times v} \\
*_{v \times o} & 0_{o \times o}
\end{pmatrix}
$$

where $*$ denotes arbitrary entries subject to symmetry.
Kipnis-Shamir OV attack when $o = \nu$

\[
\mathcal{O} := \{ x \in \mathbb{F}^n : x_1 = \ldots = x_\nu = 0 \} \quad \text{“Oilspace”}
\]

\[
\mathcal{V} := \{ x \in \mathbb{F}^n : x_{\nu+1} = \ldots = x_n = 0 \} \quad \text{“Vinegarspace”}
\]

Let $E, F$ be invertible “OV-matrices”, i.e. $E, F = \begin{pmatrix} * & * \\ * & 0 \end{pmatrix}$ Then

\[
E \cdot \mathcal{O} \subset \mathcal{V}.
\]

Since the two has the same rank, equality holds, so $(F^{-1} \cdot E) \cdot \mathcal{O} = \mathcal{O}$, i.e. $\mathcal{O}$ is an invariant subspace of $F^{-1} \cdot E$.

**Common Subspaces**

Let $H_i$ be the matrix representing the homogeneous quadratic part of the $i$-th public polynomial. Then we have $H_i = S^T \cdot E_i \cdot S$, i.e. $S^{-1}(\mathcal{O})$ is an invariant subspace of the matrix $(H_j^{-1} \cdot H_i)$, and we find $S^{-1}$.

**tl;dr Summary of the Standard UOV Attack**

- for $\nu \leq o$, breaks the balanced OV scheme in polynomial time.
- For $\nu > o$ the complexity of the attack is about $q^{\nu-o} \cdot o^4$.

$\Rightarrow$ Choose $\nu \approx 2 \cdot o$ (unbalanced Oil and Vinegar (UOV)) [KP99]
Other Attacks

- **Collision Attack**: \( o \geq \frac{2^{2\ell}}{\log_2(q)} \) for \( \ell \)-bit security.

- **Direct Attack**: Try to solve the public equation \( P(w) = z \) as an instance of the MQ-Problem. The public systems of UOV behave much like random systems, but they are highly underdetermined \((n = 3 \cdot m)\)

**Result** [Thomae]: A multivariate system of \( m \) equations in \( n = \omega \cdot m \) variables can be solved in the same time as a determined system of \( m - \lfloor \omega \rfloor + 1 \) equations.

\( \Rightarrow \) \( m \) has to be increased by 2.
Other Attacks

- **Collision Attack**: $o \geq \frac{2^2\ell}{\log_2(q)}$ for $\ell$-bit security.

- **Direct Attack**: Try to solve the public equation $P(w) = z$ as an instance of the MQ-Problem. The public systems of UOV behave much like random systems, but they are highly underdetermined $(n = 3 \cdot m) \Rightarrow m$ has to be increased by 2.

- **UOV-Reconciliation attack**: Try to find a linear transformation $S$ ("good keys") which transforms the public matrices $H_i$ into the form of UOV matrices

\[
(S^T)^{-1} \cdot H_i \cdot S^{-1} = \begin{pmatrix} * & * \\ * & 0 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix}
\]

$\Rightarrow$ Each Zero-term yields a quadratic equation in the elements of $S$. $\Rightarrow$ $S$ can be recovered by solving several MQ systems (the hardest with $v$ variables, $m$ equations if $v < m$).
# Summary of UOV

## Safe Parameters for $UOV(\mathbb{F}, \sigma, \nu)$

<table>
<thead>
<tr>
<th>security level (bit)</th>
<th>scheme</th>
<th>public key size (kB)</th>
<th>private key size (kB)</th>
<th>hash size (bit)</th>
<th>signature (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>$UOV(\mathbb{F}_{16}, 40, 80)$</td>
<td>144.2</td>
<td>135.2</td>
<td>160</td>
<td>480</td>
</tr>
<tr>
<td></td>
<td>$UOV(\mathbb{F}_{256}, 27, 54)$</td>
<td>89.8</td>
<td>86.2</td>
<td>216</td>
<td>648</td>
</tr>
<tr>
<td>100</td>
<td>$UOV(\mathbb{F}_{16}, 50, 100)$</td>
<td>280.2</td>
<td>260.1</td>
<td>200</td>
<td>600</td>
</tr>
<tr>
<td></td>
<td>$UOV(\mathbb{F}_{256}, 34, 68)$</td>
<td>177.8</td>
<td>168.3</td>
<td>272</td>
<td>816</td>
</tr>
<tr>
<td>128</td>
<td>$UOV(\mathbb{F}_{16}, 64, 128)$</td>
<td>585.1</td>
<td>538.1</td>
<td>256</td>
<td>768</td>
</tr>
<tr>
<td></td>
<td>$UOV(\mathbb{F}_{256}, 45, 90)$</td>
<td>409.4</td>
<td>381.8</td>
<td>360</td>
<td>1,080</td>
</tr>
<tr>
<td>192</td>
<td>$UOV(\mathbb{F}_{16}, 96, 192)$</td>
<td>1,964.3</td>
<td>1,786.7</td>
<td>384</td>
<td>1,152</td>
</tr>
<tr>
<td></td>
<td>$UOV(\mathbb{F}_{256}, 69, 138)$</td>
<td>1,464.6</td>
<td>1,344.0</td>
<td>552</td>
<td>1,656</td>
</tr>
<tr>
<td>256</td>
<td>$UOV(\mathbb{F}_{16}, 128, 256)$</td>
<td>4,644.1</td>
<td>4,200.3</td>
<td>512</td>
<td>1,536</td>
</tr>
<tr>
<td></td>
<td>$UOV(\mathbb{F}_{256}, 93, 186)$</td>
<td>3,572.9</td>
<td>3,252.2</td>
<td>744</td>
<td>2,232</td>
</tr>
</tbody>
</table>

## What we know today about UOV

- unbroken since 1999 $\Rightarrow$ high confidence in security
- not the fastest multivariate scheme
- very large keys, (comparably) large signatures
Rainbow Digital Signature

Ding and Schmidt, 2004

- Patented by Ding (May have had patent by T.-T. Moh, expired)
- TTS is its variant with sparse central map
Rainbow Digital Signature

Ding and Schmidt, 2004

- Finite field $\mathbb{F}$, integers $0 < v_1 < \cdots < v_u < v_{u+1} = n$.
- Set $V_i = \{1, \ldots, v_i\}$, $O_i = \{v_i + 1, \ldots, v_{i+1}\}$, $o_i = v_{i+1} - v_i$.
- Central map $Q$ consists of $m = n - v_1$ polynomials $f^{v_1+1}, \ldots, f^{(n)}$ of the form
  \[
  f^{(k)} = \sum_{i,j \in V_{\ell}} \alpha_{ij}^{(k)} x_i x_j + \sum_{i \in V_{\ell}, j \in O_{\ell}} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in V_{\ell} \cup O_{\ell}} \gamma_i^{(k)} x_i + \delta^{(k)},
  \]
  with coefficients $\alpha_{ij}^{(k)}$, $\beta_{ij}^{(k)}$, $\gamma_i^{(k)}$ and $\delta^{(k)}$ randomly chosen from $\mathbb{F}$ and $\ell$ being the only integer such that $k \in O_{\ell}$.
- Choose randomly two affine (or linear) transformations $T : \mathbb{F}^m \to \mathbb{F}^m$ and $S : \mathbb{F}^n \to \mathbb{F}^n$.
- public key: $P = T \circ Q \circ S : \mathbb{F}^n \to \mathbb{F}^m$
- private key: $T$, $Q$, $S$
Idea of Rainbow

Inversion of the central map

- Invert the single UOV layers recursively.
- Use the variables of the $i$-th layer as Vinegars of the $i+1$-th layer.

Illustration: Rainbow with two layers

\[
F(k) = \begin{cases} 
- v_1 & \text{if } v_1 + 1 \leq k \leq v_2 \\
- v_2 & \text{if } v_2 + 1 \leq k \leq n \\
- n & \text{else}
\end{cases}
\]
Idea of Rainbow

Inversion of the central map

- Invert the single UOV layers recursively.
- Use the variables of the $i$-th layer as Vinegars of the $i + 1$-th layer.

**Input:** Rainbow central map $Q = (f^{(v_1+1)}, \ldots, f^{(n)})$, vector $y \in \mathbb{F}^m$.

**Output:** vector $x \in \mathbb{F}^n$ with $Q(x) = y$.

1. Choose random values for the variables $x_1, \ldots, x_{v_1}$ and substitute these values into the polynomials $f^{(i)}$ ($i = v_1 + 1, \ldots, n$).
2. **for** $\ell = 1$ to $u$ **do**
3. Perform Gaussian Elimination on the polynomials $f^{(i)}$ ($i \in O_\ell$) to get the values of the variables $x_i$ ($i \in O_\ell$).
4. Substitute the values of $x_i$ ($i \in O_\ell$) into the polynomials $f^{(i)}$ ($i = v_{\ell+1} + 1, \ldots, n$).
5. **end for**
Idea of Rainbow

Inversion of the central map
- Invert the single UOV layers recursively.
- Use the variables of the $i$-th layer as Vinegars of the $i + 1$-th layer.

Signature Generation from message $d$

1. Use a hash function $\mathcal{H} : \{0, 1\} \rightarrow \mathbb{F}^m$ to compute $z = \mathcal{H}(d) \in \mathbb{F}^m$
2. Compute $y = T^{-1}(z) \in \mathbb{F}^m$.
3. Compute a pre-image $x \in \mathbb{F}^n$ of $y$ under the central map $Q$
4. Compute the signature $w \in \mathbb{F}^n$ by $w = S^{-1}(x)$. 
Idea of Rainbow

Inversion of the central map
- Invert the single UOV layers recursively.
- Use the variables of the $i$-th layer as Vinegars of the $i + 1$-th layer.

Signature Generation from message $d$

1. Use a hash function $\mathcal{H} : \{0, 1\} \rightarrow \mathbb{F}^m$ to compute $z = \mathcal{H}(d) \in \mathbb{F}^m$
2. Compute $y = T^{-1}(z) \in \mathbb{F}^m$.
3. Compute a pre-image $x \in \mathbb{F}^n$ of $y$ under the central map $Q$
4. Compute the signature $w \in \mathbb{F}^n$ by $w = S^{-1}(x)$.

Signature Verification from message $d$, signature $z \in \mathbb{F}^n$

1. Compute $z = \mathcal{H}(d)$.
2. Compute $z' = P(w)$.

Accept the signature $z \iff w' = w$. 
Security

Rainbow is an extension of UOV
⇒ All attacks against UOV can be used against Rainbow, too.

Additional structure of the central map allows several new attacks

- **MinRank Attack**: Look for linear combinations of the matrices $H_i$ of low rank (complexity $q^{v_1} o_1 (m^3/3 + mn^2)$).
- **HighRank Attack**: Look for the linear representation of the variables appearing the lowest number of times in the central polynomials. (Complexity $q^{o_u} o_u (n^3/3 + o_u n^2)$, can Groverize)
- **Rainbow-Band-Separation Attack**: Variant of the UOV-Reconciliation Attack using the additional Rainbow structure

Choosing Parameter Selection for Rainbow is interesting
MinRank Attack

Minors Version
Set all rank $r + 1$ minors of $\sum_i \alpha_i H_i$ to 0.

Kernel Vector Guessing Version
- Guess a vector $v$, let $\sum_i \alpha_i H_i v = 0$, hope to find a non-trivial solution.
- (If $m > n$, guess $\lceil \frac{m}{n} \rceil$ vectors.)
- Takes $q^r (m^3/3 + mn^2)$ time to find a rank $r$ kernel.

Accumulation of Kernels and Effective Rank
In the first stage of Rainbow, there are $o_1 = v_2 - v_1$ equations and $v_2$ variables. The rank should be $v_2$. But if your guess corresponds to $x_1 = x_2 = \cdots = x_{v_1} = 0$, then about $1/q$ of the time we find a kernel. The easy way to see this is that there are $q^{o_1-1}$ different kernels. We say that “effectively the rank is $v_1 + 1$.”
Rainbow Band Separation

Extension to UOV reconciliation to use the special Rainbow form.

\( n \) variables, \( n + m - 1 \) quadratic equations

1. Let \( w_i := w'_i - \lambda_i w'_n \) for \( i \leq v \), \( w_i = w'_i \) for \( i > v \). Evaluate \( z \) in \( w' \).

2. Find \( m \) equations by letting all \((w'_n)^2\) terms vanish; there are \( v \) of \( \lambda_i \)'s.

3. Set all cross-terms involving \( w'_n \) in
   \[ z_1 - \sigma^{(1)}_1 z_{v+1} - \sigma^{(1)}_2 z_{v+2} - \cdots - \sigma^{(1)}_o z_m \]
   to be zero and find \( n - 1 \) more equations.

4. Solve \( m + n - 1 \) quadratic equations in \( o + v = n \) unknowns.

5. Repeat, e.g. next set \( w'_i := w''_i - \lambda_i w''_{n-1} \) for \( i < v \), and let every
   \((w''_{n-1})^2\) and \( w''_n w''_{n-1} \) term be 0. Also set
   \[ z_2 - \sigma^{(2)}_1 z_{v+1} - \sigma^{(2)}_2 z_{v+2} - \cdots - \sigma^{(2)}_o z_m \]
   to have a zero second-to-last column. [\( 2m + n - 2 \) equations in \( n \) unknowns.]
Rainbow - Summary

- no weaknesses found since 2007
- efficient (25.5kcycles verifying, 75.5kcycles signing at NIST Cat. 1)
- suitable for low cost devices
- shorter signatures and smaller key sizes than UOV

Parameters for Rainbow

<table>
<thead>
<tr>
<th>NIST Security Category</th>
<th>parameters $F, , v_1, , o_1, , o_2$</th>
<th>public key size (kB)</th>
<th>private key size (kB)</th>
<th>hash size (bit)</th>
<th>signature (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>$F_{16}, 32, 32, 32$</td>
<td>148.5</td>
<td>97.9</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>III</td>
<td>$F_{256}, 68, 36, 36$</td>
<td>703.9</td>
<td>525.2</td>
<td>576</td>
<td>1,248</td>
</tr>
<tr>
<td>V</td>
<td>$F_{256}, 92, 48, 48$</td>
<td>1,683.3</td>
<td>1,244.4</td>
<td>768</td>
<td>1,632</td>
</tr>
</tbody>
</table>
Thank you for Listening

That’s it Folks!