The NIST Competition and Introduction to Multivariate Quadratic Public-Key Cryptography

Bo-Yin Yang

Academia Sinica

PQCrypto Mini-School
Academia Sinica
Monday, July 20, 2020

Multivariate Cryptography

MPKC: Multivariate (Quadratic) Public Key Cryptosystem
Public Key: System of nonlinear multivariate equations

$$
\begin{aligned}
p^{(1)}(w_1,\ldots,w_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(1)}\cdot w_i w_j \quad + \quad \sum_{i=1}^{n} p_i^{(1)}\cdot w_i \left(+p_0^{(1)}\right)\\
p^{(2)}(w_1,\ldots,w_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(2)}\cdot w_i w_j \quad + \quad \sum_{i=1}^{n} p_i^{(2)}\cdot w_i \left(+p_0^{(2)}\right)\\
&\vdots\\
p^{(m)}(w_1,\ldots,w_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(m)}\cdot w_i w_j \quad + \quad \sum_{i=1}^{n} p_i^{(m)}\cdot w_i \left(+p_0^{(m)}\right)
\end{aligned}
$$

Multivariate Cryptography

MPKC: Multivariate (Quadratic) Public Key Cryptosystem
Public Key: System of nonlinear multivariate equations

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

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

$$\vdots$$

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

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 $\mathbf{w} = (w_1, \ldots, w_n)$ such that $p^{(1)}(\mathbf{w}) = \ldots = p^{(m)}(\mathbf{w}) = 0$.

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

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 $\mathbf{w} = (w_1, \ldots, w_n)$ such that $p^{(1)}(\mathbf{w}) = \ldots = p^{(m)}(\mathbf{w}) = 0$.

- NP hard
- believed to be hard on average (even for quantum conputers): suppose we have a probabilistic Turing Machine T and a subexponential function η, T terminates with an answer to a random $MQ(n, m = an, \mathbb{F}_q)$ instance in time $\eta(n)$ with probability $\mathrm{negl}(n)$.
- 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 $D\mathcal{P}(\mathbf{x},\mathbf{y}) := \mathcal{P}(\mathbf{x}+\mathbf{y}) - \mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{y}) - \mathcal{P}(\mathbf{0})$
- $\mathcal{P}(\mathbf{s}) = \mathbf{p}$ is the public key, $\mathbf{s}$ is the secret.
- Peter picks and commits random $(\mathbf{r_0},\mathbf{t_0},\mathbf{e_0})$, sets $\mathbf{r_1} = \mathbf{s} - \mathbf{r_0}$ and commits $(\mathbf{r_1}, D\mathcal{P}(\mathbf{t_0},\mathbf{r_1}) + \mathbf{e_0})$.
- Vera sends random α,
- Peter sets and sends $\mathbf{t_1} := \alpha\mathbf{r_0} - \mathbf{t_0}$, $\mathbf{e_1} := \alpha\mathcal{P}(\mathbf{r_0}) - \mathbf{e_0}$.
- Vera sends challenge Ch, Peter sends $\mathbf{r}_{Ch}$.
- Vera checks the commit of either $(\mathbf{r_0}, \alpha\mathbf{r_0} - \mathbf{t_1}, \alpha\mathcal{P}(\mathbf{r_0}) - \mathbf{e_1})$ or $(\mathbf{r_1}, \alpha(\mathbf{p} - \mathcal{P}(\mathbf{r_1})) - D\mathcal{P}(\mathbf{t_1},\mathbf{r_1}) - \mathbf{e_1})$.

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

Bipolar Construction

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

Bipolar Construction

- Easily invertible quadratic map $\mathcal{Q} : \mathbb{F}^n \to \mathbb{F}^m$
- 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*: $\mathcal{P} = \mathcal{T} \circ \mathcal{Q} \circ \mathcal{S}$ supposed to look random
- *Private key*: $\mathcal{S}$, $\mathcal{Q}$, $\mathcal{T}$ 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 $\mathcal{Q} : \mathbb{F}^n \to \mathbb{F}^m$
- 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*: $\mathcal{P} = \mathcal{T} \circ \mathcal{Q} \circ \mathcal{S}$ supposed to look random
- *Private key*: $\mathcal{S}$, $\mathcal{Q}$, $\mathcal{T}$ 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

- SRTPI (broken)
- DME (dubious)
- CFPKM (Polly Cracker)

Workflow

Decryption / Signature Generation

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

$$\mathbf{w} \in \mathbb{F}^n \xrightarrow{\mathcal{P}} \mathbf{z} \in \mathbb{F}^m$$

Encryption / Signature Verification

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 $\mathcal{P}$ of a multivariate public key cryptosystem, find affine maps $\bar{\mathcal{S}}$ and $\bar{\mathcal{T}}$ as well as quadratic map $\bar{\mathcal{Q}}$ in class $\mathcal{C}$ such that $\mathcal{P} = \bar{\mathcal{T}} \circ \bar{\mathcal{Q}} \circ \bar{\mathcal{S}}$.

$\Rightarrow$ 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 $\mathcal{P}(\mathbf{w}) = \mathbf{z}$ 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

Generic (Direct) Attacks

Try to solve the public equation $\mathcal{P}(\mathbf{w}) = \mathbf{z}$ 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?

security level (bit)	number of equations			
	$\mathbb{F}_2$ *	$\mathbb{F}_{16}$	$\mathbb{F}_{31}$	$\mathbb{F}_{256}$
80	88	30	28	26
100	110	39	36	33
128	140	51	48	43
192	208	80	75	68
256	280	110	103	93

* 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

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

or

2. 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":

$$\text{Complexity} = \min_k 3q^k \cdot \binom{n-k+d_{\mathrm{XL}}}{d_{\mathrm{XL}}}^2 \cdot \binom{n-k}{d}.$$

[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 ≥ 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 ≥ 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 Faugére et al. (F_4, F_5)
- With linear algebra constant $2 < \omega \leq 3$.

$$\mathrm{Complexity}(q, m, n) = O\left(\binom{n + d_{\mathrm{reg}} - 1}{d_{\mathrm{reg}}}^{\omega}\right) \qquad \text{(for larger } q\text{)}$$

- Can also be "Hybridized":

$$\mathrm{Complexity}(q, m, n) = \min_k \; q^k \cdot O\left(\binom{n - k + d_{\mathrm{reg}} - 1}{d_{\mathrm{reg}}}^{\omega}\right)$$

- 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

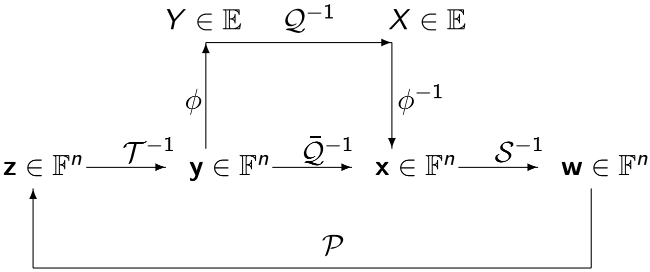
- resistant against attacks with quantum computers
- reasonably fast
- only simple arithmetic operations required
  $\Rightarrow$ can be implemented on low cost devices
  $\Rightarrow$ suitable for security solutions for the IoT
- many practical signature schemes (UOV, Rainbow, HFEv-, ...)
- short signatures (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 $\phi : \mathbb{F}_q^n \to \mathbb{F}_{q^n}$, $(a_1, \ldots, a_n) \mapsto \sum_{i=1}^n a_i \cdot X^{i-1}$
- 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^*

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

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

Key Generation

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

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

- *private key*: $\mathcal{S}, \mathcal{T}$

Linearization Attack against C^*

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

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 $\mathbf{w}$ and $\mathbf{z}$

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

1. Compute $N \geq \frac{(n+1)\cdot(n+2)}{2}$ pairs $(\mathbf{z}^{(k)}/\mathbf{w}^{(k)})$ and substitute into $(\star)$.
2. Solve the resulting linear system for the coefficients α_{ij}, β_i, γ_j and δ.
   $\Rightarrow$ n bilinear equations in $w_1, \ldots, w_n, z_1, \ldots, z_n$
3. Substitute $\mathbf{z}^\star$ into these bilinear equations and solve for $\mathbf{w}^\star$.

pFLASH: Prefixed C^{*-}signature scheme

Natural restriction of $\mathcal{Q}$ to hyperplane = set coordinate to 0

Start from a C^* scheme with $\mathcal{Q}(x) = x^{1+q^\theta}$ 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. $\mathcal{P} = T^- \circ \mathcal{Q} \circ S^-$ 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 $\mathcal{P}^{-1}(m)$ for $m \in \mathbb{F}_q^{n-r}$, the legitimate user first pads m randomly into vector $m' \in (\mathbb{F})^n$ and compute $T^{-1} \circ \mathcal{Q}^{-1} \circ S^{-1}(m')$. 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 $\mathcal{Q}$.

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 $\phi : \mathbb{F}^n \to \mathbb{E}$

Original HFE

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

$$\mathcal{Q}(X) = \sum_{0 \le i \le j}^{q^i + q^j \le D} \alpha_{ij} X^{q^i + q^j} + \sum_{i=0}^{q^i \le D} \beta_i \cdot X^{q^i} + \gamma$$

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

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

MinRank Attack against HFE

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

- the linear maps $\mathcal{S}$ and $\mathcal{T}$ relate to univariate maps $\mathcal{S}^\star(X) = \sum_{i=1}^{n-1} s_i \cdot X^{q^i}$ amd $\mathcal{T}^\star(X) = \sum_{i=1}^{n-1} t_i \cdot X^{q^i}$, with $s_i,\ t_i \in \mathbb{E}$.
- the public key $\mathcal{P}^\star$ can be expressed as $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p^\star_{ij} X^{q^i+q^j} = \underline{X} \cdot P^\star \cdot \underline{X}^T$,
- Components of $P^\star$ 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 $\mathcal{P}^\star \Rightarrow$ 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 $r = \lfloor \log_q(D-1) \rfloor + 1$.

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}

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

with $r = \lfloor \log_q(D-1) \rfloor + 1$.

$\Rightarrow$ 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 $\phi : \mathbb{F}^n \to \mathbb{E}$
- central map $\mathcal{Q} : \mathbb{F}^v \times \mathbb{E} \to \mathbb{E}$, where the β_i and γ are affine.

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

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

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

Signing Message digest $\mathbf{z}$

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

Security vs. Efficiency

Main Attacks

- MinRank Attack $\mathrm{Rank}(F) = r + a + v$

$$\Rightarrow \mathrm{Compl}_{\mathrm{MinRank}} = \binom{n+r+a+v}{r+a+v}^{\omega}$$

- Direct attack [DY13]

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

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.

$$\mathrm{Complexity}_{\mathrm{Berlekamp}} = \mathcal{O}(D^3 + n \cdot D^2)$$

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

security level	80	100	128	192	256
# equations	100	200	256	384	512
pubkey size (kB)	>250	> 500	> 1000	> 3000	> 8000

QUARTZ

- standardized by Courtois, Patarin in 2002
- HFEv$^-$ with $\mathbb{F} = \mathrm{GF}(2)$, $n = 103$, $D = 129$, $a = 3$ and $v = 4$
- public key: quadratic map $\mathcal{P} = \mathcal{T} \circ \mathcal{Q} \circ \mathcal{S} : \mathrm{GF}(2)^{107} \to \mathrm{GF}(2)^{100}$
- Prevent birthday attacks $\Rightarrow$ Generate four HFEv$^-$ signatures
  (for $\mathbf{w}$, $\mathcal{H}(\mathbf{w}|00)$, $\mathcal{H}(\mathbf{w}|01)$ and $\mathcal{H}(\mathbf{w}|11)$)
- 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 $(\mathcal{S}, \mathcal{Q}, \mathcal{T})$ message $\mathbf{d}$, repetition factor k
Output: signature $\sigma \in \mathbb{F}_2^{(n-a)+k(a+v)}$

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

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**

```
h ← Hash(d)
(S_k, X_k, ..., X_1) ← σ
for i = 1 to k do
    D_i ← first n − a bits of h
    h ← Hash(h)
end for
for i = k − 1 to 0 do
    S_i ← P(S_{i+1} || X_{i+1}) ⊕ D_{i+1}
end for
if S_0 = 0 then
    return TRUE
else
    return FALSE
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_{reg}.

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

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

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

Aggressive $\Rightarrow$ Choose D as small as possible (GUI, Patented)

- $D = 9 \Rightarrow r = \lfloor \log_2(D-1) \rfloor + 1 = 4$
- $D = 17 \Rightarrow r = \lfloor \log_2(D-1) \rfloor + 1 = 5$
- $D = 33 \Rightarrow r = \lfloor \log_2(D-1) \rfloor + 1 = 6$

Increase a and v ($0 \le v - a \le 1$) 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/GeMSS's gains.

$\Rightarrow$ very large public key size

security level	80	100	128	192	256
min # equations	117	155	208	332	457

Proposed Parameters (Signature includes 128-bit salt)

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

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/GeMSS's gains.

$\Rightarrow$ very large public key size

security level	80	100	128	192	256
min # equations	117	155	208	332	457

Proposed Parameters (Signature includes 128-bit salt)

NIST Category level (bit)	Parameters $\mathbb{F}_q, n, D, \Delta, v, nb\_ite$	public key size (kB)	private key size (kB)	signature size (bit)
I	GeMSS ($\mathbb{F}_2$,174,513,12,12,4)	417	14.5	384
III	GeMSS ($\mathbb{F}_2$,265,513,22,20,4)	1,304	40.3	704
V	GeMSS ($\mathbb{F}_2$,354,513,30,33,4)	3,604	83.7	832

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

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

$x_1, \ldots, x_v$: 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 $\tilde{p}(\mathbf{x}) = \mathbf{x}^T \cdot M \cdot \mathbf{x}$ with

$$M = \left(\begin{array}{c|c} *_{v \times v} & *_{o \times v} \\ \hline *_{v \times o} & 0_{o \times o} \end{array} \right)$$

where $*$ denotes arbitrary entries subject to symmetry.

Kipnis-Shamir OV attack when $o = v$

$$\mathcal{O} := \{\mathbf{x} \in \mathbb{F}^n : x_1 = \ldots = x_v = 0\} \quad \text{“Oilspace”}$$
$$\mathcal{V} := \{\mathbf{x} \in \mathbb{F}^n : x_{v+1} = \ldots = x_n = 0\} \quad \text{“Vinegarspace”}$$

Let E, F be invertible “OV-matrices”, i.e. $E, F = \begin{pmatrix} \star & \star \\ \star & 0 \end{pmatrix}$ 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 $v \leq o$, breaks the balanced OV scheme in polynomial time.
- For $v > o$ the complexity of the attack is about $q^{v-o} \cdot o^4$.

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

Other Attacks

- **Collision Attack**: $o \geq \frac{2^{2\ell}}{\log_2(q)}$ for ℓ-bit security.
- **Direct Attack**: Try to solve the public equation $\mathcal{P}(\mathbf{w}) = \mathbf{z}$ 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 ℓ-bit security.
- **Direct Attack**: Try to solve the public equation $\mathcal{P}(\mathbf{w}) = \mathbf{z}$ 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 $\mathcal{S}$ ("good keys") which transforms the public matrices H_i into the form of UOV matrices

$$(S^T)^{-1} \cdot H_i \cdot S^{-1} = \begin{pmatrix} \star & \star \\ \star & 0 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 & \star \\ 0 & 1 \end{pmatrix}$$

$\Rightarrow$ 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($\mathbb{F}, o, v$)

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

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 $\mathcal{Q}$ consists of $m = n - v_1$ polynomials $f^{v_1+1}, \ldots, f^{(n)}$ of the form

$$f^{(k)} = \sum_{i,j \in V_\ell} \alpha_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell, j \in O_\ell} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell \cup O_\ell} \gamma_i^{(k)} x_i + \delta^{(k)},$$

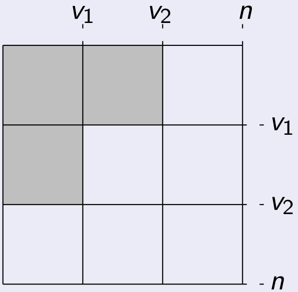
  with coefficients $\alpha_{ij}^{(k)}$, $\beta_{ij}^{(k)}$, $\gamma_i^{(k)}$ and $\delta^{(k)}$ randomly chosen from $\mathbb{F}$ and ℓ being the only integer such that $k \in O_\ell$.
- Choose randomly two affine (or linear) transformations $\mathcal{T} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^n$.
- *public key*: $\mathcal{P} = \mathcal{T} \circ \mathcal{Q} \circ \mathcal{S} : \mathbb{F}^n \to \mathbb{F}^m$
- *private key*: $\mathcal{T}$, $\mathcal{Q}$, $\mathcal{S}$

Idea of Rainbow

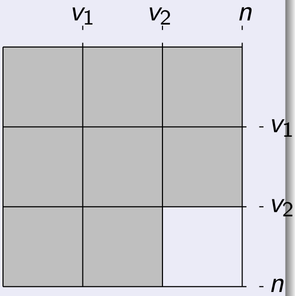
Inversion of the central map

- 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

$F^{(k)} =$ 

$v_1 + 1 \leq k \leq v_2$

$F^{(k)} =$ 

$v_2 + 1 \leq k \leq n$

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 $\mathcal{Q} = (f^{(v_1+1)}, \ldots, f^{(n)})$, vector $\mathbf{y} \in \mathbb{F}^m$.
Output: vector $\mathbf{x} \in \mathbb{F}^n$ with $\mathcal{Q}(\mathbf{x}) = \mathbf{y}$.

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

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

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

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

Accept the signature $\mathbf{z} \Leftrightarrow \mathbf{w}' = \mathbf{w}$.

Security

Rainbow is an extension of UOV
$\Rightarrow$ 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 $\mathbf{v}$, let $\sum_i \alpha_i H_i \mathbf{v} = 0$, 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 $i \le v$, $w_i = w'_i$ for $i > v$. Evaluate $\mathbf{z}$ in $\mathbf{w}'$.
2. Find m equations by letting all $(w'_n)^2$ terms vanish; there are v of λ_i's.
3. Set all cross-terms involving w'_n in $z_1 - \sigma_1^{(1)} z_{v+1} - \sigma_2^{(1)} z_{v+2} - \cdots - \sigma_o^{(1)} z_m$ 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 $z_2 - \sigma_1^{(2)} z_{v+1} - \sigma_2^{(2)} z_{v+2} - \cdots - \sigma_o^{(2)} z_m$ 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

NIST Security Category	parameters $\mathbb{F}, v_1, o_1, o_2$	public key size (kB)	private key size (kB)	hash size (bit)	signature (bit)
I	$\mathbb{F}_{16}$,32,32,32	148.5	97.9	256	512
III	$\mathbb{F}_{256}$,68,36,36	703.9	525.2	576	1,248
V	$\mathbb{F}_{256}$,92,48,48	1,683.3	1,244.4	768	1,632

Thank you for Listening

That's it Folks!

Classic Rainbow Performance Data

Processor: Intel(R) Xeon(R) CPU E3-1275 v5 @ 3.60GHz (Skylake)
Operating System: Linux 4.8.5, GCC compiler version 6.4, Use AVX2

parameter set		key gen.	sign. gen.	sign. verif.
Ia	cycles	9.01M	463K	145K
	time (ms)	2.50	0.129	0.0402
	memory	3.5MB	3.0MB	2.8MB
IIIc	cycles	103M	623K	635K
	time (ms)	28.6	0.173	0.176
	memory	4.6MB	3.5MB	3.3MB
Vc	cycles	92.0M	873K	283K
	time (ms)	25.5	0.243	0.0786
	memory	7.0MB	4.2MB	4.5MB

Table: Performance of standard Rainbow on Linux/Skylake (AVX2)

Compressed Rainbow Performance Data

Processor: Intel(R) Xeon(R) CPU E3-1275 v5 @ 3.60GHz (Skylake)
Operating System: Linux 4.8.5, GCC compiler version 6.4, Use AVX2

parameter set		key gen.	sign. gen.*	sign. verif.
Ia	cycles	9.57M	6.32M	3.56M
	time (ms)	2.66	1.75	0.99
	memory	3.5MB	3.0MB	2.8MB
IIIc	cycles	117M	69.2M	20.1M
	time (ms)	32.5	19.2	5.58
	memory	4.6MB	3.5MB	3.3MB
Vc	cycles	97.5M	72.4M	47.1M
	time (ms)	27.1	20.1	13.1
	memory	7.0MB	4.2MB	4.5MB

Table: Performance of cyclic/compressed Rainbow on Linux/Skylake (AVX2)

* decompressing from 512-bit secret key