Rainbow

Jintai Ding, Ming-Shing Chen, Matthias Kannwischer, Jacques Patarin, Albrecht Petzoldt, Dieter Schmidt, Bo Yin Yang

3rd NIST Post-Quantum Standardization Conference

07.06.2021

Multivariate Cryptography

Public Key: System of multivariate quadratic polynomials

$$
\begin{aligned}
p^{(1)}(x_1, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(1)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(1)} \cdot x_i + p_0^{(1)} \\
p^{(2)}(x_1, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(2)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(2)} \cdot x_i + p_0^{(2)} \\
& \vdots \\
p^{(m)}(x_1, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(m)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(m)} \cdot x_i + p_0^{(m)}
\end{aligned}
$$

Multivariate Cryptography

Public Key: System of multivariate quadratic polynomials

$$p^{(1)}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(1)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(1)} \cdot x_i + p_0^{(1)}$$

$$p^{(2)}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(2)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(2)} \cdot x_i + p_0^{(2)}$$

$$\vdots$$

$$p^{(m)}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(m)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(m)} \cdot x_i + p_0^{(m)}$$

Security based on the

Problem MQ: Given m multivariate quadratic polynomials $p^{(1)}(\mathbf{x}), \ldots, p^{(m)}(\mathbf{x})$, find a vector $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_n)$ such that $p^{(1)}(\bar{\mathbf{x}}) = \ldots = p^{(m)}(\bar{\mathbf{x}}) = 0$.

Construction

- Easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$

Construction

- Easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$
- Two invertible linear maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$

Construction

- Easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$
- Two invertible linear maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$
- *Public key*: $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$ supposed to look like a random system

Construction

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

Multivariate Signature Schemes

Signature Generation

$$\begin{tikzcd} d \in \{0,1\}^\star \arrow[r, "\mathcal{H}"] & \mathbf{w} \in \mathbb{F}^n \arrow[r, "\mathcal{S}^{-1}"] & \mathbf{x} \in \mathbb{F}^n \arrow[r, "\mathcal{F}^{-1}"] & \mathbf{y} \in \mathbb{F}^m \arrow[r, "\mathcal{T}^{-1}"] & \mathbf{z} \in \mathbb{F}^m \arrow[lll, bend left, "\mathcal{P}"] \end{tikzcd}$$

Signature Verification

Multivariate Signature Schemes

Signature Generation

$$\begin{tikzcd}
d \in \{0,1\}^\star \arrow[r, "\mathcal{H}"] & \mathbf{w} \in \mathbb{F}^n \arrow[r, "\mathcal{S}^{-1}"] & \mathbf{x} \in \mathbb{F}^n \arrow[r, "\mathcal{F}^{-1}"] & \mathbf{y} \in \mathbb{F}^m \arrow[r, "\mathcal{T}^{-1}"] & \mathbf{z} \in \mathbb{F}^m \arrow[lll, bend left, "\mathcal{P}"]
\end{tikzcd}$$

Signature Verification

Signature Generation: Given: document $d \in \{0,1\}^\star$, private key $\mathcal{S}, \mathcal{F}, \mathcal{T}$ compute recursively $\mathbf{w} = \mathcal{H}(d)$, $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w})$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x})$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$

Multivariate Signature Schemes

Signature Generation

$$\begin{tikzcd} d \in \{0,1\}^\star \arrow[r, "\mathcal{H}"] & \mathbf{w} \in \mathbb{F}^n \arrow[r, "\mathcal{S}^{-1}"] & \mathbf{x} \in \mathbb{F}^n \arrow[r, "\mathcal{F}^{-1}"] & \mathbf{y} \in \mathbb{F}^m \arrow[r, "\mathcal{T}^{-1}"] & \mathbf{z} \in \mathbb{F}^m \arrow[llll, bend left, "\mathcal{P}"] \end{tikzcd}$$

Signature Verification

Signature Generation: Given: document $d \in \{0,1\}^\star$, private key $\mathcal{S}, \mathcal{F}, \mathcal{T}$ compute recursively $\mathbf{w} = \mathcal{H}(d)$, $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w})$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x})$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$
Signature Verification: Given: document $d \in \{0,1\}^\star$, signature $\mathbf{z} \in \mathbb{F}^n$, public key $\mathcal{P}$ check if $\mathcal{P}(\mathbf{z}) = \mathcal{H}(d)$

The Rainbow Signature Scheme

- finite field $\mathbb{F}$ with q elements, integers $0 < v_1 < v_2 < \cdots < v_u < v_{u+1} = n$

The Rainbow Signature Scheme

- finite field $\mathbb{F}$ with q elements, integers $0 < v_1 < v_2 < \cdots < v_u < v_{u+1} = n$
- set $V_i = \{1, \ldots, v_i\}$ and $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ $(i = 1, \ldots, u) \Rightarrow |V_i| = v_i$, $|O_i| = v_{i+1} - v_i := o_i$

The Rainbow Signature Scheme

- finite field $\mathbb{F}$ with q elements, integers $0 < v_1 < v_2 < \cdots < v_u < v_{u+1} = n$
- set $V_i = \{1, \ldots, v_i\}$ and $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ $(i = 1, \ldots, u) \Rightarrow |V_i| = v_i$, $|O_i| = v_{i+1} - v_i := o_i$
- central map $\mathcal{F}$ consists of $m := n - v_1$ polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$ of the form

$$f^{(k)}(x_1, \ldots, x_n) = \sum_{i,j \in V_\ell} \alpha_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell, j \in O_\ell} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell \cup O_\ell} \gamma_i^{(k)} x_i + \delta^{(k)},$$

where ℓ is the only integer such that $k \in O_\ell$.

The Rainbow Signature Scheme

- finite field $\mathbb{F}$ with q elements, integers $0 < v_1 < v_2 < \cdots < v_u < v_{u+1} = n$
- set $V_i = \{1, \ldots, v_i\}$ and $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ $(i = 1, \ldots, u) \Rightarrow |V_i| = v_i$, $|O_i| = v_{i+1} - v_i := o_i$
- central map $\mathcal{F}$ consists of $m := n - v_1$ polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$ of the form

$$f^{(k)}(x_1, \ldots, x_n) = \sum_{i,j \in V_\ell} \alpha_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell, j \in O_\ell} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell \cup O_\ell} \gamma_i^{(k)} x_i + \delta^{(k)},$$

  where ℓ is the only integer such that $k \in O_\ell$.
- two invertible linear maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$

The Rainbow Signature Scheme

- finite field $\mathbb{F}$ with q elements, integers $0 < v_1 < v_2 < \cdots < v_u < v_{u+1} = n$
- set $V_i = \{1, \ldots, v_i\}$ and $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ $(i = 1, \ldots, u) \Rightarrow |V_i| = v_i$, $|O_i| = v_{i+1} - v_i := o_i$
- central map $\mathcal{F}$ consists of $m := n - v_1$ polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$ of the form

$$f^{(k)}(x_1, \ldots, x_n) = \sum_{i,j \in V_\ell} \alpha_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell, j \in O_\ell} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell \cup O_\ell} \gamma_i^{(k)} x_i + \delta^{(k)},$$

  where ℓ is the only integer such that $k \in O_\ell$.
- two invertible linear maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$
- **Public Key**: $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$

The Rainbow Signature Scheme

- finite field $\mathbb{F}$ with q elements, integers $0 < v_1 < v_2 < \cdots < v_u < v_{u+1} = n$
- set $V_i = \{1, \ldots, v_i\}$ and $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ $(i = 1, \ldots, u) \Rightarrow |V_i| = v_i$, $|O_i| = v_{i+1} - v_i := o_i$
- central map $\mathcal{F}$ consists of $m := n - v_1$ polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$ of the form

$$f^{(k)}(x_1, \ldots, x_n) = \sum_{i,j \in V_\ell} \alpha_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell, j \in O_\ell} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in V_\ell \cup O_\ell} \gamma_i^{(k)} x_i + \delta^{(k)},$$

  where ℓ is the only integer such that $k \in O_\ell$.
- two invertible linear maps $\mathcal{S} : \mathbb{F}^m \rightarrow \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \rightarrow \mathbb{F}^n$
- **Public Key**: $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \rightarrow \mathbb{F}^m$
- **Private Key**: $\mathcal{S}$, $\mathcal{F}$, $\mathcal{T}$

Signature Generation

Given a document $d \in \{0,1\}^\star$ to be signed, perform the following steps

Signature Generation

Given a document $d \in \{0,1\}^\star$ to be signed, perform the following steps

1. Use a hash function $\mathcal{H} : \{0,1\}^\star \rightarrow \mathbb{F}^m$ to compute $\mathbf{w} = \mathcal{H}(d)$.

Signature Generation

Given a document $d \in \{0,1\}^\star$ to be signed, perform the following steps

1. Use a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute $\mathbf{w} = \mathcal{H}(d)$.
2. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$.

Signature Generation

Given a document $d \in \{0,1\}^\star$ to be signed, perform the following steps

1. Use a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute $\mathbf{w} = \mathcal{H}(d)$.
2. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$.
3. Choose the Vinegar variables $y_1, \ldots, y_{v_1}$ at random and substitute them into the polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$.

Signature Generation

Given a document $d \in \{0,1\}^\star$ to be signed, perform the following steps

1. Use a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute $\mathbf{w} = \mathcal{H}(d)$.
2. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$.
3. Choose the Vinegar variables $y_1, \ldots, y_{v_1}$ at random and substitute them into the polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$.
4. For $i = 1, \ldots, u$ do
   - Solve the linear system provided by $f^{(v_i+1)}, \ldots, f^{(v_{i+1})}$ to get the values of $y_{v_i+1}, \ldots, y_{v_{i+1}}$. and substitute them into the polynomials $f^{(v_{i+1}+1)}, \ldots, f^{(n)}$.

Signature Generation

Given a document $d \in \{0,1\}^\star$ to be signed, perform the following steps

1. Use a hash function $\mathcal{H} : \{0,1\}^\star \rightarrow \mathbb{F}^m$ to compute $\mathbf{w} = \mathcal{H}(d)$.
2. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$.
3. Choose the Vinegar variables $y_1, \ldots, y_{v_1}$ at random and substitute them into the polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$.
4. For $i = 1, \ldots, u$ do
   - Solve the linear system provided by $f^{(v_i+1)}, \ldots, f^{(v_{i+1})}$ to get the values of $y_{v_i+1}, \ldots, y_{v_{i+1}}$. and substitute them into the polynomials $f^{(v_{i+1}+1)}, \ldots, f^{(n)}$.
   - (if $i < u$) Substitute the values of $y_{v_i+1}, \ldots, y_{v_{i+1}}$ into the polynomials $f^{(v_{i+1}+1)}, \ldots, f^{(n)}$.

Signature Generation

Given a document $d \in \{0,1\}^\star$ to be signed, perform the following steps

1. Use a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute $\mathbf{w} = \mathcal{H}(d)$.
2. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$.
3. Choose the Vinegar variables $y_1, \ldots, y_{v_1}$ at random and substitute them into the polynomials $f^{(v_1+1)}, \ldots, f^{(n)}$.
4. For $i = 1, \ldots, u$ do
   - Solve the linear system provided by $f^{(v_i+1)}, \ldots, f^{(v_{i+1})}$ to get the values of $y_{v_i+1}, \ldots, y_{v_{i+1}}$. and substitute them into the polynomials $f^{(v_{i+1}+1)}, \ldots, f^{(n)}$.
   - (if $i < u$) Substitute the values of $y_{v_i+1}, \ldots, y_{v_{i+1}}$ into the polynomials $f^{(v_{i+1}+1)}, \ldots, f^{(n)}$.
5. Set $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}^n$.

Signature Generation

Given a document $d \in \{0,1\}^\star$ to be signed, perform the following steps

1. Use a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute $\mathbf{w} = \mathcal{H}(d)$.
2. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$.
3. Choose the Vinegar variables $y_1, \dots, y_{v_1}$ at random and substitute them into the polynomials $f^{(v_1+1)}, \dots, f^{(n)}$.
4. For $i = 1, \dots, u$ do
   - Solve the linear system provided by $f^{(v_i+1)}, \dots, f^{(v_{i+1})}$ to get the values of $y_{v_i+1}, \dots, y_{v_{i+1}}$. and substitute them into the polynomials $f^{(v_{i+1}+1)}, \dots, f^{(n)}$.
   - (if $i < u$) Substitute the values of $y_{v_i+1}, \dots, y_{v_{i+1}}$ into the polynomials $f^{(v_{i+1}+1)}, \dots, f^{(n)}$.
5. Set $\mathbf{y} = (y_1, \dots, y_n) \in \mathbb{F}^n$.
6. Compute the signature $\mathbf{z} \in \mathbb{F}^n$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.

Signature Verification

Given a document $d \in \{0,1\}^\star$ and a signature $\mathbf{z} \in \mathbb{F}^n$, compute

Signature Verification

Given a document $d \in \{0,1\}^\star$ and a signature $\mathbf{z} \in \mathbb{F}^n$, compute

- $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$ and

Signature Verification

Given a document $d \in \{0,1\}^\star$ and a signature $\mathbf{z} \in \mathbb{F}^n$, compute

- $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$ and
- $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$.

Signature Verification

Given a document $d \in \{0,1\}^\star$ and a signature $\mathbf{z} \in \mathbb{F}^n$, compute

- $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$ and
- $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$.

If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted; otherwise it is rejected.

Conventions

- We restrict to Rainbow schemes with two layers

Conventions

- We restrict to Rainbow schemes with two layers
- We restrict to homogeneous maps $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$

Conventions

- We restrict to Rainbow schemes with two layers
- We restrict to homogeneous maps $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$
- The linear maps $\mathcal{S}$ and $\mathcal{T}$ are represented by matrices S, T of the form

$$S = \begin{pmatrix} I_{o_1} & S'_{o_1 \times o_2} \\ 0_{o_2 \times o_2} & I_{o_2} \end{pmatrix} \quad T = \begin{pmatrix} I_{v_1} & T^{(1)}_{v_1 \times o_1} & T^{(2)}_{v_1 \times o_2} \\ 0_{o_1 \times v_1} & 1_{v_1} & T^{(3)}_{o_1 \times o_2} \\ 0_{o_2 \times v_1} & 0_{o_2 \times o_1} & I_{o_2} \end{pmatrix}.$$

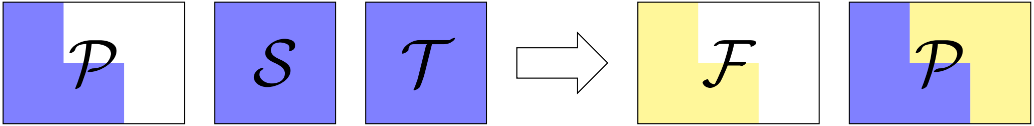
Conventions

- We restrict to Rainbow schemes with two layers
- We restrict to homogeneous maps $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$
- The linear maps $\mathcal{S}$ and $\mathcal{T}$ are represented by matrices S, T of the form

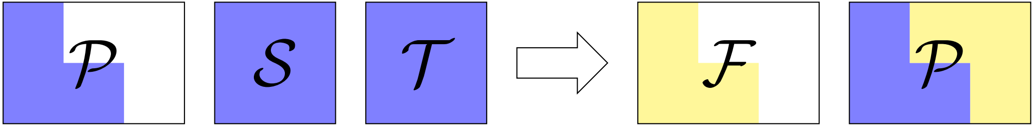
$$S = \begin{pmatrix} I_{o_1} & S'_{o_1 \times o_2} \\ 0_{o_2 \times o_2} & I_{o_2} \end{pmatrix} \quad T = \begin{pmatrix} I_{v_1} & T^{(1)}_{v_1 \times o_1} & T^{(2)}_{v_1 \times o_2} \\ 0_{o_1 \times v_1} & 1_{v_1} & T^{(3)}_{o_1 \times o_2} \\ 0_{o_2 \times v_1} & 0_{o_2 \times o_1} & I_{o_2} \end{pmatrix}.$$

- Our conventions don't weaken the security of the scheme and enable us to speed up the key generation process drastically

CZ-Rainbow

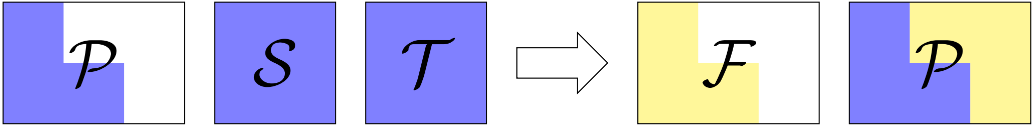


CZ-Rainbow



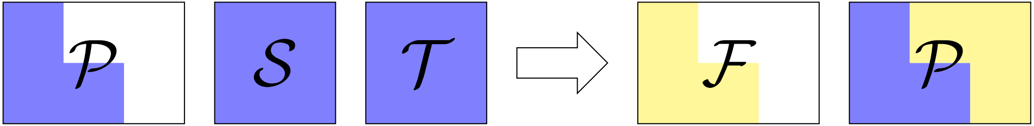
- Generate the blue parts of the public key as well as the linear maps $\mathcal{S}$ and $\mathcal{T}$ using a PRNG

CZ-Rainbow



- Generate the blue parts of the public key as well as the linear maps $\mathcal{S}$ and $\mathcal{T}$ using a PRNG
- Compute the remaining parts of the public key as well as the central map $\mathcal{F}$
  $\Rightarrow$ Drastical Reduction of the public key size (up to 75%)

CZ-Rainbow



- Generate the blue parts of the public key as well as the linear maps $\mathcal{S}$ and $\mathcal{T}$ using a PRNG
- Compute the remaining parts of the public key as well as the central map $\mathcal{F}$
  $\Rightarrow$ Drastical Reduction of the public key size (up to 75%)
- We propose also a compressed version in which the central map is generated from a seed, too.

Key and Signature Sizes

NIST security category	parameters (q, v_1, o_2, o_2)	signature size (bit)
I	(16,36,32,32)	528
III	(256,68,32,48)	1,312
V	(256,96,36,64)	1,696

Key and Signature Sizes

NIST security category	parameters (q, v_1, o_2, o_2)	signature size (bit)
I	(16,36,32,32)	528
III	(256,68,32,48)	1,312
V	(256,96,36,64)	1,696

NIST security category	standard Rainbow		CZ-Rainbow		compressed Rainbow	
	$\|pk\|$	$\|sk\|$	$\|pk\|$	$\|sk\|$	$\|pk\|$	$\|sk\|$
I	157.8	101.2	58.8	101.2	58.1	64B
III	861.4	611.3	258.4	611.3	258.4	64B
V	1,885.4	1,375.7	523.6	1,375,7	523.6	64B

Performance

NIST security category	standard Rainbow			CZ-Rainbow			compressed Rainbow		
	keygen	sign	verify	keygen	sign	verify	keygen	sign	verify
I	9.9M	67k	34k	10.7M	67k	3.5M	10.7M	7.0M	3.5M
III	52M	285k	132k	64M	285k	20M	64M	41M	20M
V	192M	739k	392k	235M	739k	47M	235M	118M	47M

Linux / Skylake (using AVX2 instructions)

See also the talk by M. Kannwischer on Implementing Rainbow on Cortex-M4

Security

- new attacks by Beullens (Intersection and New MinRank)

Security

- new attacks by Beullens (Intersection and New MinRank)
- similar to other candidates (NTRUprime, Crystals) we distinguish between “free” and “real” cost of the attacks

Security

- new attacks by Beullens (Intersection and New MinRank)
- similar to other candidates (NTRUprime, Crystals) we distinguish between “free” and “real” cost of the attacks
- the “real” cost considers the cost of memory access

Security

- new attacks by Beullens (Intersection and New MinRank)
- similar to other candidates (NTRUprime, Crystals) we distinguish between "free" and "real" cost of the attacks
- the "real" cost considers the cost of memory access
- cost of one multiplication
  $\approx (\text{ bit} - \text{length memory transitions}) \times \sqrt{\#\mathit{bits\ non\ sequentially\ accessed}}/2^5$

Security

- new attacks by Beullens (Intersection and New MinRank)
- similar to other candidates (NTRUprime, Crystals) we distinguish between "free" and "real" cost of the attacks
- the "real" cost considers the cost of memory access
- cost of one multiplication
  $\approx (\text{ bit} - \text{length memory transitions}) \times \sqrt{\#\mathit{bits\ non\ sequentially\ accessed}}/2^5$
- For Wiedemann over $(\mathbb{F}_{2^k})^V$ the cost of one multiplication is $\lg V \sqrt{kV}/2^5$ "gates"

Security (2)

security category	parameters (q, v_1, o_1, o_2)	Intersection		New MinRank		target cost
		mults	cost	mults	cost	
I	(16,36,32,32)	$2^{134.3}$	$2^{162.1}$	$2^{122.4}$	$2^{152.3}$	2^{143}
III	(256,68,32,48)	$2^{105.4}$	$2^{248.3}$	$2^{171.8}$	$2^{216.2}$	2^{207}
V	(256,96,36,64)	$2^{254.5}$	$2^{309.9}$	$2^{219.6}$	$2^{276.2}$	2^{272}

Security (2)

security category	parameters (q, v_1, o_1, o_2)	Intersection		New MinRank		target cost
		mults	cost	mults	cost	
I	(16,36,32,32)	$2^{134.3}$	$2^{162.1}$	$2^{122.4}$	$2^{152.3}$	2^{143}
III	(256,68,32,48)	$2^{105.4}$	$2^{248.3}$	$2^{171.8}$	$2^{216.2}$	2^{207}
V	(256,96,36,64)	$2^{254.5}$	$2^{309.9}$	$2^{219.6}$	$2^{276.2}$	2^{272}

$\Rightarrow$ Our parameter proposals meet the NIST requirements

Speeding up the Verification Process

Speeding up the Verification Process

Verification Process for Multivariate Signature Schemes

Speeding up the Verification Process

Verification Process for Multivariate Signature Schemes

Given: message $d \in \{0,1\}^\star$, signature $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{F}^n$, public key $(p^{(1)}, \ldots, p^{(m)})$

Speeding up the Verification Process

Verification Process for Multivariate Signature Schemes

Given: message $d \in \{0,1\}^\star$, signature $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{F}^n$, public key $(p^{(1)}, \ldots, p^{(m)})$

1. Compute the hash value $\mathbf{w} = (w_1, \ldots, w_m) = \mathcal{H}(m)$

Speeding up the Verification Process

Verification Process for Multivariate Signature Schemes

Given: message $d \in \{0,1\}^\star$, signature $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{F}^n$, public key $(p^{(1)}, \ldots, p^{(m)})$

1. Compute the hash value $\mathbf{w} = (w_1, \ldots, w_m) = \mathcal{H}(m)$
2. Check, for $i = 1, \ldots, m$, if
$$p^{(i)}(\mathbf{z}) = w_i$$
holds.

Speeding up the Verification Process

Verification Process for Multivariate Signature Schemes

Given: message $d \in \{0,1\}^\star$, signature $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{F}^n$, public key $(p^{(1)}, \ldots, p^{(m)})$

1. Compute the hash value $\mathbf{w} = (w_1, \ldots, w_m) = \mathcal{H}(m)$
2. Check, for $i = 1, \ldots, m$, if
$$p^{(i)}(\mathbf{z}) = w_i$$
holds.

Accept the signature, if and only if all the tests are fulfilled.

Speeding up the Verification Process

Verification Process for Multivariate Signature Schemes

Given: message $d \in \{0,1\}^\star$, signature $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{F}^n$, public key $(p^{(1)}, \ldots, p^{(m)})$

1. Compute the hash value $\mathbf{w} = (w_1, \ldots, w_m) = \mathcal{H}(m)$
2. Check, for $i = 1, \ldots, m$, if
$$p^{(i)}(\mathbf{z}) = w_i$$
holds.

Accept the signature, if and only if all the tests are fulfilled.

First Observation: We can stop, as soon as we find an i with $p^{(i)}(\mathbf{z}) \neq w_i$.

Checking a Fixed Subset of Equations

- we don't have to check all equations $p^{(i)}(\mathbf{z}) = w_i$ $(i = 1, \ldots, m)$

Checking a Fixed Subset of Equations

- we don't have to check all equations $p^{(i)}(\mathbf{z}) = w_i$ $(i = 1, \ldots, m)$
- we only have to check k equations, where k is the smallest number such that

$$\mathrm{compl}_{direct}(q, k, n) \geq 2^{\lambda}$$

Checking a Fixed Subset of Equations

- we don't have to check all equations $p^{(i)}(\mathbf{z}) = w_i$ $(i = 1, \ldots, m)$
- we only have to check k equations, where k is the smallest number such that

$$\mathrm{compl}_{direct}(q, k, n) \geq 2^{\lambda}$$

Security category	parameters (q, v_1, o_1, o_2)	# equations m	# equations to be checked	compl. of sig. forgery classical	quantum
I	(16,36,32,32)	64	55	145	127
III	(256,68,32,48)	80	72	212	185
V	(256,96,36,64)	100	98	280	247

Checking a Randomly Chosen Subset of Equations

- the signer / adversary does not know which k of the m equations will be checked

Checking a Randomly Chosen Subset of Equations

- the signer / adversary does not know which k of the m equations will be checked
- the number k of equations we have to check is the smallest number such that

$$\min_{k \leq \ell \leq m} \left(\frac{\binom{m}{k}}{\binom{\ell}{k}} \cdot \mathrm{comp}_{direct}(q, \ell, n) \right) \geq 2^{\lambda}$$

Checking a Randomly Chosen Subset of Equations

- the signer / adversary does not know which k of the m equations will be checked
- the number k of equations we have to check is the smallest number such that

$$\min_{k \leq \ell \leq m} \left(\frac{\binom{m}{k}}{\binom{\ell}{k}} \cdot \mathrm{comp}_{direct}(q, \ell, n) \right) \geq 2^{\lambda}$$

- selecting ℓ of the m public equations is costly, but

Checking a Randomly Chosen Subset of Equations

- the signer / adversary does not know which k of the m equations will be checked
- the number k of equations we have to check is the smallest number such that

$$\min_{k \leq \ell \leq m} \left(\frac{\binom{m}{k}}{\binom{\ell}{k}} \cdot \mathrm{comp}_{direct}(q, \ell, n) \right) \geq 2^{\lambda}$$

- selecting ℓ of the m public equations is costly, but
- if the public key is known beforehand (e.g. SecureBoot), this step can be done offline

Checking a Randomly Chosen Subset of Equations

- the signer / adversary does not know which k of the m equations will be checked
- the number k of equations we have to check is the smallest number such that

$$\min_{k \leq \ell \leq m} \left(\frac{\binom{m}{k}}{\binom{\ell}{k}} \cdot \mathrm{comp}_{direct}(q, \ell, n) \right) \geq 2^{\lambda}$$

- selecting ℓ of the m public equations is costly, but
- if the public key is known beforehand (e.g. SecureBoot), this step can be done offline

Security category	parameters (q, v_1, o_1, o_2)	# equations m	# equations to be checked	compl. of sig. forgery classical	compl. of sig. forgery quantum
I	(16,36,32,32)	64	32	143	125
II	(256,68,32,48)	80	48	212	192
III	(256,96,36,64)	100	72	276	245

Checking Randomly Chosen Linear Combinations

- we check k equations of the form $\tilde{p}^{(i)} = \sum_{j=1}^{m} \left(\alpha_j^{(i)} p^{(j)} \right) (\mathbf{z}) = \sum_{j=1}^{m} \alpha_j^{(i)} w_j = \tilde{w}_i$

Checking Randomly Chosen Linear Combinations

- we check k equations of the form $\tilde{p}^{(i)} = \sum_{j=1}^{m} \left(\alpha_j^{(i)} p^{(j)} \right) (\mathbf{z}) = \sum_{j=1}^{m} \alpha_j^{(i)} w_j = \tilde{w}_i$
- we have $\alpha_j^{(i)} \in \{0, 1\}$ and $\sum_{i=1}^{m} \alpha_j^{(i)} = 1$

Checking Randomly Chosen Linear Combinations

- we check k equations of the form $\tilde{p}^{(i)} = \sum_{j=1}^{m} \left(\alpha_j^{(i)} p^{(j)} \right) (\mathbf{z}) = \sum_{j=1}^{m} \alpha_j^{(i)} w_j = \tilde{w}_i$
- we have $\alpha_j^{(i)} \in \{0, 1\}$ and $\sum_{i=1}^{m} \alpha_j^{(i)} = 1$
- the adversary can either solve the whole public system or try to guess $\mathbf{z}$ in such a way that the k equations are fulfilled

Checking Randomly Chosen Linear Combinations

- we check k equations of the form $\tilde{p}^{(i)} = \sum_{j=1}^{m} \left(\alpha_j^{(i)} p^{(j)} \right) (\mathbf{z}) = \sum_{j=1}^{m} \alpha_j^{(i)} w_j = \tilde{w}_i$
- we have $\alpha_j^{(i)} \in \{0, 1\}$ and $\sum_{i=1}^{m} \alpha_j^{(i)} = 1$
- the adversary can either solve the whole public system or try to guess $\mathbf{z}$ in such a way that the k equations are fulfilled
- if the public key is known beforehand, we can generate the system $\tilde{\mathcal{P}}_i$ offline

Checking Randomly Chosen Linear Combinations

- we check k equations of the form $\tilde{p}^{(i)} = \sum_{j=1}^{m} \left(\alpha_j^{(i)} p^{(j)}\right)(\mathbf{z}) = \sum_{j=1}^{m} \alpha_j^{(i)} w_j = \tilde{w}_i$
- we have $\alpha_j^{(i)} \in \{0,1\}$ and $\sum_{i=1}^{m} \alpha_j^{(i)} = 1$
- the adversary can either solve the whole public system or try to guess $\mathbf{z}$ in such a way that the k equations are fulfilled
- if the public key is known beforehand, we can generate the system $\tilde{\mathcal{P}}_i$ offline

Security category	parameters (q, v_1, o_1, o_2)	# equations m	# variables n	# equations to be checked
I	(16,36,32,32)	64	100	32
III	(256,68,32,48)	80	148	24
V	(256,96,36,64)	100	196	32

Rainbow: Advantages and Disadvantages

Advantages

- short signatures
- fast key generation
- very fast signature generation
- very fast signature verification

Rainbow: Advantages and Disadvantages

Advantages

- short signatures
- fast key generation
- very fast signature generation
- very fast signature verification

Disadvantages

- large key sizes
- no security proof

Thank you for your attention

The Team:
Jintai Ding, Ming-Shing Chen, Matthias Kannwischer, Jacques Patarin,
Albrecht Petzoldt, Dieter Schmidt, BoYin Yang

www.pqcrainbow.org