Key recovery from one vector in UOV schemes

Pierre Pébereau
Sorbonne Université, LIP6, CNRS, Thales SIX

January 19, 2024

Post-Quantum Zoo

Scheme	**Assumptions**	**Public key size** (bytes)	**Signature size** (bytes)
EdDSA	Discrete log	32	64
Sphincs+ 128s	Hash-based	32	7856
Falcon 512	Structured lattices	897	666
Dilithium2	Structured lattices	1312	2420

Figure: Pre-quantum and NIST standard signatures

Source: PQShield (https://pqshield.github.io/nist-sigs-zoo/)

Post-Quantum Zoo

Scheme	Assumptions	**Public key size** (bytes)	**Signature size** (bytes)
EdDSA	Discrete log	32	64
Sphincs+ 128s	Hash-based	32	7856
Falcon 512	Structured lattices	897	666
Dilithium2	Structured lattices	1312	2420
uov-Ip	Multivariate	43 576	128

Figure: Pre-quantum and NIST standard signatures

Source: PQShield (https://pqshield.github.io/nist-sigs-zoo/)

Post-Quantum Zoo

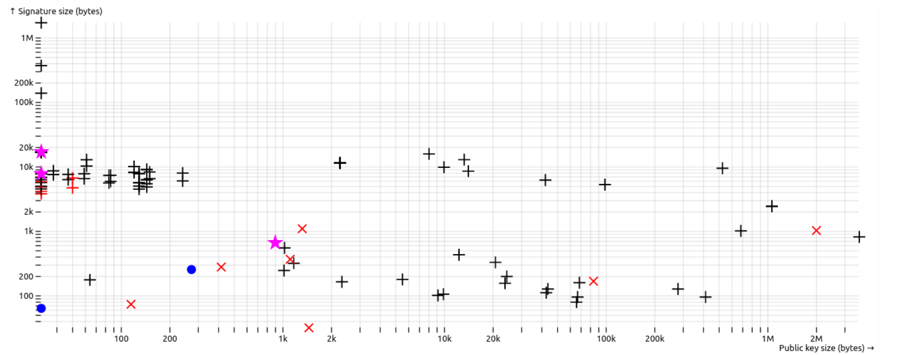


Figure: Signature and key sizes in the NIST competition versus standards (pink stars) and classical cryptography (blue dots) at security level I.

Source: PQShield (https://pqshield.github.io/nist-sigs-zoo/)

Post-Quantum Zoo

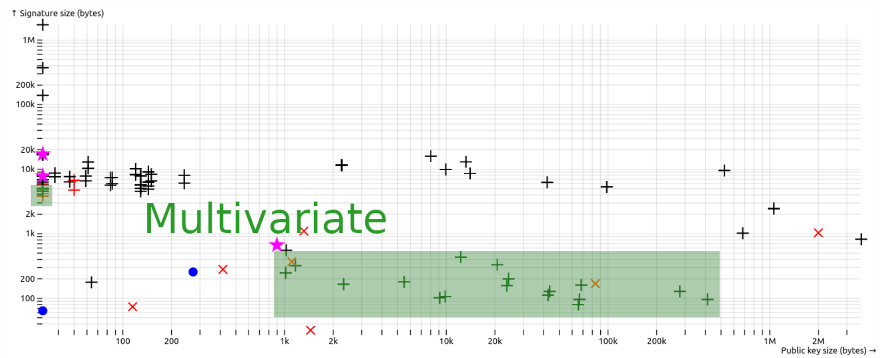
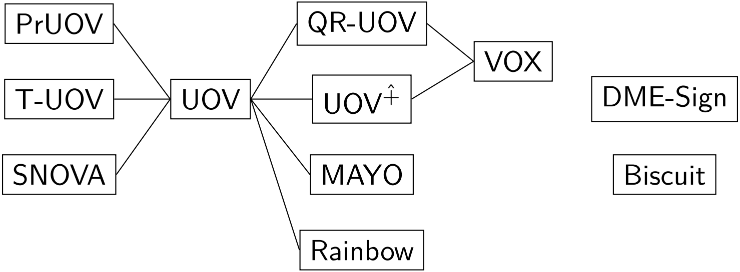


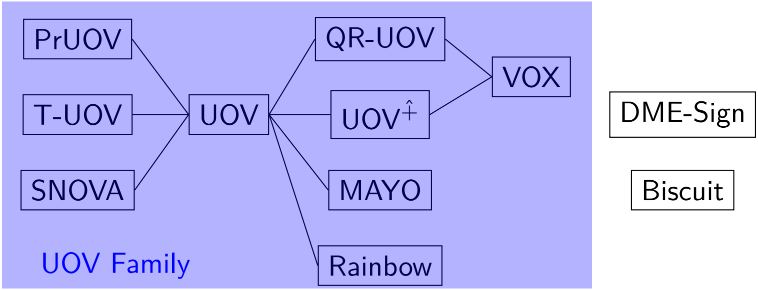
Figure: Signature and key sizes in the NIST competition versus standards (pink stars) and classical cryptography (blue dots) at security level I.

Source: PQShield (https://pqshield.github.io/nist-sigs-zoo/)

Multivariate Post-Quantum Zoo



Multivariate Post-Quantum Zoo



UOV Signature Scheme

Unbalanced Oil and Vinegar, informally [Kipnis, Patarin, Goubin, 1999]

- The legitimate signer solves a linear system to **sign**. EASY

UOV Signature Scheme

Unbalanced Oil and Vinegar, informally [Kipnis, Patarin, Goubin, 1999]

- The legitimate signer solves a linear system to **sign**. EASY
- An adversary solves a quadratic system to **forge** a signature. HARD

UOV Signature Scheme

Unbalanced Oil and Vinegar, informally [Kipnis, Patarin, Goubin, 1999]

- The legitimate signer solves a linear system to **sign**. EASY
- An adversary solves a quadratic system to **forge** a signature. HARD
- The receiver evaluates a quadratic map to verify a signature. EASY

Polynomial system solving crash course

Polynomial system

A collection of *m polynomials* in n variables: $P_1, \ldots, P_m \in \mathbb{F}_q[x_1, \ldots, x_n]$

Polynomial system solving crash course

Polynomial system

A collection of *m polynomials* in *n* variables: $P_1, \ldots, P_m \in \mathbb{F}_q[x_1, \ldots, x_n]$

Ideal

This system defines an ideal of the *polynomial ring* $\mathcal{R} = \mathbb{F}_q[x_1, \ldots, x_n]$:

$$I = \langle P_1, \ldots, P_m \rangle := \left\{ \sum_{i=1}^{m} a_i P_i, \quad (a_i) \in \mathcal{R}^m \right\}$$

Polynomial system solving crash course

Polynomial system

A collection of *m polynomials* in *n* variables: $P_1, \ldots, P_m \in \mathbb{F}_q[x_1, \ldots, x_n]$

Ideal

This system defines an ideal of the *polynomial ring* $\mathcal{R} = \mathbb{F}_q[x_1, \ldots, x_n]$:

$$I = \langle P_1, \ldots, P_m \rangle := \left\{ \sum_{i=1}^{m} a_i P_i, \quad (a_i) \in \mathcal{R}^m \right\}$$

Variety

The set of solutions of the system is called an *algebraic variety*

$$V(I) = \{x \in \overline{\mathbb{F}_q}^n, \forall p \in I, p(x) = 0\}$$

Polynomial system solving crash course

Polynomial system

A collection of m *polynomials* in n variables: $P_1, \ldots, P_m \in \mathbb{F}_q[x_1, \ldots, x_n]$

Ideal

This system defines an ideal of the *polynomial ring* $\mathcal{R} = \mathbb{F}_q[x_1, \ldots, x_n]$:

$$I = \langle P_1, \ldots, P_m \rangle := \left\{ \sum_{i=1}^{m} a_i P_i, \quad (a_i) \in \mathcal{R}^m \right\}$$

Variety

The set of solutions of the system is called an *algebraic variety*

$$V(I) = \{x \in \overline{\mathbb{F}_q}^n, \forall p \in I, p(x) = 0\}$$

If the system is *regular*, then $V(I)$ has *dimension* $n - m$.

Polynomial system solving crash course

Multivariate Quadratic Problem

Find **a** solution $x \in \mathbb{F}_q^n$ to a system of m quadratic equations in n variables

$$\mathcal{P}(x) = 0 \in \mathbb{F}_q^m$$

This problem is **NP-hard** (Equivalent to SAT in $\mathbb{F}_2$).

Polynomial system solving crash course

Multivariate Quadratic Problem

Find **a** solution $x \in \mathbb{F}_q^n$ to a system of m quadratic equations in n variables

$$\mathcal{P}(x) = 0 \in \mathbb{F}_q^m$$

This problem is **NP-hard** (Equivalent to SAT in $\mathbb{F}_2$).

Complexity

Under **regularity** assumptions and for **zero-dimensional systems**, solved by performing linear algebra on Macaulay matrix in degree d_{reg}, the first non-positive index in the Hilbert series:

$$H_{\mathcal{R}/I}(t) = \frac{(1-t^2)^m}{(1-t)^n} \to O\left(\binom{n+d_{reg}}{d_{reg}}^{\omega}\right)$$

UOV: Original formulation

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, 1999]

Private Key: - **structured** triangular matrices $F = (F_1, \ldots, F_m) \in (\mathbb{F}_q^{n \times n})^m$
- $A \in GL_n(\mathbb{F}_q)$ random change of variables

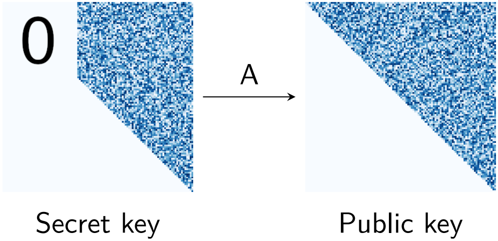


Figure: uov($m = 44, n = 112$) Key Pair in $\mathbb{F}_{257}$

UOV: Original formulation

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, 1999]

Private Key: - **structured** triangular matrices $F = (F_1, \ldots, F_m) \in (\mathbb{F}_q^{n \times n})^m$
- $A \in GL_n(\mathbb{F}_q)$ random change of variables

Public Key: triangular matrices $G = F \circ A = (A^T F_1 A, \ldots, A^T F_m A)$

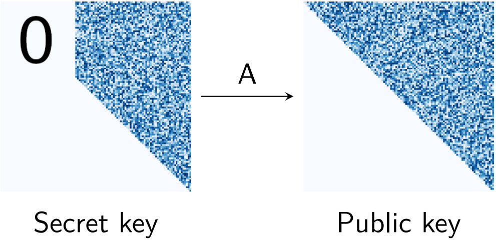


Figure: uov($m = 44, n = 112$) Key Pair in $\mathbb{F}_{257}$

UOV: Original formulation

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, 1999]

Private Key: - **structured** triangular matrices $F = (F_1, \ldots, F_m) \in (\mathbb{F}_q^{n\times n})^m$
- $A \in GL_n(\mathbb{F}_q)$ random change of variables

Public Key: triangular matrices $G = F \circ A = (A^T F_1 A, \ldots, A^T F_m A)$

Link with standard multivariate cryptography

Private key polynomials: m quadratic forms $\boldsymbol{x}^T F_i \boldsymbol{x}$ **linear** in $x_1, \ldots, x_m$
Public key polynomials: m quadratic forms $\boldsymbol{x}^T G_i \boldsymbol{x}$ in n variables.

UOV: Original formulation

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, 1999]

Private Key: - **structured** triangular matrices $F = (F_1, \dots, F_m) \in (\mathbb{F}_q^{n\times n})^m$
- $A \in GL_n(\mathbb{F}_q)$ random change of variables

Public Key: triangular matrices $G = F \circ A = (A^T F_1 A, \dots, A^T F_m A)$

Link with standard multivariate cryptography

Private key polynomials: m quadratic forms $\boldsymbol{x}^T F_i \boldsymbol{x}$ **linear** in $x_1, \dots, x_m$

Public key polynomials: m quadratic forms $\boldsymbol{x}^T G_i \boldsymbol{x}$ in n variables.

$x_1, \dots, x_m \rightarrow$ **oil variables**

$x_{m+1}, \dots, x_n \rightarrow$ **vinegar variables**

UOV: Original formulation

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, 1999]

Private Key: - **structured** triangular matrices $F = (F_1, \dots, F_m) \in (\mathbb{F}_q^{n\times n})^m$
- $A \in GL_n(\mathbb{F}_q)$ random change of variables

Public Key: triangular matrices $G = F \circ A = (A^T F_1 A, \dots, A^T F_m A)$

Link with standard multivariate cryptography

Private key polynomials: m quadratic forms $\mathbf{x}^T F_i \mathbf{x}$ **linear** in $x_1, \dots, x_m$

Public key polynomials: m quadratic forms $\mathbf{x}^T G_i \mathbf{x}$ in n variables.

$x_1, \dots, x_m \rightarrow$ **oil variables**

$x_{m+1}, \dots, x_n \rightarrow$ **vinegar variables**

In practice: $\underbrace{2m \leq}_{\text{[KS98]}} n$

UOV: Original formulation

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, 1999]

Private Key: - **structured** triangular matrices $F = (F_1, \ldots, F_m) \in (\mathbb{F}_q^{n\times n})^m$
- $A \in GL_n(\mathbb{F}_q)$ random change of variables

Public Key: triangular matrices $G = F \circ A = (A^T F_1 A, \ldots, A^T F_m A)$

Link with standard multivariate cryptography

Private key polynomials: m quadratic forms $\mathbf{x}^T F_i \mathbf{x}$ **linear** in $x_1, \ldots, x_m$

Public key polynomials: m quadratic forms $\mathbf{x}^T G_i \mathbf{x}$ in n variables.

$x_1, \ldots, x_m \rightarrow$ **oil variables**

$x_{m+1}, \ldots, x_n \rightarrow$ **vinegar variables**

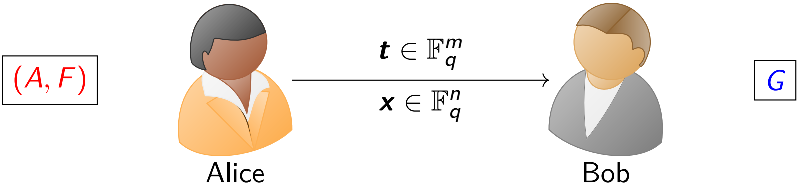
In practice: $\underbrace{2m \leq}_{\text{[KS98]}} n \underbrace{\leq 3m}_{\text{Key sizes}}$

UOV: Signing process

Signing

A **signature** for the message $\boldsymbol{t} \in \mathbb{F}_q^m$ is a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ such that

$$1 \leq i \leq m, G_i(\boldsymbol{x}) = t_i$$



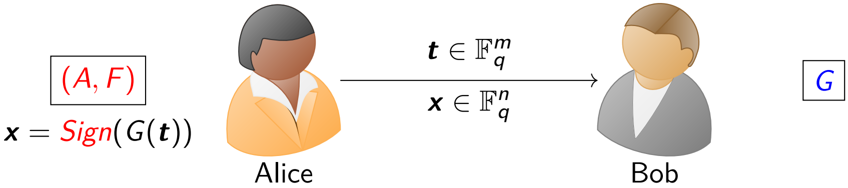
UOV: Signing process

Signing

A **signature** for the message $\boldsymbol{t} \in \mathbb{F}_q^m$ is a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ such that

$$1 \leq i \leq m, G_i(\boldsymbol{x}) = t_i$$

- Alice *signs*: $\boldsymbol{y}$ solution of $G(A^{-1}y) = t$ **linear** in $y_1, \ldots, y_m$.



UOV: Signing process

Signing

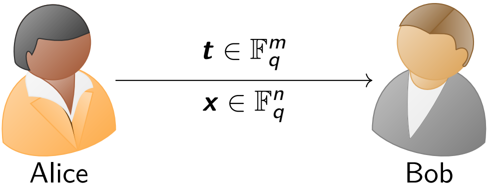
A **signature** for the message $\boldsymbol{t} \in \mathbb{F}_q^m$ is a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ such that

$$1 \leq i \leq m, G_i(\boldsymbol{x}) = t_i$$

- Alice *signs*: $\boldsymbol{y}$ solution of $G(A^{-1}y) = t$ **linear** in $y_1, \ldots, y_m$.
  Sample $y_{m+1}, \ldots, y_n$ uniformly and solve a **square linear system**.
  Return $\boldsymbol{x} = A^{-1}y$

$\boxed{(A, F)}$

$\boldsymbol{x} = Sign(G(\boldsymbol{t}))$



$\boxed{G}$

UOV: Signing process

Signing

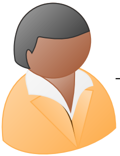
A **signature** for the message $\boldsymbol{t} \in \mathbb{F}_q^m$ is a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ such that

$$1 \leq i \leq m, G_i(\boldsymbol{x}) = t_i$$

- Alice *signs*: $\boldsymbol{y}$ solution of $G(A^{-1}y) = t$ **linear** in $y_1, \ldots, y_m$. Sample $y_{m+1}, \ldots, y_n$ uniformly and solve a **square linear system**. Return $\boldsymbol{x} = A^{-1}y$
- Bob verifies: checks that for $1 \leq i \leq m$, $G_i(\boldsymbol{x}) = t_i$.

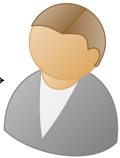
$\boxed{(A, F)}$

$\boldsymbol{x} = Sign(G(\boldsymbol{t}))$



Alice

$\xrightarrow[\boldsymbol{x} \in \mathbb{F}_q^n]{\boldsymbol{t} \in \mathbb{F}_q^m}$



Bob

$\boxed{G}$

$G(\boldsymbol{x}) \stackrel{?}{=} \boldsymbol{t}$

UOV: Signing process

Signing

A **signature** for the message $\boldsymbol{t} \in \mathbb{F}_q^m$ is a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ such that

$$1 \leq i \leq m, G_i(\boldsymbol{x}) = t_i$$

- Alice *signs*: $\boldsymbol{y}$ solution of $G(A^{-1}y) = t$ **linear** in $y_1, \ldots, y_m$.
  Sample $y_{m+1}, \ldots, y_n$ uniformly and solve a **square linear system**.
  Return $\boldsymbol{x} = A^{-1}y$
- Bob verifies: checks that for $1 \leq i \leq m$, $G_i(\boldsymbol{x}) = t_i$.

Hash-and-sign

In practice, $\boldsymbol{t} = \mathcal{H}(M), M \in \{0, 1\}^*$

UOV: Parameters

	NIST SL	n	m	$\mathbb{F}_q$	|pk| (bytes)	|sk| (bytes)	|cpk| (bytes)	|sig+salt| (bytes)
`ov-Ip`	1	112	44	$\mathbb{F}_{256}$	278 432	237 912	43 576	128
`ov-Is`	1	160	64	$\mathbb{F}_{16}$	412 160	348 720	66 576	96
`ov-III`	3	184	72	$\mathbb{F}_{256}$	1 225 440	1 044 336	189 232	200
`ov-V`	5	244	96	$\mathbb{F}_{256}$	2 869 440	2 436 720	446 992	260

Figure: Modern UOV[Beullens, Chen, Hung, Kannwischer, Peng, Shih, Yang 2023]

UOV: Alternative formulation

Equivalent characterisation of the trapdoor [Beullens 2020]

Trapdoor: subspace $\mathcal{O} \subset \mathbb{F}_q^n$ of dimension m such that

$$\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{O}^2, \quad \boldsymbol{x}^T G_1 \boldsymbol{y} = \cdots = \boldsymbol{x}^T G_m \boldsymbol{y} = 0$$

UOV: Alternative formulation

Equivalent characterisation of the trapdoor [Beullens 2020]

Trapdoor: subspace $\mathcal{O} \subset \mathbb{F}_q^n$ of dimension m such that

$$\forall(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{O}^2, \quad \boldsymbol{x}^T G_1 \boldsymbol{y} = \cdots = \boldsymbol{x}^T G_m \boldsymbol{y} = 0$$

Observation 1

The first m columns of A^{-1} form a basis of $\mathcal{O}$.

UOV: Alternative formulation

Equivalent characterisation of the trapdoor [Beullens 2020]

Trapdoor: subspace $\mathcal{O} \subset \mathbb{F}_q^n$ of dimension m such that

$$\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{O}^2, \quad \boldsymbol{x}^T G_1 \boldsymbol{y} = \cdots = \boldsymbol{x}^T G_m \boldsymbol{y} = 0$$

Observation 1

The first m columns of A^{-1} form a basis of $\mathcal{O}$.

Observation 2

All vectors in $\mathcal{O}$ are signatures of the message $(0, \ldots, 0) \in \mathbb{F}_q^m$, but the converse is false.

Cryptanalysis

Forgery

Goal: Find **a** signature $\boldsymbol{x} \in \mathbb{F}_q^n$ for a **single** message $\boldsymbol{t} \in \mathbb{F}_q^m$.

$$V(\boldsymbol{t}) := \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = t_i\}$$

Cryptanalysis

Forgery

Goal: Find **a** signature $\boldsymbol{x} \in \mathbb{F}_q^n$ for a **single** message $\boldsymbol{t} \in \mathbb{F}_q^m$.

$$V(\boldsymbol{t}) := \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = t_i\}$$

Computational problem: Find a point in a **variety of dimension** $n - m$

Cryptanalysis

Forgery

Goal: Find **a** signature $\boldsymbol{x} \in \mathbb{F}_q^n$ for a **single** message $\boldsymbol{t} \in \mathbb{F}_q^m$.

$$V(\boldsymbol{t}) := \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = t_i\}$$

Computational problem: Find a point in a **variety of dimension** $n - m$

Key recovery

Goal: find an equivalent secret key to sign **any** message.

$$\mathcal{O} \subset \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = 0\}$$

Cryptanalysis

Forgery

Goal: Find **a** signature $\boldsymbol{x} \in \mathbb{F}_q^n$ for a **single** message $\boldsymbol{t} \in \mathbb{F}_q^m$.

$$V(\boldsymbol{t}) := \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = t_i\}$$

Computational problem: Find a point in a **variety of dimension** $n - m$

Key recovery

Goal: find an equivalent secret key to sign **any** message.

$$\mathcal{O} \subset \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = 0\}$$

Computational problem: Find a **linear subspace of dimension** m **in** $V(0)$

Contribution

Input: $G, v \in \mathcal{O}$ $\xrightarrow{O(mn^{\omega})}$ **Output**: $\mathcal{O}$

Main result [P. 2023]

- Polynomial-time algorithm that takes as input **one vector** in $\mathcal{O}$ and the public key G, and returns a basis of $\mathcal{O}$.

Contribution

Input $G, v \in \mathcal{O}$ $\xrightarrow{O(mn^{\omega})}$ **Output** $\mathcal{O}$

Main result [P. 2023]

- Polynomial-time algorithm that takes as input **one vector** in $\mathcal{O}$ and the public key G, and returns a basis of $\mathcal{O}$.
- Polynomial-time algorithm that takes as input a vector $x \in \mathbb{F}_q^n$ and the public key G, and that answers the question "$x \in \mathcal{O}$?".

Contribution

Input		**Output**
$G, v \in \mathcal{O}$	$O(mn^{\omega})$	$\mathcal{O}$

Main result [P. 2023]

- Polynomial-time algorithm that takes as input **one vector** in $\mathcal{O}$ and the public key G, and returns a basis of $\mathcal{O}$.
- Polynomial-time algorithm that takes as input a vector $x \in \mathbb{F}_q^n$ and the public key G, and that answers the question "$x \in \mathcal{O}$?".

Consequence for the security of UOV

- An attacker needs to find a single vector in $\mathcal{O}$ to retrieve the **secret key** up to equivalence. This is enough to sign **any** message.

Contribution

Input $G, v \in \mathcal{O}$ $\xrightarrow{O(mn^{\omega})}$ $\mathcal{O}$ **Output**

Main result [P. 2023]

- Polynomial-time algorithm that takes as input **one vector** in $\mathcal{O}$ and the public key G, and returns a basis of $\mathcal{O}$.
- Polynomial-time algorithm that takes as input a vector $x \in \mathbb{F}_q^n$ and the public key G, and that answers the question "$x \in \mathcal{O}$?".

Consequence for the security of UOV

- An attacker needs to find a single vector in $\mathcal{O}$ to retrieve the **secret key** up to equivalence. This is enough to sign **any** message.
- Finding a vector of $\mathcal{O}$ remains challenging.

Contribution: Implementation

n	112	160	184	244
Time	1.7s	4.4s	5.7s	13.3s

Figure: Implementation of our attack with native **sagemath** functions on a laptop

Contribution: Implementation

n	112	160	184	244
Time	1.7s	4.4s	5.7s	13.3s

Figure: Implementation of our attack with native **sagemath** functions on a laptop

In the context of **side-channel attacks**, Aulbach, Campos, Krämer, Samardjiska, Stöttinger [1] previously obtained a similar result, with a practical key recovery from one vector.

[1] https://tches.iacr.org/index.php/TCHES/article/view/10962/10269

n	112	160	184	244
Time	19m34s		3h7m55s	11h41m7s

Figure: Implementation in the context of side-channel attacks

State-of-the-art of Key Recovery Attacks

Reconciliation [Ding, Yang, Chen, Chen, Cheng 2008], [Beullens 2020/21]

Key recovery attacks benefit from knowledge of some vectors of $\mathcal{O}$: additional equations in quadratic system.

State-of-the-art of Key Recovery Attacks

Reconciliation [Ding, Yang, Chen, Chen, Cheng 2008], [Beullens 2020/21]

Key recovery attacks benefit from knowledge of some vectors of $\mathcal{O}$: additional equations in quadratic system. $\rightarrow$ Reconciliation

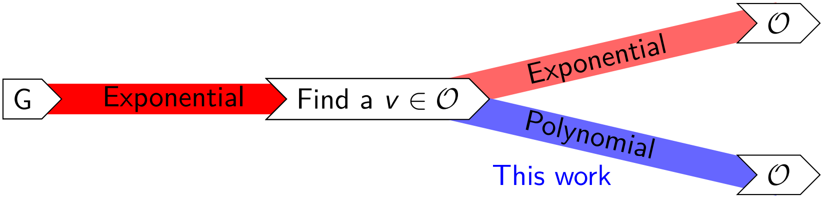
State-of-the-art of Key Recovery Attacks

Reconciliation [Ding, Yang, Chen, Chen, Cheng 2008], [Beullens 2020/21]

Key recovery attacks benefit from knowledge of some vectors of $\mathcal{O}$: additional equations in quadratic system. $\rightarrow$ Reconciliation

This work

Any vector in $\mathcal{O}$ characterizes it. $\rightarrow$ Polynomial reconciliation



Contribution: The algorithm

Equivalent characterisation of the trapdoor [Beullens 2020]

Trapdoor: subspace $\mathcal{O}$ of dimension m such that

$$\forall(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{O}^2, \quad \boldsymbol{x}^T G_1 \boldsymbol{y} = \cdots = \boldsymbol{x}^T G_m \boldsymbol{y} = 0$$

Contribution: The algorithm

Equivalent characterisation of the trapdoor [Beullens 2020]

Trapdoor: subspace $\mathcal{O}$ of dimension m such that

$$\forall(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{O}^2, \quad \boldsymbol{x}^T G_1 \boldsymbol{y} = \cdots = \boldsymbol{x}^T G_m \boldsymbol{y} = 0$$

Reformulation

$$\forall \boldsymbol{x} \in \mathcal{O}, \quad \mathcal{O} \subset J(\boldsymbol{x}) := \ker(\boldsymbol{x}^T G_1) \cap \ldots \cap \ker(\boldsymbol{x}^T G_m)$$

Contribution: The algorithm

Equivalent characterisation of the trapdoor [Beullens 2020]

Trapdoor: subspace $\mathcal{O}$ of dimension m such that

$$\forall(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{O}^2, \quad \boldsymbol{x}^T G_1 \boldsymbol{y} = \cdots = \boldsymbol{x}^T G_m \boldsymbol{y} = 0$$

Reformulation

$$\forall \boldsymbol{x} \in \mathcal{O}, \quad \mathcal{O} \subset J(\boldsymbol{x}) := \ker(\boldsymbol{x}^T G_1) \cap \ldots \cap \ker(\boldsymbol{x}^T G_m)$$

Observation

$J(\boldsymbol{x})$ is of dimension $n - m$ generically.

Contribution: The algorithm

Public key: $G \in (\mathbb{F}_q^{n\times n})^m$ Secret vector: $\boldsymbol{x} \in \mathbb{F}_q^n$ $\dim(J(\boldsymbol{x})) = n - m$

Reduction

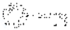
Restriction $G_{|J(\boldsymbol{x})} \rightarrow$ UOV instance with **smaller parameters**.

Contribution: The algorithm

Public key: $G \in (\mathbb{F}_q^{n\times n})^m$ Secret vector: $\boldsymbol{x} \in \mathbb{F}_q^n$ $\dim(J(\boldsymbol{x})) = n - m$

Reduction

Restriction $G_{|J(\boldsymbol{x})} \rightarrow$ UOV instance with **smaller parameters**.



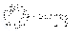
$$\longrightarrow \boxed{G, \boldsymbol{x}}$$

Contribution: The algorithm

Public key: $G \in (\mathbb{F}_q^{n\times n})^m$ Secret vector: $\boldsymbol{x} \in \mathbb{F}_q^n$ $\dim(J(\boldsymbol{x})) = n - m$

Reduction

Restriction $G_{|J(\boldsymbol{x})} \rightarrow$ UOV instance with **smaller parameters**.



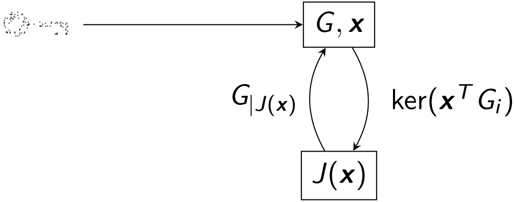
$$\begin{tikzcd} \boxed{G, \boldsymbol{x}} \arrow[d, bend left, "\ker(\boldsymbol{x}^T G_i)"] \\ \boxed{J(\boldsymbol{x})} \end{tikzcd}$$

Contribution: The algorithm

Public key: $G \in (\mathbb{F}_q^{n\times n})^m$ Secret vector: $\boldsymbol{x} \in \mathbb{F}_q^n$ $\dim(J(\boldsymbol{x})) = n - m$

Reduction

Restriction $G_{|J(\boldsymbol{x})} \to$ UOV instance with **smaller parameters**.

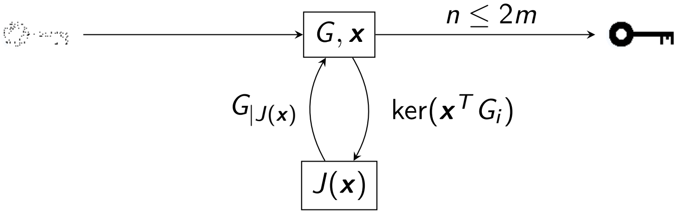


Contribution: The algorithm

Public key: $G \in (\mathbb{F}_q^{n \times n})^m$ Secret vector: $\boldsymbol{x} \in \mathbb{F}_q^n$ $\dim(J(\boldsymbol{x})) = n - m$

Reduction

Restriction $G_{|J(\boldsymbol{x})} \to$ UOV instance with **smaller parameters**.



Concluding the attack

$n \leq 2m \implies G_{|J(x)}$ is singular $\to$ broken in **polynomial time**.

Contribution: Complexity analysis

$$J(\boldsymbol{x}) = \ker \begin{pmatrix} \boldsymbol{x}^T G_1 \\ \vdots \\ \boldsymbol{x}^T G_m \end{pmatrix}$$

Public key: $G \in (\mathbb{F}_q^{n \times n})^m$ Secret vector: $\boldsymbol{x} \in \mathbb{F}_q^n$ $\dim(J(\boldsymbol{x})) = n - m$

Complexity of the attack

1. Computing B, a basis of $J(\boldsymbol{x})$ $O(n^\omega)$ and $2 \leq \omega \leq 3$

Contribution: Complexity analysis

$$J(\boldsymbol{x}) = \ker \begin{pmatrix} \boldsymbol{x}^T G_1 \\ \vdots \\ \boldsymbol{x}^T G_m \end{pmatrix}$$

Public key: $G \in (\mathbb{F}_q^{n\times n})^m$ Secret vector: $\boldsymbol{x} \in \mathbb{F}_q^n$ $\dim(J(\boldsymbol{x})) = n - m$

Complexity of the attack

1. Computing B, a basis of $J(\boldsymbol{x})$ — $O(n^\omega)$ and $2 \leq \omega \leq 3$
2. Computing the restrictions: $G_{i|J(\boldsymbol{x})} = B^T G_i B$ — $O(mn^\omega)$

Contribution: Complexity analysis

$$J(\boldsymbol{x}) = \ker \begin{pmatrix} \boldsymbol{x}^T G_1 \\ \vdots \\ \boldsymbol{x}^T G_m \end{pmatrix}$$

Public key: $G \in (\mathbb{F}_q^{n \times n})^m$ Secret vector: $\boldsymbol{x} \in \mathbb{F}_q^n$ $\dim(J(\boldsymbol{x})) = n - m$

Complexity of the attack

1. Computing B, a basis of $J(\boldsymbol{x})$ — $O(n^\omega)$ and $2 \leq \omega \leq 3$
2. Computing the restrictions: $G_{i|J(\boldsymbol{x})} = B^T G_i B$ — $O(mn^\omega)$
3. Kernel computations — $O(mn^\omega)$
4. Total cost: $\boldsymbol{O(mn^\omega)}$

Gap between key recovery and forgery

Key recovery versus forgery

- Experimentally, observe large gap between forgery attacks and key recovery attacks.

Gap between key recovery and forgery

Key recovery versus forgery

- Experimentally, observe large gap between forgery attacks and key recovery attacks.
- Key size: $G \in (\mathbb{F}_q^{n\times n})^m, n = \lceil 2.5m \rceil$

Gap between key recovery and forgery

Key recovery versus forgery

- Experimentally, observe large gap between forgery attacks and key recovery attacks.
- Key size: $G \in (\mathbb{F}_q^{n\times n})^m, n = \lceil 2.5m \rceil$

Gap between key recovery and forgery

Key recovery versus forgery

- Experimentally, observe large gap between forgery attacks and key recovery attacks.
- Key size: $G \in (\mathbb{F}_q^{n\times n})^m, n = \lceil 2.5m \rceil$

m	9	10	11	12	13	14	15	16	17
Forgery	0.1s	0.3s	1s	4s	20s	144s	930s	2h	14h
Recovery	40s	1h	2h	>11000h					

Figure: CPU-time in $\mathbb{F}_{31}$ with **msolve** [Berthomieu, Eder, Safey el Din, 2021]

Gap between key recovery and forgery

Key recovery versus forgery

- Experimentally, observe large gap between forgery attacks and key recovery attacks.
- Key size: $G \in (\mathbb{F}_q^{n\times n})^m, n = \lceil 2.5m \rceil$

m	9	10	11	12	13	14	15	16	17
Forgery	0.1s	0.3s	1s	4s	20s	144s	930s	2h	14h
Recovery	40s	1h	2h	>11000h					

Figure: CPU-time in $\mathbb{F}_{31}$ with **msolve** [Berthomieu, Eder, Safey el Din, 2021]

Key Recovery

This is the time it takes to retrieve **one** vector in $\mathcal{O}$.

Forgery attacks are key-recovery attacks

Forgery

Goal: forge **a** signature $\boldsymbol{x} \in \mathbb{F}_q^n$ for a **single** message $M \in \mathbb{F}_q^m$.

$$V(M) = \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = M_i\}$$

Reminder: $\mathcal{O} \subset V(O)$

Forgery attacks are key-recovery attacks

Forgery

Goal: forge **a** signature $\boldsymbol{x} \in \mathbb{F}_q^n$ for a **single** message $M \in \mathbb{F}_q^m$.

$$V(M) = \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = M_i\}$$

Reminder: $\mathcal{O} \subset V(O)$

Key recovery from forgery

Attempt to forge a signature $\boldsymbol{x}$ for the message 0 until $\boldsymbol{x}$ belongs to $\mathcal{O}$.

Forgery attacks are key-recovery attacks

Forgery

Goal: forge **a** signature $\boldsymbol{x} \in \mathbb{F}_q^n$ for a **single** message $M \in \mathbb{F}_q^m$.

$$V(M) = \{\boldsymbol{x} \in \mathbb{F}_q^n \quad | \quad \forall i \leq m, G_i(\boldsymbol{x}) = M_i\}$$

Reminder: $\mathcal{O} \subset V(O)$

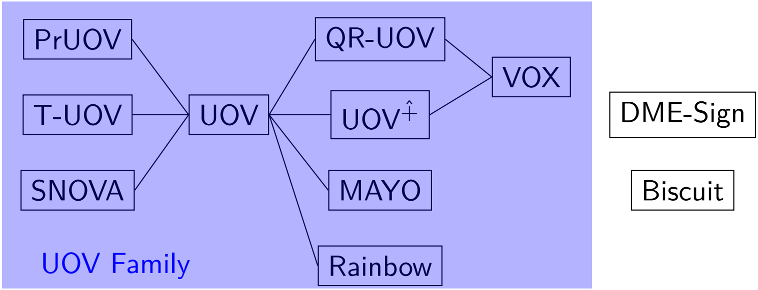
Key recovery from forgery

Attempt to forge a signature $\boldsymbol{x}$ for the message 0 until $\boldsymbol{x}$ belongs to $\mathcal{O}$.

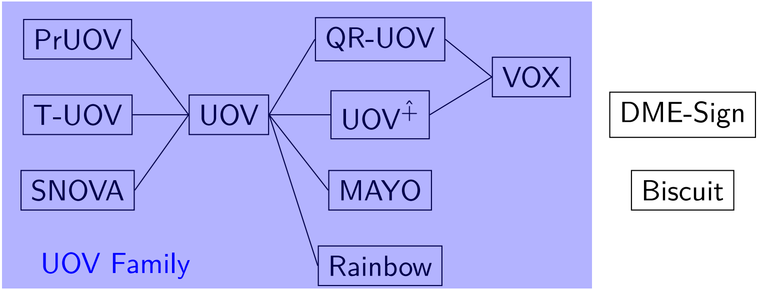
n	112	160	184	244
Time	0.2s	0.5s	0.7s	1.5s

Figure: Implementation of our test $\boldsymbol{x} \in \mathcal{O}$? on a laptop

Multivariate Post-Quantum Zoo



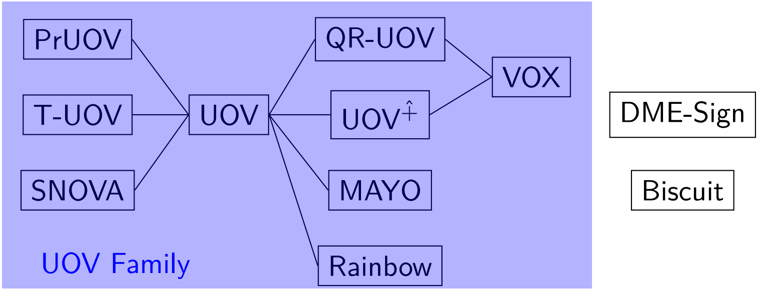
Multivariate Post-Quantum Zoo



The UOV family

- "Multi-layer structure": Rainbow [DY05, Beu22]

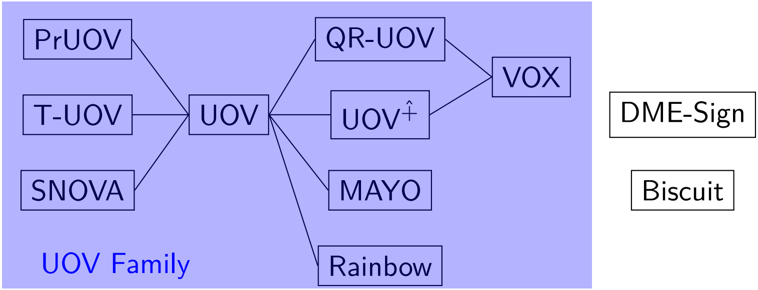
Multivariate Post-Quantum Zoo



The UOV family

- "Multi-layer structure": Rainbow [DY05, Beu22]
- MAYO: key size/signature size trade-off. [Beu21]

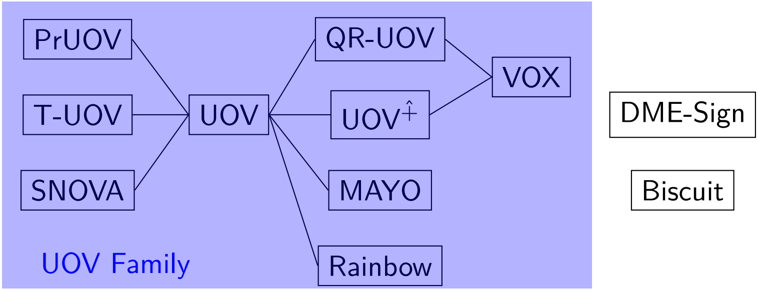
Multivariate Post-Quantum Zoo



The UOV family

- "Multi-layer structure": Rainbow [DY05, Beu22]
- MAYO: key size/signature size trade-off. [Beu21]
- Structured keys: QR-UOV, VOX, SNOVA [FIKT20, WTKC22]

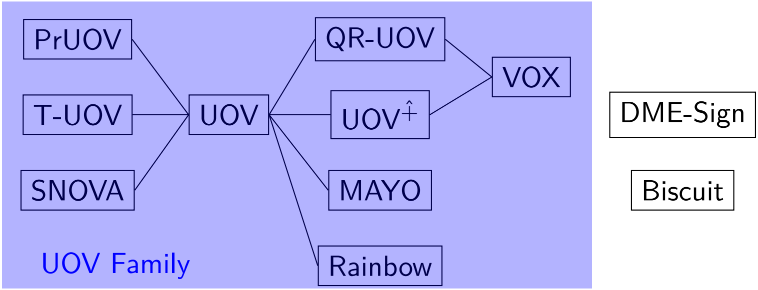
Multivariate Post-Quantum Zoo



The UOV family

- "Multi-layer structure": Rainbow [DY05, Beu22]
- MAYO: key size/signature size trade-off. [Beu21]
- Structured keys: QR-UOV, VOX, SNOVA [FIKT20, WTKC22]
- "Noisy" public key to increase security: UOV$\hat{+}$, VOX [CFFG+23]

Multivariate Post-Quantum Zoo



The UOV family

- "Multi-layer structure": Rainbow [DY05, Beu22]
- MAYO: key size/signature size trade-off. [Beu21]
- Structured keys: QR-UOV, VOX, SNOVA [FIKT20, WTKC22]
- "Noisy" public key to increase security: UOV$^{\hat{+}}$, VOX [CFFG+23]
- Formal security proof: T-UOV, PrUOV [DGGH+23], [CFFG+23]

Application to UOV variants in the NIST competition

For schemes that are instances of UOV $\rightarrow$ direct application

- QR-UOV
- SNOVA
- PrUOV

Application to UOV variants in the NIST competition

For schemes that are instances of UOV $\rightarrow$ direct application

- QR-UOV
- SNOVA
- PrUOV

Result already known on MAYO [Beullens 2021]

Application to UOV variants in the NIST competition

For schemes that are instances of UOV $\rightarrow$ direct application

- QR-UOV
- SNOVA
- PrUOV

Result already known on MAYO [Beullens 2021]

More work required for schemes using modified UOV keys.

- $\text{UOV}^{\hat{+}}$ (VOX/FOX)
- T-UOV

Perspectives

Contribution [P. 2023]

- One secret vector $\rightarrow$ polynomial key recovery.
- Distinguish secret vectors from random signatures of 0.

Perspectives

Contribution [P. 2023]

- One secret vector $\rightarrow$ polynomial key recovery.
- Distinguish secret vectors from random signatures of 0.

New directions

- Efficiently generalize tools to UOV-inspired schemes: T-UOV, VOX

Perspectives

Contribution [P. 2023]

- One secret vector $\rightarrow$ polynomial key recovery.
- Distinguish secret vectors from random signatures of 0.

New directions

- Efficiently generalize tools to UOV-inspired schemes: T-UOV, VOX
- Key recovery attacks targeting one vector

Perspectives

Contribution [P. 2023]

- One secret vector $\rightarrow$ polynomial key recovery.
- Distinguish secret vectors from random signatures of 0.

New directions

- Efficiently generalize tools to UOV-inspired schemes: T-UOV, VOX
- Key recovery attacks targeting one vector

Links

`https://eprint.iacr.org/2023/1131`
`https://github.com/pi-r2/OneVector`