Cryptography
Research Center

Improved Attacks for SNOVA by Exploiting Stability under a Group Action

Daniel Cabarcas, Peigen Li, Javier Verbel, and Ricardo Villanueva-Polanco

https://eprint.iacr.org/2024/1770

Crypto 2025

Technology
Innovation
Institute

SNOVA: A Signature Scheme based on UOV

- **2nd round candidate** in the NIST process for post-quantum signatures.

SNOVA: A Signature Scheme based on UOV

- **2nd round candidate** in the NIST process for post-quantum signatures.
- **Aims to reduce the** pk**-size of UOV**.

SNOVA: A Signature Scheme based on UOV

◇ **2nd round candidate** in the NIST process for post-quantum signatures.

◇ **Aims to reduce the** pk**-size of UOV**.

◇ **Fast and compact** when compared with similar proposals.

SNOVA: A Signature Scheme based on UOV

- **2nd round candidate** in the NIST process for post-quantum signatures.
- **Aims to reduce the** pk**-size of UOV**.
- **Fast and compact** when compared with similar proposals.
- **Based on a new construction**: Several attacks since submitted (e.g., [IA24, LD24, Beu25, NTF24])

Our contributions

1. Analysis algebraic properties of SNOVA systems.

2. New key-recovery attack.

3. New forgery attack.

Contents

Introduction

SNOVA: A UOV-like Signature Scheme

[0] Singing-time of ESK versions `https://pqsort.tii.ae/`. Verify-time $\approx$ sign-time/2.

SNOVA: A UOV-like Signature Scheme

A *keypair* (sk, pk) $\in$ UOV(q, n, o, m):

sk $= \mathcal{O} \leq \mathbb{F}_q^n$ with dim$(\mathcal{O}) = o$.

pk $= (p_1, \ldots, p_m) \in \mathbb{F}_q[x_1, \ldots, x_n]$ with $\deg(p_i) = 2$ and

$$p_1(\mathbf{o}) = \cdots = p_m(\mathbf{o}) = 0 \quad \forall \mathbf{o} \in \mathcal{O}.$$

[0] Singing-time of ESK versions `https://pqsort.tii.ae/`. Verify-time $\approx$ sign-time/2.

SNOVA: A UOV-like Signature Scheme

A *keypair* $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{UOV}(q, n, o, m)$:

$\mathsf{sk} = \mathcal{O} \leq \mathbb{F}_q^n$ with $\dim(\mathcal{O}) = o$.

$\mathsf{pk} = (p_1, \ldots, p_m) \in \mathbb{F}_q[x_1, \ldots, x_n]$ with $\deg(p_i) = 2$ and

$$p_1(\mathbf{o}) = \cdots = p_m(\mathbf{o}) = 0 \quad \forall \mathbf{o} \in \mathcal{O}.$$

A signature $\sigma = (\mathsf{s}, \mathsf{salt}) \Rightarrow \tilde{\mathsf{pk}}(\mathsf{s}) = \mathsf{Hash}(\mathtt{message} \| \mathtt{salt}) \in \mathbb{F}_q^m$

Verification map: $\tilde{\mathsf{pk}} = \mathsf{Expand}(\mathsf{pk})$

[0] Singing-time of ESK versions `https://pqsort.tii.ae/`. Verify-time ≈ sign-time/2.

SNOVA: A UOV-like Signature Scheme

A *keypair* $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{UOV}(q, n, o, m)$:

$\mathsf{sk} = \mathcal{O} \leq \mathbb{F}_q^n$ with $\dim(\mathcal{O}) = o$.

$\mathsf{pk} = (p_1, \dots, p_m) \in \mathbb{F}_q[x_1, \dots, x_n]$ with $\deg(p_i) = 2$ and

$$p_1(\mathbf{o}) = \cdots = p_m(\mathbf{o}) = 0 \quad \forall \mathbf{o} \in \mathcal{O}.$$

A signature $\sigma = (\mathsf{s}, \mathsf{salt}) \Rightarrow \tilde{\mathsf{pk}}(\mathsf{s}) = \mathsf{Hash}(\mathsf{message} \| \mathsf{salt}) \in \mathbb{F}_q^m$

Verification map:
$\tilde{\mathsf{pk}} = \mathsf{Expand}(\mathsf{pk})$

For level I:

$|\text{pk-SNOVA}| \approx$ 1KB, 2KB and 10KB

sign-time ≈ 0.5Mc, 0.4Mc and 0.3Mc

[0] Singing-time of ESK versions `https://pqsort.tii.ae/`. Verify-time ≈ sign-time/2.

SNOVA Sequences

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l \times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{S}^i & & \\ & \ddots & \\ & & \mathbf{S}^i \end{bmatrix}$.

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l\times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{S}^i & & \\ & \ddots & \\ & & \mathbf{S}^i \end{bmatrix}$.

◇ Given $\mathbf{P} \in \mathbb{F}_q^{n'\times n'}$ define

$$\mathcal{F}_{\mathbf{P}}(\mathbf{u}) := \left(\qquad \right)$$

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l\times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{S}^i & & \\ & \ddots & \\ & & \mathbf{S}^i \end{bmatrix}$.

◇ Given $\mathbf{P} \in \mathbb{F}_q^{n'\times n'}$ define

$$\mathcal{F}_{\mathbf{P}}(\mathbf{u}) := \left(\quad \mathbf{u}^t \cdot \mathbf{P} \cdot \mathbf{u}, \quad \right)$$

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l \times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{S}^i & & \\ & \ddots & \\ & & \mathbf{S}^i \end{bmatrix}$.

- Given $\mathbf{P} \in \mathbb{F}_q^{n' \times n'}$ define

$$\mathcal{F}_{\mathbf{P}}(\mathbf{u}) := \left(\quad \mathbf{u}^t \cdot \mathbf{P} \cdot \mathbf{u}, \quad \mathbf{u}^t \cdot (\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, \quad \right)$$

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l\times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{S}^i & & \\ & \ddots & \\ & & \mathbf{S}^i \end{bmatrix}$.

◇ Given $\mathbf{P} \in \mathbb{F}_q^{n'\times n'}$ define

$$\mathcal{F}_{\mathbf{P}}(\mathbf{u}) := \left(\begin{array}{ccc} \mathbf{u}^t \cdot \mathbf{P} \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}) \cdot \mathbf{u}, \\ & & \end{array} \right)$$

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l\times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{S}^i & & \\ & \ddots & \\ & & \mathbf{S}^i \end{bmatrix}$.

◇ Given $\mathbf{P} \in \mathbb{F}_q^{n'\times n'}$ define

$$\mathcal{F}_{\mathbf{P}}(\mathbf{u}) := \begin{pmatrix} \mathbf{u}^t \cdot \mathbf{P} \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}) \cdot \mathbf{u}, \\ \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & & \end{pmatrix}$$

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l\times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{s}^i & & \\ & \ddots & \\ & & \mathbf{s}^i \end{bmatrix}$.

◇ Given $\mathbf{P} \in \mathbb{F}_q^{n'\times n'}$ define

$$\mathcal{F}_{\mathbf{P}}(\mathbf{u}) := \begin{pmatrix} \mathbf{u}^t \cdot \mathbf{P} \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}) \cdot \mathbf{u}, \\ \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & \cdots & \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}^{l-1}}\mathbf{P}\Lambda_{\mathbf{S}^{l-1}}) \cdot \mathbf{u} \end{pmatrix}$$

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l\times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{S}^i & & \\ & \ddots & \\ & & \mathbf{S}^i \end{bmatrix}$.

- Given $\mathbf{P} \in \mathbb{F}_q^{n'\times n'}$ define

$$\mathcal{F}_{\mathbf{P}}(\mathbf{u}) := \begin{pmatrix} \mathbf{u}^t \cdot \mathbf{P} \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}) \cdot \mathbf{u}, \\ \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & \cdots & \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}^{l-1}}\mathbf{P}\Lambda_{\mathbf{S}^{l-1}}) \cdot \mathbf{u} \end{pmatrix}$$

- A *SNOVA sequence* is set of the form $(\mathcal{F}_{\mathbf{P}_1}, \mathcal{F}_{\mathbf{P}_2}, \ldots, \mathcal{F}_{\mathbf{P}_m})$.

SNOVA Sequences

$\mathbf{S} \in \mathbb{F}_q^{l\times l}$ with CharPoly$(\mathbf{S})$ =irreducible and $\Lambda_{\mathbf{S}^i} = \begin{bmatrix} \mathbf{S}^i & & \\ & \ddots & \\ & & \mathbf{S}^i \end{bmatrix}$.

◇ Given $\mathbf{P} \in \mathbb{F}_q^{n'\times n'}$ define

$$\mathcal{F}_{\mathbf{P}}(\mathbf{u}) := \begin{pmatrix} \mathbf{u}^t \cdot \mathbf{P} \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}) \cdot \mathbf{u}, \\ \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}}\mathbf{P}\Lambda_{\mathbf{S}}) \cdot \mathbf{u}, & \cdots & \mathbf{u}^t \cdot (\Lambda_{\mathbf{S}^{l-1}}\mathbf{P}\Lambda_{\mathbf{S}^{l-1}}) \cdot \mathbf{u} \end{pmatrix}$$

◇ A *SNOVA sequence* is set of the form $(\mathcal{F}_{\mathbf{P}_1}, \mathcal{F}_{\mathbf{P}_2}, \ldots, \mathcal{F}_{\mathbf{P}_m})$.

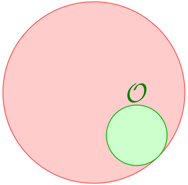
■ $\forall$ pk is associated to a SNOVA sequence $\mathcal{F}$.

Attacks using a SNOVA Sequences $\mathcal{F}$ [IA24, LD24, Beu25]

Attacks using a SNOVA Sequences $\mathcal{F}$ [IA24, LD24, Beu25]

- **Reconciliation** (*key-recovery*) attacks ⇒ Find $\mathbf{u} \in \mathcal{O}$ (the secret space) such that

$$\mathcal{F}(\mathbf{u}) = (0, \ldots, 0), \quad \text{where}$$

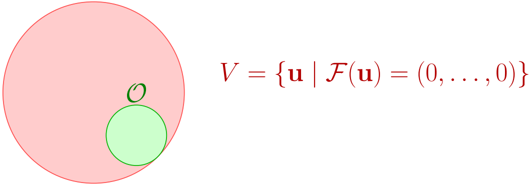


$V = \{\mathbf{u} \mid \mathcal{F}(\mathbf{u}) = (0, \ldots, 0)\}$

Attacks using a SNOVA Sequences $\mathcal{F}$ [IA24, LD24, Beu25]

- **Reconciliation** (*key-recovery*) attacks $\Rightarrow$ Find $\mathbf{u} \in \mathcal{O}$ (the secret space) such that

$$\mathcal{F}(\mathbf{u}) = (0, \ldots, 0),$$ where



- **Beullens** (*forgery*) attack $\Rightarrow$ Find $\mathbf{u} \in \mathbb{F}_q^{n'}$

$$\mathbf{E} \cdot \mathcal{F}(\mathbf{u}) + \mathcal{L}_{\text{linear}}(\mathbf{u}) = (a_1, \ldots, a_{ol^2}),$$

where $\mathbf{E}$ is a known matrix.

New Key-recovery Attack

Main Theorem: Given a SNOVA sequence $\mathcal{F} = (f_1, \ldots, f_{ml^2}) \subset \mathbb{F}_q[\mathbf{u}]$

Main Theorem: Given a SNOVA sequence $\mathcal{F} = (f_1, \ldots, f_{ml^2}) \subset \mathbb{F}_q[\mathbf{u}]$

We can compute matrices $\mathbf{P}$ and $\mathbf{A}$ over $\mathbb{F}_{q^l}$ such that

$$\mathcal{H} = \mathbf{A} \cdot (f_1^{\Lambda_{\mathbf{P}}}, \ldots, f_{ml^2}^{\Lambda_{\mathbf{P}}})^t \subset \mathbb{F}_{q^l}[\mathbf{u}],$$

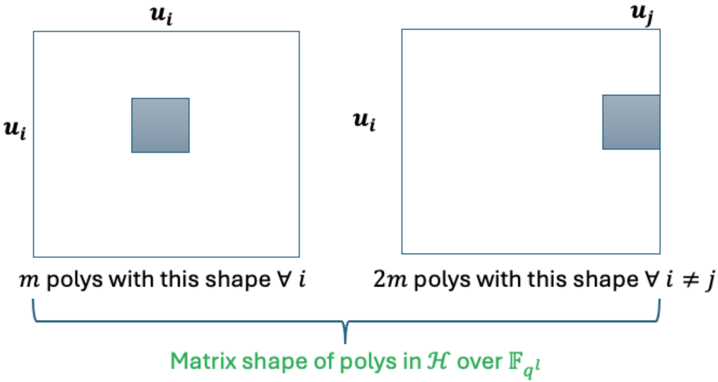
and polys in $\mathcal{H}$ are multi-homogeneous wrt $\mathbf{u} = \mathbf{u}_1 \sqcup \cdots \sqcup \mathbf{u}_l$.

Main Theorem: Given a SNOVA sequence $\mathcal{F} = (f_1, \ldots, f_{ml^2}) \subset \mathbb{F}_q[\mathbf{u}]$

We can compute matrices $\mathbf{P}$ and $\mathbf{A}$ over $\mathbb{F}_{q^l}$ such that

$$\mathcal{H} = \mathbf{A} \cdot (f_1^{\Lambda_\mathbf{P}}, \ldots, f_{ml^2}^{\Lambda_\mathbf{P}})^t \subset \mathbb{F}_{q^l}[\mathbf{u}],$$

and polys in $\mathcal{H}$ are multi-homogeneous wrt $\mathbf{u} = \mathbf{u}_1 \sqcup \cdots \sqcup \mathbf{u}_l$.



The Multi-Homogeneous (MH) XL Algorithm

The Multi-Homogeneous (MH) XL Algorithm

Consider the **multi-homogeneous** polynomial system

$$f_1(\mathbf{u}) = \cdots = f_{ml^2}(\mathbf{u}) = 0$$

The Multi-Homogeneous (MH) XL Algorithm

Consider the **multi-homogeneous** polynomial system

$$f_1(\mathbf{u}) = \cdots = f_{ml^2}(\mathbf{u}) = 0$$

XL: *input* an integer D

1. Solve $\mathbf{M} \cdot \mathbf{z} = 0$ for $\mathbf{z} \neq 0$, and

$$\mathbf{M} = \textbf{Macaulay}\left(f \;\middle|\; \begin{array}{l} \deg(f) \leq D \\ f = \text{mon} \cdot f_i \end{array} \right)$$

2. Extract a solution $\mathbf{u}$ from $\mathbf{z}$.

The Multi-Homogeneous (MH) XL Algorithm

Consider the **multi-homogeneous** polynomial system

$$f_1(\mathbf{u}) = \cdots = f_{ml^2}(\mathbf{u}) = 0$$

XL: *input* an integer D

1. Solve $\mathbf{M} \cdot \mathbf{z} = 0$ for $\mathbf{z} \neq 0$, and

$$\mathbf{M} = \textbf{Macaulay}\left(f \mid \begin{array}{c} \deg(f) \leq D \\ f = \text{mon} \cdot f_i \end{array} \right)$$

2. Extract a solution $\mathbf{u}$ from $\mathbf{z}$.

Multi-homogeneous-XL: *Input* a l-tuple $\mathbf{d}$

1. Solve $\mathbf{M} \cdot \mathbf{z} = 0$ for $\mathbf{z} \neq 0$, and

$$\mathbf{M} = \textbf{Macaulay}\left(f \mid \begin{array}{c} \text{multi-}\deg(f) \leq \mathbf{d} \\ f = \text{mon} \cdot f_i \end{array} \right)$$

2. Extract a solution $\mathbf{u}$ from $\mathbf{z}$.

$\mathbf{e} < \mathbf{d}$ iif $\mathbf{e} \neq \mathbf{d}$ and $\forall e_i \leq d_i$

The Multi-Homogeneous (MH) XL Algorithm

Consider the **multi-homogeneous** polynomial system

$$f_1(\mathbf{u}) = \cdots = f_{ml^2}(\mathbf{u}) = 0$$

XL: *input* an integer D

1. Solve $\mathbf{M} \cdot \mathbf{z} = 0$ for $\mathbf{z} \neq 0$, and

$$\mathbf{M} = \textbf{Macaulay}\left(f \mid \begin{array}{c} \deg(f) \leq D \\ f = \text{mon} \cdot f_i \end{array} \right)$$

2. Extract a solution $\mathbf{u}$ from $\mathbf{z}$.

Multi-homogeneous-XL: *Input* a l-tuple $\mathbf{d}$

1. Solve $\mathbf{M} \cdot \mathbf{z} = 0$ for $\mathbf{z} \neq 0$, and

$$\mathbf{M} = \textbf{Macaulay}\left(f \mid \begin{array}{c} \text{multi-}\deg(f) \leq \mathbf{d} \\ f = \text{mon} \cdot f_i \end{array} \right)$$

2. Extract a solution $\mathbf{u}$ from $\mathbf{z}$.

Multi-homogeneous-XL yields **smaller** Macaulay matrices.

$\mathbf{e} < \mathbf{d}$ iif $\mathbf{e} \neq \mathbf{d}$ and $\forall e_i \leq d_i$

Solving a SNOVA System $(f_1(\mathbf{u}), \ldots, f_{ml^2}(\mathbf{u})) = \mathbf{a}$

Solving a SNOVA System $(f_1(\mathbf{u}), \ldots, f_{ml^2}(\mathbf{u})) = \mathbf{a}$

1. Compute $\mathcal{H} = \mathbf{A} \cdot (f_1^{\Lambda_{\mathbf{P}}}, \ldots, f_{ml^2}^{\Lambda_{\mathbf{P}}})^t$ (multi-homogeneous over $\mathbb{F}_{q^l}$)
2. Use MH-XL to solve $\mathcal{H}(\tilde{\mathbf{u}}) = \tilde{\mathbf{a}}$, for $\tilde{\mathbf{u}} \in \mathbb{F}_{q^l}^{n'}$.
3. If $\mathbf{u} = \Lambda_{\mathbf{P}} \cdot \tilde{\mathbf{u}} \subset \mathbb{F}_q$, **output** $\mathbf{u}$. Otherwise, go to step 2.

Solving a SNOVA System $(f_1(\mathbf{u}), \ldots, f_{ml^2}(\mathbf{u})) = \mathbf{a}$

1. Compute $\mathcal{H} = \mathbf{A} \cdot (f_1^{\Lambda_\mathbf{P}}, \ldots, f_{ml^2}^{\Lambda_\mathbf{P}})^t$ (multi-homogeneous over $\mathbb{F}_{q^l}$)
2. Use MH-XL to solve $\mathcal{H}(\tilde{\mathbf{u}}) = \tilde{\mathbf{a}}$, for $\tilde{\mathbf{u}} \in \mathbb{F}_{q^l}^{n'}$.
3. If $\mathbf{u} = \Lambda_\mathbf{P} \cdot \tilde{\mathbf{u}} \subset \mathbb{F}_q$, **output** $\mathbf{u}$. Otherwise, go to step 2.

- Use hybrid approach over $\mathbb{F}_q$ at step 2.
- Complexity estimation of MH-XL.
- Experimental verification expected behavior of MH-XL.

Solving a SNOVA System $(f_1(\mathbf{u}), \ldots, f_{ml^2}(\mathbf{u})) = \mathbf{a}$

1. Compute $\mathcal{H} = \mathbf{A} \cdot (f_1^{\Lambda_{\mathbf{P}}}, \ldots, f_{ml^2}^{\Lambda_{\mathbf{P}}})^t$ (multi-homogeneous over $\mathbb{F}_{q^l}$)
2. Use MH-XL to solve $\mathcal{H}(\tilde{\mathbf{u}}) = \tilde{\mathbf{a}}$, for $\tilde{\mathbf{u}} \in \mathbb{F}_{q^l}^{n'}$.
3. If $\mathbf{u} = \Lambda_{\mathbf{P}} \cdot \tilde{\mathbf{u}} \subset \mathbb{F}_q$, **output** $\mathbf{u}$. Otherwise, go to step 2.

- Use hybrid approach over $\mathbb{F}_q$ at step 2.
- Complexity estimation of MH-XL.
- Experimental verification expected behavior of MH-XL.

Security level	l	previous best reconciliation attack	**our attack**
I	2	197	195
	3	196	187
	4	269	252

New Forgery Attack

Forgery Attack by Beullens [Beu25]

Let $\widetilde{\mathsf{pk}}(\mathbf{U})$ be the **verification map**, with $\mathbf{U} \in \mathbb{F}_q^{n' \times l}$

Forgery Attack by Beullens [Beu25]

Let $\tilde{\text{pk}}(\mathbf{U})$ be the **verification map**, with $\mathbf{U} \in \mathbb{F}_q^{n' \times l}$

- After a **change of vars.** $\mathbf{U} = \text{ch}(\mathbf{u})$ with $\mathbf{u} \in \mathbb{F}_q^{n'}$, (∃ many ch of that kind)

Forgery Attack by Beullens [Beu25]

Let $\tilde{\mathsf{pk}}(\mathbf{U})$ be the **verification map**, with $\mathbf{U} \in \mathbb{F}_q^{n' \times l}$

- After a **change of vars.** $\mathbf{U} = \mathsf{ch}(\mathbf{u})$ with $\mathbf{u} \in \mathbb{F}_q^{n'}$, (∃ many ch of that kind)

$$\tilde{\mathsf{pk}}(\mathbf{u}) = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix} \cdot \mathcal{F}(\mathbf{u}) + \mathcal{L}_{\mathsf{linear}}(\mathbf{u}) + \mathbf{c},$$

($\mathcal{F}$ assoc. SNOVA seq. to $\tilde{\mathsf{pk}}$)

Forgery Attack by Beullens [Beu25]

Let $\tilde{\mathsf{pk}}(\mathbf{U})$ be the **verification map**, with $\mathbf{U} \in \mathbb{F}_q^{n' \times l}$

◇ After a **change of vars.** $\mathbf{U} = \mathsf{ch}(\mathbf{u})$ with $\mathbf{u} \in \mathbb{F}_q^{n'}$, (∃ many ch of that kind)

$$\tilde{\mathsf{pk}}(\mathbf{u}) = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix} \cdot \mathcal{F}(\mathbf{u}) + \mathcal{L}_{\mathsf{linear}}(\mathbf{u}) + \mathbf{c},$$

($\mathcal{F}$ assoc. SNOVA seq. to $\tilde{\mathsf{pk}}$)

Attack: Given $r <$ Ncols$(\tilde{\mathbf{E}}_{\mathsf{ch}})$:

1. Brute-force ch with rank$(\tilde{\mathbf{E}}_{\mathsf{ch}}) = r$.
2. Solve the **easier** system involving
$$\tilde{\mathsf{pk}}(\mathbf{u}) = \mathsf{Hash}(\mathtt{message} \| \mathtt{salt}).$$
3. **Output** $\sigma = (\mathsf{ch}^{-1}(\mathbf{u}), \mathtt{salt})$

Forgery Attack by Beullens [Beu25]

Let $\tilde{\mathsf{pk}}(\mathbf{U})$ be the **verification map**, with $\mathbf{U} \in \mathbb{F}_q^{n' \times l}$

◇ After a **change of vars.** $\mathbf{U} = \mathsf{ch}(\mathbf{u})$ with $\mathbf{u} \in \mathbb{F}_q^{n'}$, (∃ many ch of that kind)

$$\tilde{\mathsf{pk}}(\mathbf{u}) = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix} \cdot \mathcal{F}(\mathbf{u}) + \mathcal{L}_{\mathsf{linear}}(\mathbf{u}) + \mathbf{c},$$

($\mathcal{F}$ assoc. SNOVA seq. to $\tilde{\mathsf{pk}}$)

Attack: Given $r < \mathsf{Ncols}(\tilde{\mathbf{E}}_{\mathsf{ch}})$:

1. Brute-force ch with $\mathsf{rank}(\tilde{\mathbf{E}}_{\mathsf{ch}}) = r$.
2. Solve the **easier** system involving

$$\tilde{\mathsf{pk}}(\mathbf{u}) = \mathsf{Hash}(\texttt{message} \| \texttt{salt}).$$

3. **Output** $\sigma = (\mathsf{ch}^{-1}(\mathbf{u}), \texttt{salt})$

Our goal: Exploit the structure of

$$\mathcal{H}(\tilde{\mathbf{u}}) = \mathbf{A} \cdot \mathcal{F}^{\Lambda_{\mathbf{P}}}(\tilde{\mathbf{u}}) + \text{low-rank of } \tilde{\mathbf{E}}_{\mathsf{ch}}$$

Forgery Attack by Beullens [Beu25]

Let $\tilde{\text{pk}}(\mathbf{U})$ be the **verification map**, with $\mathbf{U} \in \mathbb{F}_q^{n' \times l}$

◇ After a **change of vars.** $\mathbf{U} = \text{ch}(\mathbf{u})$ with $\mathbf{u} \in \mathbb{F}_q^{n'}$, (∃ many ch of that kind)

$$\tilde{\text{pk}}(\mathbf{u}) = \begin{bmatrix} \tilde{\mathbf{E}}_{\text{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\text{ch}} \end{bmatrix} \cdot \mathcal{F}(\mathbf{u}) + \mathcal{L}_{\text{linear}}(\mathbf{u}) + \mathbf{c},$$

($\mathcal{F}$ assoc. SNOVA seq. to $\tilde{\text{pk}}$)

Attack: Given $r < \text{Ncols}(\tilde{\mathbf{E}}_{\text{ch}})$:

1. Brute-force ch with $\text{rank}(\tilde{\mathbf{E}}_{\text{ch}}) = r$.
2. Solve the **easier** system involving

$$\tilde{\text{pk}}(\mathbf{u}) = \text{Hash}(\texttt{message} \| \texttt{salt}).$$

3. **Output** $\sigma = (\text{ch}^{-1}(\mathbf{u}), \texttt{salt})$

Our goal: Exploit the structure of

$$\mathcal{H}(\tilde{\mathbf{u}}) = \mathbf{A} \cdot \mathcal{F}^{\Lambda_{\mathbf{P}}}(\tilde{\mathbf{u}}) + \text{low-rank of } \tilde{\mathbf{E}}_{\text{ch}}$$

Main issue:

$$\begin{bmatrix} \tilde{\mathbf{E}}_{\text{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\text{ch}} \end{bmatrix} \cdot \mathcal{H}(\tilde{\mathbf{u}}) \text{ isn't multi-homogeneous.}$$

Our Forgery Attack

Our Forgery Attack

Use a (slightly) different ch so that $\tilde{\mathsf{pk}}(\mathbf{u}) = \mathbf{E}_{\mathsf{ch}} \cdot \mathcal{F}(\mathbf{u})$, with $\mathbf{E}_{\mathsf{ch}} = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix}$

Our Forgery Attack

Use a (slightly) different ch so that $\tilde{\mathsf{pk}}(\mathbf{u}) = \mathbf{E}_{\mathsf{ch}} \cdot \mathcal{F}(\mathbf{u})$, with $\mathbf{E}_{\mathsf{ch}} = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix}$

Attack: Given $r < \mathsf{Ncols}(\tilde{\mathbf{E}}_{\mathsf{ch}})$:

Our Forgery Attack

Use a (slightly) different ch so that $\tilde{\mathsf{pk}}(\mathbf{u}) = \mathbf{E}_{\mathsf{ch}} \cdot \mathcal{F}(\mathbf{u})$, with $\mathbf{E}_{\mathsf{ch}} = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix}$

Attack: Given $r < \mathsf{Ncols}(\tilde{\mathbf{E}}_{\mathsf{ch}})$:

1. Brute-force ch with $\mathsf{rank}(\tilde{\mathbf{E}}_{\mathsf{ch}}) = r$.

Our Forgery Attack

Use a (slightly) different ch so that $\tilde{\mathsf{pk}}(\mathbf{u}) = \mathbf{E}_{\mathsf{ch}} \cdot \mathcal{F}(\mathbf{u})$, with $\mathbf{E}_{\mathsf{ch}} = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix}$

Attack: Given $r < \mathsf{Ncols}(\tilde{\mathbf{E}}_{\mathsf{ch}})$:

1. Brute-force ch with $\mathsf{rank}(\tilde{\mathbf{E}}_{\mathsf{ch}}) = r$.
2. **Brute-force** $\texttt{salt} \in \{0,1\}^{128}$ with

   $\mathsf{Hash}(\texttt{message} \| \texttt{salt}) \in \mathsf{ColSpace}(\mathbf{E}_{\mathsf{ch}})$.

Our Forgery Attack

Use a (slightly) different ch so that $\tilde{\mathsf{pk}}(\mathbf{u}) = \mathbf{E}_{\mathsf{ch}} \cdot \mathcal{F}(\mathbf{u})$, with $\mathbf{E}_{\mathsf{ch}} = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix}$

Attack: Given $r < \mathsf{Ncols}(\tilde{\mathbf{E}}_{\mathsf{ch}})$:

1. Brute-force ch with $\mathsf{rank}(\tilde{\mathbf{E}}_{\mathsf{ch}}) = r$.
2. **Brute-force** $\texttt{salt} \in \{0,1\}^{128}$ with

   $\mathsf{Hash}(\texttt{message}\|\texttt{salt}) \in \mathsf{ColSpace}(\mathbf{E}_{\mathsf{ch}})$.
3. Solve for $\mathbf{u} \in \mathbb{F}_q^{n'}$, $y_i \in \mathbb{F}_q$, a system

   $$0 = \mathcal{F}(\mathbf{u}) + \mathbf{W} \cdot (1, y_1, \ldots, y_p)^t$$

   where $\mathbf{W}$ is known matrix.

Our Forgery Attack

Use a (slightly) different ch so that $\tilde{\mathsf{pk}}(\mathbf{u}) = \mathbf{E}_{\mathsf{ch}} \cdot \mathcal{F}(\mathbf{u})$, with $\mathbf{E}_{\mathsf{ch}} = \begin{bmatrix} \tilde{\mathbf{E}}_{\mathsf{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\mathsf{ch}} \end{bmatrix}$

Attack: Given $r < \mathsf{Ncols}(\tilde{\mathbf{E}}_{\mathsf{ch}})$:

1. Brute-force ch with $\mathsf{rank}(\tilde{\mathbf{E}}_{\mathsf{ch}}) = r$.
2. **Brute-force** $\mathtt{salt} \in \{0,1\}^{128}$ with

   $\mathsf{Hash}(\mathtt{message} \| \mathtt{salt}) \in \mathsf{ColSpace}(\mathbf{E}_{\mathsf{ch}})$.
3. Solve for $\mathbf{u} \in \mathbb{F}_q^{n'}$, $y_i \in \mathbb{F}_q$, a system

   $$0 = \mathcal{F}(\mathbf{u}) + \mathbf{W} \cdot (1, y_1, \ldots, y_p)^t$$

   where $\mathbf{W}$ is known matrix.
4. **Output** $\sigma = (\mathsf{ch}^{-1}(\mathbf{u}), \mathtt{salt})$

Our Forgery Attack

Use a (slightly) different ch so that $\tilde{\text{pk}}(\mathbf{u}) = \mathbf{E}_{\text{ch}} \cdot \mathcal{F}(\mathbf{u})$, with $\mathbf{E}_{\text{ch}} = \begin{bmatrix} \tilde{\mathbf{E}}_{\text{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\text{ch}} \end{bmatrix}$

Attack: Given $r < \text{Ncols}(\tilde{\mathbf{E}}_{\text{ch}})$:

1. Brute-force ch with $\text{rank}(\tilde{\mathbf{E}}_{\text{ch}}) = r$.
2. **Brute-force** $\texttt{salt} \in \{0,1\}^{128}$ with

   $\text{Hash}(\texttt{message} \| \texttt{salt}) \in \text{ColSpace}(\mathbf{E}_{\text{ch}})$.
3. Solve for $\mathbf{u} \in \mathbb{F}_q^{n'}$, $y_i \in \mathbb{F}_q$, a system

   $$0 = \mathcal{F}(\mathbf{u}) + \mathbf{W} \cdot (1, y_1, \ldots, y_p)^t$$

   where $\mathbf{W}$ is known matrix.
4. **Output** $\sigma = (\text{ch}^{-1}(\mathbf{u}), \texttt{salt})$

Solving at Step 3:

- Lift the system over $\mathbb{F}_{q^l}$ to obtain

  $$0 = \mathcal{H}(\tilde{\mathbf{u}}) + \tilde{\mathbf{W}} \cdot (1, y_1, \ldots, y_p)^t$$
- Solve using Hybrid-F4.

Our Forgery Attack

Use a (slightly) different ch so that $\tilde{\text{pk}}(\mathbf{u}) = \mathbf{E}_{\text{ch}} \cdot \mathcal{F}(\mathbf{u})$, with $\mathbf{E}_{\text{ch}} = \begin{bmatrix} \tilde{\mathbf{E}}_{\text{ch}} & & \\ & \ddots & \\ & & \tilde{\mathbf{E}}_{\text{ch}} \end{bmatrix}$

Attack: Given $r < \text{Ncols}(\tilde{\mathbf{E}}_{\text{ch}})$:

1. Brute-force ch with $\text{rank}(\tilde{\mathbf{E}}_{\text{ch}}) = r$.
2. **Brute-force** $\texttt{salt} \in \{0,1\}^{128}$ with

   $\text{Hash}(\texttt{message} \| \texttt{salt}) \in \text{ColSpace}(\mathbf{E}_{\text{ch}})$.
3. Solve for $\mathbf{u} \in \mathbb{F}_q^{n'}$, $y_i \in \mathbb{F}_q$, a system

   $$0 = \mathcal{F}(\mathbf{u}) + \mathbf{W} \cdot (1, y_1, \ldots, y_p)^t$$

   where $\mathbf{W}$ is known matrix.
4. **Output** $\sigma = (\text{ch}^{-1}(\mathbf{u}), \texttt{salt})$

Solving at Step 3:

- Lift the system over $\mathbb{F}_{q^l}$ to obtain

  $$0 = \mathcal{H}(\tilde{\mathbf{u}}) + \tilde{\mathbf{W}} \cdot (1, y_1, \ldots, y_p)^t$$
- Solve using Hybrid-F4.

Remarks:

✓ Able to exploit the structure of $\mathcal{H}$.

✗ p extra variables linear y_i.

✗ We have an extra brute-force step.

Complexity of Forgery for Level I

l	rank($\tilde{\mathbf{E}}_{ch}$)	Fraction of weak keys	Previous best	**This paper** ($\omega = 2$)
2	3	1	137	**109**
	2	$2^{-8.9}$	**97**	N.A
	1	$2^{-17.1}$	**45**	N.A
3	7	1	150	**123**
	6	$2^{-12.0}$	130	**110**
	5	$2^{-40.0}$	**112**	142*
4	13	1	167	**139**
	12	2^{-16}	156	**125**
	11	2^{-52}	145	**117**

[0] N.A = 2nd brute-force step unsuccessful. * attack dominated by the 2nd brute-force step.

Thanks.

Technology
Innovation
Institute

tii.ae

References I

[Beu25] Ward Beullens. Improved cryptanalysis of SNOVA. In *Advances in Cryptology – EUROCRYPT 2025: 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4–8, 2025, Proceedings, Part VI*, page 277–293, Berlin, Heidelberg, 2025. Springer-Verlag.

[FS13] Jean-Charles Faugère and Jules Svartz. Gröbner bases of ideals invariant under a commutative group: the non-modular case. In *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*, ISSAC '13, page 347–354, New York, NY, USA, 2013. Association for Computing Machinery.

[IA24] Yasuhiko Ikematsu and Rika Akiyama. Revisiting the security analysis of SNOVA. *Proceedings of the 11th ACM Asia Public-Key Cryptography Workshop*, 2024.

[LD24] Peigen Li and Jintai Ding. Cryptanalysis of the SNOVA signature scheme. In *International Conference on Post-Quantum Cryptography*, pages 79–91. Springer, 2024.

References II

[NTF24] Shuhei Nakamura, Yusuke Tani, and Hiroki Furue. Lifting approach against the SNOVA scheme. Cryptology ePrint Archive, Paper 2024/1374, 2024.