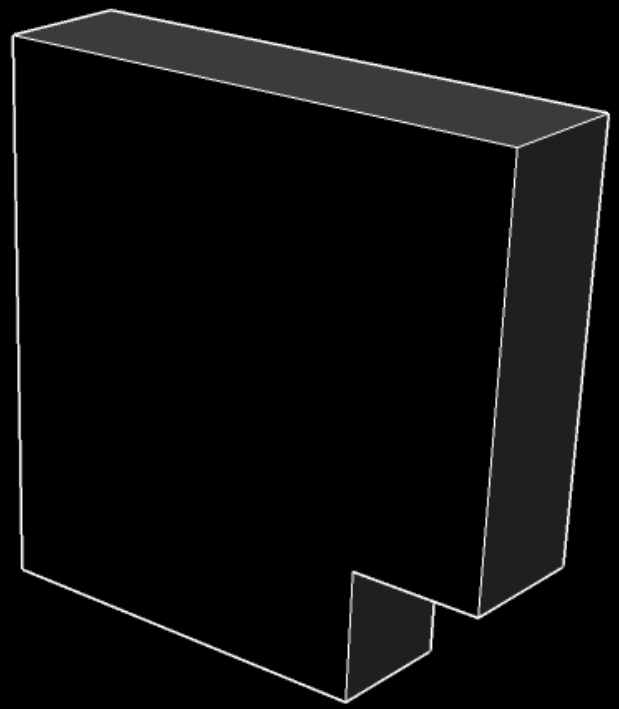
LUOV



Ward Beullens, Bart Preneel, Alan Szepieniec, Frederik Vercauteren

Overview

What is LUOV? (baby don't hurt me)

Unbalanced Oil and Vinegar (UOV) [Patarin 1997]

- Quadratic trapdoor function: $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ with $n > m$.
- Trapdoor is a factorization of $\mathcal{P}$ $\quad$ $\mathcal{F} \circ \mathcal{T}$, where $\mathcal{T}$ is linear and $\mathcal{F}$ linear in the last m variables (oil variables).
- Well understood signature scheme, fast, small signatures, but large keys. Used as building block for other MQ schemes (e.g. Rainbow).

What is LUOV? (baby don't hurt me)

Unbalanced Oil and Vinegar (UOV) [Patarin 1997]

- Quadratic trapdoor function: $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ with $n > m$.
- Trapdoor is a factorization of $\mathcal{P}$ $\quad$ $\mathcal{F} \circ \mathcal{T}$, where $\mathcal{T}$ is linear and $\mathcal{F}$ linear in the last m variables (oil variables).
- Well understood signature scheme, fast, small signatures, but large keys. Used as building block for other MQ schemes (e.g. Rainbow).

Goal of LUOV is to reduce the key sizes.
(while preserving the good properties of UOV)

What is LUOV? (baby don't hurt me)

Unbalanced Oil and Vinegar (UOV) [Patarin 1997]

- Quadratic trapdoor function: $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ with $n > m$.
- Trapdoor is a factorization of $\mathcal{P} \quad \mathcal{F} \circ \mathcal{T}$, where $\mathcal{T}$ is linear and $\mathcal{F}$ linear in the last m variables (oil variables).
- Well understood signature scheme, fast, small signatures, but large keys. Used as building block for other MQ schemes (e.g. Rainbow).

Goal of LUOV is to reduce the key sizes.

(while preserving the good properties of UOV)

- Generate SK from seed
- Generate most of PK from seed [Petzoldt]
- Field lifting

Given a UOV key pair $(\mathcal{P}, \mathcal{T})$ over $\mathbb{F}_2$, we can use it as a key pair over $\mathbb{F}_{2^r}$.

$$\underbrace{\begin{matrix} x_1^2 + x_1x_2 + x_3 + x_1x_4 + x_4x_5 + x_5 \\ x_2x_3 + x_3^2 + x_2x_6 + x_3x_4 + x_3x_5 + x_6^2 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_2 + x_5x_6 \end{matrix}}_{\mathcal{P}(\mathbf{x})} \quad \underbrace{\begin{matrix} 1 + \alpha^2 + \quad + \alpha^{30} \\ 1 + \alpha + \quad + \alpha^{31} \\ \alpha + \alpha^5 + \quad + \alpha^{31} \end{matrix}}_{\mathcal{H}(M)}$$

Given a UOV key pair $(\mathcal{P}, \mathcal{T})$ over $\mathbb{F}_2$, we can use it as a key pair over $\mathbb{F}_{2^r}$.

$$\underbrace{\begin{matrix} x_1^2 + x_1x_2 + x_3 + x_1x_4 + x_4x_5 + x_5 \\ x_2x_3 + x_3^2 + x_2x_6 + x_3x_4 + x_3x_5 + x_6^2 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_2 + x_5x_6 \end{matrix}}_{\mathcal{P}(\mathbf{x})} \quad \underbrace{\begin{matrix} 1 + \alpha^2 + \quad + \alpha^{30} \\ 1 + \alpha + \quad + \alpha^{31} \\ \alpha + \alpha^5 + \quad + \alpha^{31} \end{matrix}}_{\mathcal{H}(M)}$$

Field Lifting Assumption:

Solving a random system $\mathcal{P}(x) \quad y$ over $\mathbb{F}_{2^r}$ is as hard as solving a random system $\mathcal{P}(x) \quad y$, where $\mathcal{P}$ is defined over $\mathbf{F}_2$,

Given a UOV key pair $(\mathcal{P}, \mathcal{T})$ over $\mathbb{F}_2$, we can use it as a key pair over $\mathbb{F}_{2^r}$.

$$\underbrace{\begin{matrix} x_1^2 + x_1x_2 + x_3 + x_1x_4 + x_4x_5 + x_5 \\ x_2x_3 + x_3^2 + x_2x_6 + x_3x_4 + x_3x_5 + x_6^2 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_2 + x_5x_6 \end{matrix}}_{\mathcal{P}(\mathbf{x})} \quad \underbrace{\begin{matrix} 1 + \alpha^2 + \quad + \alpha^{30} \\ 1 + \alpha + \quad + \alpha^{31} \\ \alpha + \alpha^5 + \quad + \alpha^{31} \end{matrix}}_{\mathcal{H}(M)}$$

Field Lifting Assumption:

Solving a random system $\mathcal{P}(x) \quad y$ over $\mathbb{F}_{2^r}$ is as hard as solving a random system $\mathcal{P}(x) \quad y$, where $\mathcal{P}$ is defined over $\mathbf{F}_2$, when r is prime.

Attacks

- Key recovery attacks
  Studied since 1997
- Forgery attacks: Solve $\mathcal{P}(\mathbf{x}) \quad \mathbf{y}$ for $\mathbf{x}$.

Attacks

- Key recovery attacks
  Studied since 1997
- Forgery attacks: Solve $\mathcal{P}(\mathbf{x}) \quad \mathbf{y}$ for $\mathbf{x}$.

Subfield differential attack (Ding et al. 2019):
Pick random $\mathbf{x}_0$ and solve $\mathcal{P}(\mathbf{x}_0 + \mathbf{x}') \quad y$ for $\mathbf{x}'$ in a subfield.

Attacks

- Key recovery attacks
  Studied since 1997
- Forgery attacks: Solve $\mathcal{P}(\mathbf{x}) \quad \mathbf{y}$ for $\mathbf{x}$.

Subfield differential attack (Ding et al. 2019):
Pick random $\mathbf{x}_0$ and solve $\mathcal{P}(\mathbf{x}_0 + \mathbf{x}') \quad y$ for $\mathbf{x}'$ in a subfield.

Claimed complexity of the attack:

Parameters	Security lvl	Subfield	Complexity
LUOV-8-58-237	2	$\mathbb{F}_{2^2} \subset \mathbb{F}_{2^8}$	2^{107}
LUOV-48-43-222	2	$\mathbb{F}_{2^8} \subset \mathbb{F}_{2^{48}}$	2^{135}

- Key recovery attacks
  Studied since 1997
- Forgery attacks: Solve $\mathcal{P}(\mathbf{x}) \quad \mathbf{y}$ for $\mathbf{x}$.

Subfield differential attack (Ding et al. 2019):
Pick random $\mathbf{x}_0$ and solve $\mathcal{P}(\mathbf{x}_0 + \mathbf{x}') \quad y$ for $\mathbf{x}'$ in a subfield.

Claimed complexity of the attack:

Parameters	Security lvl	Subfield	Complexity
LUOV-8-58-237	2	$\mathbb{F}_{2^2} \subset \mathbb{F}_{2^8}$	2^{107}
LUOV-48-43-222	2	$\mathbb{F}_{2^8} \subset \mathbb{F}_{2^{48}}$	2^{135}

Solution: Choose $\mathbb{F}_{2^r}$, with r prime, such that there are no subfields to exploit. $\Rightarrow$ **No performance penalty**.

- Key recovery attacks
  Studied since 1997
- Forgery attacks: Solve $\mathcal{P}(\mathbf{x}) \quad \mathbf{y}$ for $\mathbf{x}$.

Subfield differential attack (Ding et al. 2019):
Pick random $\mathbf{x}_0$ and solve $\mathcal{P}(\mathbf{x}_0 + \mathbf{x}') \quad y$ for $\mathbf{x}'$ in a subfield.

Claimed complexity of the attack:

Parameters	Security lvl	Subfield	Complexity
LUOV-8-58-237	2	$\mathbb{F}_{2^2} \subset \mathbb{F}_{2^8}$	2^{107}
LUOV-48-43-222	2	$\mathbb{F}_{2^8} \subset \mathbb{F}_{2^{48}}$	2^{135}

Solution: Choose $\mathbb{F}_{2^r}$, with r prime, such that there are no subfields to exploit. $\Rightarrow$ **No performance penalty**.

We study some generalization of the attack in revised LUOV submission document.

Round 2 improvements

- Take smaller parameters $\Rightarrow$ more efficient

Round 2 improvements

- Take smaller parameters $\Rightarrow$ more efficient
- Add salt to message before signing
  $\Rightarrow$ Improved security against fault injection attacks and side-channel attacks.

Round 2 improvements

- Take smaller parameters $\Rightarrow$ more efficient
- Add salt to message before signing
  $\Rightarrow$ Improved security against fault injection attacks and side-channel attacks.
- Break up PRNG calls into multiple smaller calls.
  $\Rightarrow$ Speed up by parallelization, lower memory usage.

Round 2 improvements

- Take smaller parameters $\Rightarrow$ more efficient
- Add salt to message before signing
  $\Rightarrow$ Improved security against fault injection attacks and side-channel attacks.
- Break up PRNG calls into multiple smaller calls.
  $\Rightarrow$ Speed up by parallelization, lower memory usage.
- Constant time AVX2 optimized implementation.

Round 2 improvements

- Take smaller parameters $\Rightarrow$ more efficient
- Add salt to message before signing
  $\Rightarrow$ Improved security against fault injection attacks and side-channel attacks.
- Break up PRNG calls into multiple smaller calls.
  $\Rightarrow$ Speed up by parallelization, lower memory usage.
- Constant time AVX2 optimized implementation.
- Add option to use Chacha8 instead of SHAKE to expand public randomness. $\Rightarrow$ $\times 2.5$ and $\times 5.2$ faster signing and verification respectively (SL1).

Round 2.1 modifications

- Choose field extension of prime degree.

Original	New
$\mathbb{F}_{2^8}$	$\mathbb{F}_{2^7}$
$\mathbb{F}_{2^{48}}$	$\mathbb{F}_{2^{47}}$
$\mathbb{F}_{2^{64}}$	$\mathbb{F}_{2^{61}}$
$\mathbb{F}_{2^{80}}$	$\mathbb{F}_{2^{79}}$

- Aim for security level 1,3,5 instead of 2,4,5.
  $\Rightarrow$ Smaller keys and signatures and better performance.

Round 2.1 modifications

- Choose field extension of prime degree.

Original	New
$\mathbb{F}_{2^8}$	$\mathbb{F}_{2^7}$
$\mathbb{F}_{2^{48}}$	$\mathbb{F}_{2^{47}}$
$\mathbb{F}_{2^{64}}$	$\mathbb{F}_{2^{61}}$
$\mathbb{F}_{2^{80}}$	$\mathbb{F}_{2^{79}}$

- Aim for security level 1,3,5 instead of 2,4,5.
  $\Rightarrow$ Smaller keys and signatures and better performance.

Updated submission package will be online next week.

Some numbers

Key and signature sizes for SL1:

	$\lvert\text{sig}\rvert$	Δ	$\lvert\text{pk}\rvert$	Δ	$\lvert\text{sk}\rvert$
LUOV-7-57-197	239 B	-23%	11.5 KB	-5%	32B
LUOV-47-42-182	1332 B	-17%	4.7 KB	-6%	32B

[1] Requires 250 KB to store expanded PK or SK

[2] Requires 23 KB to store partial signature

Some numbers

Key and signature sizes for SL1:

	|sig|	Δ	|pk|	Δ	|sk|
LUOV-7-57-197	239 B	−23%	11.5 KB	−5%	32B
LUOV-47-42-182	1332 B	−17%	4.7 KB	−6%	32B

Performance of AVX2 constant-time implementation (SL I):

	PRG	keygen (cycles)	sign (cycles)	verify (cycles)
Standard LUOV	Keccak	1.9 M	1.4 M	1.0 M
	Chacha8	1.1M	515 K	197 K
Precompute Keys[1]	⋆		300 K	90 K
Finish signature [2]	⋆		11 K	

[1] Requires 250 KB to store expanded PK or SK

[2] Requires 23 KB to store partial signature

Conclusion (part 1)

Disadvantages:

- Public key size (11.5 KB)
- Relatively new LUOV assumption

Advantages:

- Small signatures (239 B)
- Small private key (32 B)
- Solid foundation (UOV)
- Simple arithmetic ($\mathbb{F}_{2^7}$)
- Low latency signing (11K cycles)
- No patent claims

Conclusion (part 2)

"All you need is LUOV"

John Lennon

"All you need is LUOV"
John Lennon

Questions?