LUOV

Ward Beullens, Bart Preneel, Alan Szepieniec, Frederik Vercauteren
1 Introduction

2 Modifications

3 Some numbers

4 Conclusion
What is LUOV? (baby don’t hurt me)

Unbalanced Oil and Vinegar (UOV) [Patarin 1997]

- Quadratic trapdoor function: $P : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ with $n > m$.
- Trapdoor is a factorization of $P \circ \mathcal{F}$, 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: $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ with $n > m$.
- Trapdoor is a factorization of $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$, 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)
Unbalanced Oil and Vinegar (UOV) [Patarin 1997]

- Quadratic trapdoor function: $P : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ with $n > m$.
- Trapdoor is a factorization of $P = F \circ T$, where $T$ is linear and $F$ 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
Field lifting

Given a UOV key pair \((P, T)\) over \(F_2\), we can use it as a key pair over \(F_{2^r}\).

\[
\begin{align*}
\mathcal{P}(x) &= x_1^2 + x_1x_2 + x_3 + x_1x_4 + x_4x_5 + x_5 \\
x_2x_3 + x_3^2 + x_2x_6 + x_3x_4 + x_3x_5 + x_6^2 \\
x_1x_2 + x_2x_3 + x_3x_4 + x_2 + x_5x_6
\end{align*}
\]

\[
\begin{align*}
1 + \alpha^2 &+ \alpha^{30} \\
1 + \alpha &+ \alpha^{31} \\
\alpha + \alpha^5 &+ \alpha^{31}
\end{align*}
\]
Field lifting

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

\[
\begin{align*}
\mathcal{P}(x) &= x_1^2 + x_1x_2 + x_3 + x_1x_4 + x_4x_5 + x_5 \\
&\quad + x_2x_3 + x_3^2 + x_2x_6 + x_3x_4 + x_3x_5 + x_6^2 \\
&\quad + x_1x_2 + x_2x_3 + x_3x_4 + x_2 + x_5x_6 \\
\mathcal{H}(M) &= 1 + \alpha^2 + \cdots + \alpha^{30} \\
&\quad + 1 + \alpha + \cdots + \alpha^{31} \\
&\quad + \alpha + \alpha^5 + \cdots + \alpha^{31}
\end{align*}
\]

Field Lifting Assumption:
Solving a random system \(\mathcal{P}(x) \cdot y\) over \(\mathbb{F}_{2^r}\) is as hard as solving a random system \(\mathcal{P}(x) \cdot y\), where \(\mathcal{P}\) is defined over \(\mathbb{F}_2\).
Given a UOV key pair \((P, T)\) over \(\mathbb{F}_2\), we can use it as a key pair over \(\mathbb{F}_{2^r}\).

\[
\begin{align*}
    x_1^2 + x_1 x_2 + x_3 + x_1 x_4 + x_4 x_5 + x_5 \\
    x_2 x_3 + x_3^2 + x_2 x_6 + x_3 x_4 + x_3 x_5 + x_6^2 \\
    x_1 x_2 + x_2 x_3 + x_3 x_4 + x_2 + x_5 x_6
\end{align*}
\]

\[
\begin{align*}
    1 + \alpha^2 + &+ \alpha^{30} \\
    1 + \alpha + &+ \alpha^{31} \\
    \alpha + \alpha^5 + &+ \alpha^{31}
\end{align*}
\]

\[\mathcal{P}(x)\text{ }\mathcal{H}(M)\]

**Field Lifting Assumption:**
Solving a random system \(\mathcal{P}(x)\ y\) over \(\mathbb{F}_{2^r}\) is as hard as solving a random system \(\mathcal{P}(x)\ y\), where \(\mathcal{P}\) is defined over \(\mathbb{F}_2\), when \(r\) is prime.
Subfield differential attack (Ding et al. 2019):
Pick random $x_0$ and solve $P(x_0 + x_0) = y$ for $x_0$ in a subfield.

Claimed complexity of the attack:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Security lvl</th>
<th>Subfield</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUOV-8-58-237</td>
<td>2</td>
<td>$F_2^{22} \subset F_2^{28}$</td>
<td>$2^{107}$</td>
</tr>
<tr>
<td>LUOV-48-43-222</td>
<td>2</td>
<td>$F_2^{28} \subset F_2^{248}$</td>
<td>$2^{135}$</td>
</tr>
</tbody>
</table>

Solution: Choose $F_2^r$, 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.
• Key recovery attacks
  Studied since 1997
• Forgery attacks: Solve $P(x) \mod y$ for $x$.

Subfield differential attack (Ding et al. 2019):
Pick random $x_0$ and solve $P(x_0 + x') \mod y$ for $x'$ in a subfield.
• Key recovery attacks
  Studied since 1997
• Forgery attacks: Solve $\mathcal{P}(x) \equiv y$ for $x$.

Subfield differential attack (Ding et al. 2019):
Pick random $x_0$ and solve $\mathcal{P}(x_0 + x') \equiv y$ for $x'$ in a subfield.

Claimed complexity of the attack:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Security lvl</th>
<th>Subfield</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUOV-8-58-237</td>
<td>2</td>
<td>$\mathbb{F}<em>{2^{2}} \subset \mathbb{F}</em>{2^{8}}$</td>
<td>$2^{107}$</td>
</tr>
<tr>
<td>LUOV-48-43-222</td>
<td>2</td>
<td>$\mathbb{F}<em>{2^{8}} \subset \mathbb{F}</em>{2^{48}}$</td>
<td>$2^{135}$</td>
</tr>
</tbody>
</table>
• Key recovery attacks
  Studied since 1997
• Forgery attacks: Solve $P(x) \ y$ for $x$.

Subfield differential attack (Ding et al. 2019):
Pick random $x_0$ and solve $P(x_0 + x') \ y$ for $x'$ in a subfield.

Claimed complexity of the attack:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Security lvl</th>
<th>Subfield</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUOV-8-58-237</td>
<td>2</td>
<td>$F_{2^2} \subset F_{2^8}$</td>
<td>$2^{107}$</td>
</tr>
<tr>
<td>LUOV-48-43-222</td>
<td>2</td>
<td>$F_{2^8} \subset F_{2^{48}}$</td>
<td>$2^{135}$</td>
</tr>
</tbody>
</table>

Solution: Choose $F_{2^r}$, with $r$ prime, such that there are no subfields to exploit. ⇒ **No performance penalty.**
Attacks

- Key recovery attacks
  Studied since 1997
- Forgery attacks: Solve $P(x) \cdot y$ for $x$.

Subfield differential attack (Ding et al. 2019):
Pick random $x_0$ and solve $P(x_0 + x') \cdot y$ for $x'$ in a subfield.

Claimed complexity of the attack:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Security lvl</th>
<th>Subfield</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUOV-8-58-237</td>
<td>2</td>
<td>$F_{22} \subset F_{28}$</td>
<td>$2^{107}$</td>
</tr>
<tr>
<td>LUOV-48-43-222</td>
<td>2</td>
<td>$F_{28} \subset F_{248}$</td>
<td>$2^{135}$</td>
</tr>
</tbody>
</table>

Solution: Choose $F_{2^r}$, with $r$ prime, such that there are no subfields to exploit. ⇒ **No performance penalty.**

We study some generalization of the attack in revised LUOV submission document.
Round 2 improvements

• Take smaller parameters ⇒ more efficient
Round 2 improvements

- Take smaller parameters ⇒ more efficient
- Add salt to message before signing
  ⇒ Improved security against fault injection attacks and side-channel attacks.
Round 2 improvements

• Take smaller parameters ⇒ more efficient
• Add salt to message before signing
  ⇒ Improved security against fault injection attacks and side-channel attacks.
• Break up PRNG calls into multiple smaller calls.
  ⇒ Speed up by parallelization, lower memory usage.
• Take smaller parameters ⇒ more efficient
• Add salt to message before signing
  ⇒ Improved security against fault injection attacks and side-channel attacks.
• Break up PRNG calls into multiple smaller calls.
  ⇒ Speed up by parallelization, lower memory usage.
• Constant time AVX2 optimized implementation.
Round 2 improvements

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

<table>
<thead>
<tr>
<th>Original</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{2^8}$</td>
<td>$F_{2^7}$</td>
</tr>
<tr>
<td>$F_{2^{48}}$</td>
<td>$F_{2^{47}}$</td>
</tr>
<tr>
<td>$F_{2^{64}}$</td>
<td>$F_{2^{61}}$</td>
</tr>
<tr>
<td>$F_{2^{80}}$</td>
<td>$F_{2^{79}}$</td>
</tr>
</tbody>
</table>

• Aim for security level 1,3,5 instead of 2,4,5.
  ⇒ Smaller keys and signatures and better performance.
Round 2.1 modifications

- Choose field extension of prime degree.

<table>
<thead>
<tr>
<th>Original</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{2^8}$</td>
<td>$F_{2^7}$</td>
</tr>
<tr>
<td>$F_{2^{48}}$</td>
<td>$F_{2^{47}}$</td>
</tr>
<tr>
<td>$F_{2^{64}}$</td>
<td>$F_{2^{61}}$</td>
</tr>
<tr>
<td>$F_{2^{80}}$</td>
<td>$F_{2^{79}}$</td>
</tr>
</tbody>
</table>

- Aim for security level 1, 3, 5 instead of 2, 4, 5.

⇒ Smaller keys and signatures and better performance.

Updated submission package will be online next week.
Some numbers

Key and signature sizes for SL1:

<table>
<thead>
<tr>
<th></th>
<th>sig</th>
<th>Δ</th>
<th>pk</th>
<th>Δ</th>
<th>sk</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUOV-7-57-197</td>
<td>239 B</td>
<td>−23%</td>
<td>11.5 KB</td>
<td>−5%</td>
<td>32B</td>
</tr>
<tr>
<td>LUOV-47-42-182</td>
<td>1332 B</td>
<td>−17%</td>
<td>4.7 KB</td>
<td>−6%</td>
<td>32B</td>
</tr>
</tbody>
</table>

1Requires 250 KB to store expanded PK or SK
2Requires 23 KB to store partial signature
Some numbers

Key and signature sizes for SL1:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>sig</strong></td>
<td><strong>Δ</strong></td>
<td><strong>pk</strong></td>
<td><strong>Δ</strong></td>
</tr>
<tr>
<td>LUOV-7-57-197</td>
<td>239 B</td>
<td>−23%</td>
<td>11.5 KB</td>
<td>−5%</td>
</tr>
<tr>
<td>LUOV-47-42-182</td>
<td>1332 B</td>
<td>−17%</td>
<td>4.7 KB</td>
<td>−6%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>PRG</th>
<th>keygen (cycles)</th>
<th>sign (cycles)</th>
<th>verify (cycles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard LUOV</td>
<td>Keccak</td>
<td>1.9 M</td>
<td>1.4 M</td>
<td>1.0 M</td>
</tr>
<tr>
<td></td>
<td>Chacha8</td>
<td>1.1M</td>
<td>515 K</td>
<td>197 K</td>
</tr>
<tr>
<td>Precompute Keys(^1)</td>
<td>*</td>
<td></td>
<td>300 K</td>
<td>90 K</td>
</tr>
<tr>
<td>Finish signature (^2)</td>
<td>*</td>
<td></td>
<td>11 K</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) 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
“All you need is LUOV”

John Lennon
"All you need is LUOV"
John Lennon

Questions?