On the Side-Channel Resistance of UOV
Survey of Physical Attacks and Recent Developments

Thomas Aulbach¹
NIST PQC Seminars, 07.07.2023

¹Universität Regensburg, Regensburg, Germany
1. UOV from Two Perspectives

2. Fault Attacks
   - Skip Random Sampling of Vinegar Variables [SK20]
   - Bit-Flip in Central Map [FKN+22]

3. Side Channel Attacks
   - Horizontal SCA on Linear Transformation [PSK+18]
   - Template Attack on Evaluation of Vinegar Variables [ACK+23]

4. Takeaways
UOV from Two Perspectives
UOV stands out, since

- it is a comparably old scheme with 25 years of cryptanalysis
- many current (and past) multivariate signature schemes are modifications of it

1https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lkGPk
UOV stands out, since

- it is a comparably old scheme with 25 years of cryptanalysis
- many current (and past) multivariate signature schemes are modifications of it

NIST would like submissions for signature schemes that:\footnote{https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lkGPk}
UOV stands out, since

- it is a comparably old scheme with 25 years of cryptanalysis
- many current (and past) multivariate signature schemes are modifications of it

NIST would like submissions for signature schemes that:\(^1\)

- ‘are not based on structured lattices’  ✓

---

\(^1\)https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lkgPk
UOV stands out, since

- it is a comparably old scheme with 25 years of cryptanalysis
- many current (and past) multivariate signature schemes are modifications of it

NIST would like submissions for signature schemes that:\(^1\)

- ‘are not based on structured lattices’ ✓
- ‘have short signatures’ ✓

\(^1\)https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lkGPk
UOV stands out, since

- it is a comparably old scheme with 25 years of cryptanalysis
- many current (and past) multivariate signature schemes are modifications of it

NIST would like submissions for signature schemes that:\footnote{https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lkGPk}

- ‘are not based on structured lattices’ ✓
- ‘have short signatures’ ✓
- ‘and fast verification’ ✓
UOV stands out, since

- it is a comparably old scheme with 25 years of cryptanalysis
- many current (and past) multivariate signature schemes are modifications of it

NIST would like submissions for signature schemes that:\footnote{https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lKGPk}

- ‘are not based on structured lattices’ ✓
- ‘have short signatures’ ✓
- ‘and fast verification’ ✓
- ‘e.g., UOV’ ✓
## Comparison with Dilithium

### Oil and Vinegar: Modern Parameters and Implementations

Key sizes and performance data

<table>
<thead>
<tr>
<th>Signature Scheme</th>
<th>public key</th>
<th>secret key</th>
<th>signature</th>
<th>KeyGen</th>
<th>Sign Cycles</th>
<th>Verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>ov-lp</td>
<td>278 432</td>
<td>237 912</td>
<td>128</td>
<td>2 903 434</td>
<td>105 324</td>
<td>90 336</td>
</tr>
<tr>
<td>ov-lp-pkc</td>
<td>43 576</td>
<td>237 912</td>
<td>128</td>
<td>2 858 724</td>
<td>105 324</td>
<td>224 006</td>
</tr>
<tr>
<td>ov-lp-pkc-skc</td>
<td>43 576</td>
<td>64</td>
<td>128</td>
<td>2 848 774</td>
<td>1 876 442</td>
<td>224 006</td>
</tr>
<tr>
<td>Dilithium2</td>
<td>1 312</td>
<td>2 544</td>
<td>2 420</td>
<td>124 031</td>
<td>333 013</td>
<td>118 412</td>
</tr>
</tbody>
</table>

---

Signatures from multivariate quadratic equations:

- Key objects are multivariate quadratic maps $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$
Signatures from multivariate quadratic equations:

- Key objects are multivariate quadratic maps $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$
- $\mathcal{P}$ consists of $m$ homogeneous quadratic polynomials

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha^{(k)}_{i,j} x_i x_j, \text{ where } x = (x_1, \ldots, x_n)^\top \in \mathbb{F}_q^n$$
Signatures from multivariate quadratic equations:

- Key objects are multivariate quadratic maps $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$
- $\mathcal{P}$ consists of $m$ homogeneous quadratic polynomials

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{i,j}^{(k)} x_i x_j, \text{ where } x = (x_1, \ldots, x_n)^\top \in \mathbb{F}_q^n$$

- **Signing** $d$ in a nutshell: For $t = H(d) \in \mathbb{F}_q^m$, find $s \in \mathbb{F}_q^n$, such that $\mathcal{P}(s) = t$
Signatures from multivariate quadratic equations:

- Key objects are multivariate quadratic maps $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$
- $\mathcal{P}$ consists of $m$ homogeneous quadratic polynomials

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{i,j}^{(k)} x_i x_j, \text{ where } x = (x_1, \ldots, x_n)^\top \in \mathbb{F}_q^n$$

- **Signing** $d$ in a nutshell: For $t = H(d) \in \mathbb{F}_q^m$, find $s \in \mathbb{F}_q^n$, such that $\mathcal{P}(s) = t$
  - In general this is really difficult
Signatures from multivariate quadratic equations:

• Key objects are multivariate quadratic maps $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$

• $\mathcal{P}$ consists of $m$ homogeneous quadratic polynomials

$$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{i,j}^{(k)} x_i x_j, \text{ where } x = (x_1, \ldots, x_n)^\top \in \mathbb{F}_q^n$$

• **Signing** $d$ in a nutshell: For $t = H(d) \in \mathbb{F}_q^m$, find $s \in \mathbb{F}_q^n$, such that $\mathcal{P}(s) = t$

  • In general this is really difficult
  • Include a trapdoor that can only be used with the secret key
Signatures from multivariate quadratic equations:

- Key objects are multivariate quadratic maps $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$
- $\mathcal{P}$ consists of $m$ homogeneous quadratic polynomials
  $$p_k(x) = \sum_{1 \leq i \leq j \leq n} \alpha_{i,j}^{(k)} x_i x_j, \text{ where } x = (x_1, \ldots, x_n)^\top \in \mathbb{F}_q^n$$

- **Signing** $d$ in a nutshell: For $t = H(d) \in \mathbb{F}_q^m$, find $s \in \mathbb{F}_q^n$, such that $\mathcal{P}(s) = t$
  - In general this is really difficult
  - Include a trapdoor that can only be used with the secret key

- **Verify** if $\mathcal{P}(s) = t$ really holds
Two Descriptions of UOV in the Literature (1/2)

**UOV with hidden central map $\mathcal{F}$**

- $\mathcal{P} = \mathcal{F} \circ T$, where $\mathcal{F}$ is structured and easy to invert and $T$ is a linear transformation

\begin{align*}
\mathcal{F} &\text{ consists of } m \text{ homogeneous quadratic polynomials } f_k(x) = \\
&\sum_{1 \leq i \leq j \leq v} \alpha(k) i, j x_i x_j + \\
&\sum_{1 \leq i \leq v < j \leq n} \alpha(k) i, j x_i x_j,
\end{align*}

where $x = (x_1, \ldots, x_n) \in \mathbb{F}^n_q$
Two Descriptions of UOV in the Literature (1/2)

UOV with hidden central map \( \mathcal{F} \)

- \( \mathcal{P} = \mathcal{F} \circ T \), where \( \mathcal{F} \) is structured and easy to invert and \( T \) is a linear transformation
- \( \mathcal{F} \) consists of \( m \) homogeneous quadratic polynomials

\[
f_k(x) = \sum_{1 \leq i \leq j \leq v} \alpha_{i,j}^{(k)} x_i x_j + \sum_{1 \leq i < j \leq n} \alpha_{i,j}^{(k)} x_i x_j, \text{ where } x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n
\]
Two Descriptions of UOV in the Literature (1/2)

**UOV with hidden central map $\mathcal{F}$**

- $\mathcal{P} = \mathcal{F} \circ T$, where $\mathcal{F}$ is structured and easy to invert and $T$ is a linear transformation
- $\mathcal{F}$ consists of $m$ homogeneous quadratic polynomials
  $$f_k(x) = \sum_{1 \leq i \leq j \leq v} \alpha^{(k)}_{i,j} x_i x_j + \sum_{1 \leq i \leq v < j \leq n} \alpha^{(k)}_{i,j} x_i x_j, \text{ where } x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$$
- Sort coefficients to matrices $F^{(k)}$ such that $f_k(x) = x^\top F^{(k)} x$
UOV with hidden central map $\mathcal{F}$

- $\mathcal{P} = \mathcal{F} \circ T$, where $\mathcal{F}$ is structured and easy to invert and $T$ is a linear transformation
- $\mathcal{F}$ consists of $m$ homogeneous quadratic polynomials

$$f_k(x) = \sum_{1 \leq i \leq j \leq v} \alpha_{i,j}^{(k)} x_i x_j + \sum_{1 \leq i \leq v < j \leq n} \alpha_{i,j}^{(k)} x_i x_j, \text{ where } x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$$

- Sort coefficients to matrices $F^{(k)}$ such that $f_k(x) = x^\top F^{(k)} x$

$$\begin{pmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_v \\ y_1 \\ \vdots \\ y_m \end{pmatrix}^\top \begin{pmatrix} \alpha_{1,1}^{(k)} & \ldots & \alpha_{1,v}^{(k)} & \alpha_{1,v+1}^{(k)} & \ldots & \alpha_{1,n}^{(k)} \\ 0 & \ddots & 0 & \ddots & \ldots & \cdot \\ 0 & 0 & \alpha_{v,v}^{(k)} & \alpha_{v,v+1}^{(k)} & \ldots & \alpha_{v,n}^{(k)} \\ 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ldots & 0 & 0 & \ldots & 0 \end{pmatrix} \begin{pmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_v \\ y_1 \\ \vdots \\ y_m \end{pmatrix} = l^{(k)}_1 y_1 + \ldots + l^{(k)}_m y_m + c^{(k)}$$
Two Descriptions of UOV in the Literature (1/2)

### UOV with hidden central map $\mathcal{F}$

- $\mathcal{P} = \mathcal{F} \circ T$, where $\mathcal{F}$ is structured and easy to invert and $T$ is a linear transformation
- $\mathcal{F}$ consists of $m$ homogeneous quadratic polynomials

$$f_k(x) = \sum_{1 \leq i \leq j \leq v} \alpha_{i,j}^{(k)} x_i x_j + \sum_{1 \leq v < j \leq n} \alpha_{i,j}^{(k)} x_i x_j,$$

where $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$

- Sort coefficients to matrices $F^{(k)}$ such that $f_k(x) = x^\top F^{(k)} x$

- Fix and insert vinegar variables $\tilde{v}_i$ to get $m$ linear equations in $m$ oil variables

$$\begin{bmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_v \\ y_1 \\ \vdots \\ y_m \end{bmatrix}^\top \begin{bmatrix} \alpha_{1,1}^{(k)} & \cdots & \alpha_{1,v}^{(k)} & \alpha_{1,v+1}^{(k)} & \cdots & \alpha_{1,n}^{(k)} \\ 0 & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \alpha_{v,v}^{(k)} & \alpha_{v,v+1}^{(k)} & \cdots & \alpha_{v,n}^{(k)} \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_v \\ y_1 \\ \vdots \\ y_m \end{bmatrix} = l_1^{(k)} y_1 + \cdots + l_m^{(k)} y_m + c^{(k)}$$
Two Descriptions of UOV in the Literature (1/2)

UOV with hidden central map \( \mathcal{F} \)

\[ \begin{align*}
\text{\cdot Compute between } pk = \mathcal{P} \text{ and } sk = (\mathcal{F}, T) \text{ with } \\
p^{(k)} &= T^\top F^{(k)} T
\end{align*} \]
Two Descriptions of UOV in the Literature (1/2)

UOV with hidden central map $\mathcal{F}$

- Compute between $pk = \mathcal{P}$ and $sk = (\mathcal{F}, T)$ with
  \[
  p^{(k)} = T^\top F^{(k)} T
  \]

- Visualization of signing $t$

\[
\begin{align*}
  t &= \left( \begin{array}{c}
  t_1 \\
  \vdots \\
  t_m
  \end{array} \right) \\
  \mathcal{F}^{-1} \quad \rightarrow \\
  \left( \begin{array}{c}
  v_1 \\
  \vdots \\
  v_{n-m} \\
  y_1 \\
  \vdots \\
  y_m
  \end{array} \right) &= \left( \begin{array}{c}
  v \\
  y
  \end{array} \right) \\
  T^{-1} \quad \rightarrow \\
  \left( \begin{array}{c}
  s_1 \\
  s_2
  \end{array} \right) &= \left( \begin{array}{c}
  v + T_1 \cdot y \\
  y
  \end{array} \right)
\end{align*}
\]
Two Descriptions of UOV in the Literature (1/2)

UOV with hidden central map $\mathcal{F}$

- Compute between $pk = \mathcal{P}$ and $sk = (\mathcal{F}, T)$ with
  $$p^{(k)} = T^T F^{(k)} T$$

- Visualization of signing $t$

\[
\mathbf{t} = \begin{pmatrix} t_1 \\ \vdots \\ t_m \end{pmatrix} \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} v_1 \\ \vdots \\ v_{n-m} \\ y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{y} \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{v} + T_1 \cdot \mathbf{y} \\ \mathbf{y} \end{pmatrix}
\]

- $T$ has block matrix structure
  $$T = \begin{pmatrix} I_v & T_1 \\ 0 & I_m \end{pmatrix}$$
Two Descriptions of UOV in the Literature (2/2)

UOV with secret oil space

• Define $\mathcal{P}$ such that it vanishes on secret linear oil space $O \subset \mathbb{F}_q^n$ of dimension $m$, i.e.

$$\mathcal{P}(o) = 0 \text{ for all } o \in O$$
UOV with secret oil space

- Define \( P \) such that it vanishes on secret linear oil space \( O \subset \mathbb{F}_q^n \) of dimension \( m \), i.e.
  \[ P(o) = 0 \quad \text{for all} \quad o \in O \]

- The map \( P'(x, y) := P(x + y) - P(x) - P(y) \) is bilinear and symmetric.
Two Descriptions of UOV in the Literature (2/2)

**UOV with secret oil space**

- Define $P$ such that it vanishes on secret linear oil space $O \subset \mathbb{F}^n_q$ of dimension $m$, i.e.
  $$P(o) = 0 \text{ for all } o \in O$$

- The map $P'(x, y) := P(x + y) - P(x) - P(y)$ is bilinear and symmetric

**Signing strategy:**

- Generate random $v \in \mathbb{F}^n_q$
UOV with secret oil space

- Define $\mathcal{P}$ such that it vanishes on secret linear oil space $O \subseteq \mathbb{F}_q^n$ of dimension $m$, i.e.,

$$\mathcal{P}(o) = 0 \text{ for all } o \in O$$

- The map $\mathcal{P}'(x, y) := \mathcal{P}(x + y) - \mathcal{P}(x) - \mathcal{P}(y)$ is bilinear and symmetric

Signing strategy:

- Generate random $v \in \mathbb{F}_q^n$
- Solve $\mathcal{P}(v + o) = \mathcal{P}(v) + \mathcal{P}(o) + \mathcal{P}'(v, o) = t$ for $o \in O$. 
Two Descriptions of UOV in the Literature (2/2)

UOV with secret oil space

- Define $\mathcal{P}$ such that it vanishes on secret linear oil space $O \subset \mathbb{F}_q^n$ of dimension $m$, i.e.
  
  $$\mathcal{P}(o) = 0 \text{ for all } o \in O$$

- The map $\mathcal{P}'(x, y) := \mathcal{P}(x + y) - \mathcal{P}(x) - \mathcal{P}(y)$ is bilinear and symmetric

Signing strategy:

- Generate random $v \in \mathbb{F}_q^n$
- Solve $\mathcal{P}(v + o) = \mathcal{P}(v) + \mathcal{P}(o) + \mathcal{P}'(v, o) = t$ for $o \in O$.
  
  $\rightarrow$ Computing $\mathcal{P}(v)$ implies the insertion of the vinegar variables into the quadratic map
UOV with secret oil space

- Define $\mathcal{P}$ such that it vanishes on secret linear oil space $O \subseteq \mathbb{F}_q^n$ of dimension $m$, i.e.
  \[ \mathcal{P}(o) = 0 \text{ for all } o \in O \]
- The map $\mathcal{P}'(x, y) := \mathcal{P}(x + y) - \mathcal{P}(x) - \mathcal{P}(y)$ is bilinear and symmetric

Signing strategy:

- Generate random $v \in \mathbb{F}_q^n$
- Solve $\mathcal{P}(v + o) = \mathcal{P}(v) + \mathcal{P}(o) + \mathcal{P}'(v, o) = t$ for $o \in O$.
  - Computing $\mathcal{P}(v)$ implies the insertion of the vinegar variables into the quadratic map
  - Solving $\mathcal{P}'(v, o) = t - \mathcal{P}(v)$ means solving a system with $m$ variables in $m$ equations
UOV with secret oil space

- Define $P$ such that it vanishes on secret linear oil space $O \subset \mathbb{F}_q^n$ of dimension $m$, i.e.
  $$P(o) = 0 \text{ for all } o \in O$$

- The map $P'(x, y) := P(x + y) - P(x) - P(y)$ is bilinear and symmetric

Signing strategy:

- Generate random $v \in \mathbb{F}_q^n$
- Solve $P(v + o) = P(v) + P(o) + P'(v, o) = t$ for $o \in O$.
  \rightarrow \text{Computing } P(v) \text{ implies the insertion of the vinegar variables into the quadratic map}
  \rightarrow \text{Solving } P'(v, o) = t - P(v) \text{ means solving a system with } m \text{ variables in } m \text{ equations}

- The vector $s = v + o$ forms a valid signature
Fault Attacks
Skip Random Sampling of Vinegar Variables

Main idea

• Skip the random sampling of vinegar values (already discussed in [HTS11]\(^3\) and [KL19]\(^4\))

\[
\begin{align*}
t & \xrightarrow{\mathcal{F}^{-1}} (v, y) \xrightarrow{T^{-1}} (s_1, s_2) = (v + T_1 \cdot y, y)
\end{align*}
\]

\(^3\)Hashimoto, Y., Takagi, T., and Sakurai, K.: General Fault Attacks on Multivariate Public Key Cryptosystems. PQCrypto 2011

Skip Random Sampling of Vinegar Variables

Main idea

- Skip the random sampling of vinegar values (already discussed in [HTS11] and [KL19])

\[
t \xrightarrow{\mathcal{F}^{-1}} (v, y) \xrightarrow{T^{-1}} (s_1, s_2) = (v + T_1 \cdot y, y)
\]

- Solution to $\mathcal{F}^{-1}$ are the randomly generated vinegar values $v = (v_1, \ldots, v_{n-m})^T$ and the computed oil variables $y = (y_1, \ldots, y_m)^T$

---

3 Hashimoto, Y., Takagi, T., and Sakurai, K.: General Fault Attacks on Multivariate Public Key Cryptosystems. PQCrypto 2011

Skip Random Sampling of Vinegar Variables

\[ t \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} v \\ y \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} v + T_1 \cdot y \\ y \end{pmatrix} \]

\[ t^{(i)} \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} v \\ y^{(i)} \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} s_1^{(i)} \\ s_2^{(i)} \end{pmatrix} = \begin{pmatrix} v + T_1 \cdot y^{(i)} \\ y^{(i)} \end{pmatrix} \]
Skip random sampling enforces reuse of $v$

\[ t \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} v \\ y \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} v + T_1 \cdot y \\ y \end{pmatrix} \]

\[ t^{(i)} \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} v \\ y^{(i)} \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} s_1^{(i)} \\ s_2^{(i)} \end{pmatrix} = \begin{pmatrix} v + T_1 \cdot y^{(i)} \\ y^{(i)} \end{pmatrix} \]
Skip Random Sampling of Vinegar Variables

Skip random sampling enforces reuse of \( \mathbf{v} \)

We have

\[
\begin{pmatrix}
\mathbf{s}_1 \\
\mathbf{s}_2
\end{pmatrix}
= 
\begin{pmatrix}
\mathbf{v} + \mathbf{T}_1 \cdot \mathbf{y} \\
\mathbf{y}
\end{pmatrix}
\]

\[
\begin{pmatrix}
\mathbf{s}_1^{(i)} \\
\mathbf{s}_2^{(i)}
\end{pmatrix}
= 
\begin{pmatrix}
\mathbf{v} + \mathbf{T}_1 \cdot \mathbf{y}^{(i)} \\
\mathbf{y}^{(i)}
\end{pmatrix}
\]

Repeat \( m \) times to solve for \( \mathbf{T}_1 \) (requires \( m \) faulted signatures)
Skip Random Sampling of Vinegar Variables

\[
\begin{align*}
\mathbf{t} & \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} \mathbf{v} \\ \mathbf{y} \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{v} + T_1 \cdot \mathbf{y} \\ \mathbf{y} \end{pmatrix} \\
\mathbf{t}^{(i)} & \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} \mathbf{v} \\ \mathbf{y}^{(i)} \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} \mathbf{s}_1^{(i)} \\ \mathbf{s}_2^{(i)} \end{pmatrix} = \begin{pmatrix} \mathbf{v} + T_1 \cdot \mathbf{y}^{(i)} \\ \mathbf{y}^{(i)} \end{pmatrix}
\end{align*}
\]

- Skip random sampling enforces reuse of \( \mathbf{v} \)
- We have \( \begin{pmatrix} \mathbf{s}_1^{(i)} \\ \mathbf{s}_2^{(i)} \end{pmatrix} - \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} = \begin{pmatrix} T_1 \cdot (\mathbf{y}^{(i)} - \mathbf{y}) \\ (\mathbf{y}^{(i)} - \mathbf{y}) \end{pmatrix} \)
- Repeat \( m \) times to solve for \( T_1 \) (requires \( m \) faulted signatures)
Reduce Number of Needed Faulted Signatures

In fact, one can do even better

- The vector

\[
\begin{pmatrix}
    s_1^{(i)} \\
    s_2^{(i)}
\end{pmatrix}
- \begin{pmatrix}
    s_1 \\
    s_2
\end{pmatrix}
= \begin{pmatrix}
    T_1 \cdot (y^{(i)} - y) \\
    (y^{(i)} - y)
\end{pmatrix}
\]

represents an oil vector, i.e.

\[
\mathcal{P} \left( \begin{pmatrix}
    T_1 \cdot (y^{(i)} - y) \\
    (y^{(i)} - y)
\end{pmatrix} \right) = \mathcal{F} \left( \begin{pmatrix}
    I_v & T_1 \\
    0 & I_m
\end{pmatrix} \right) \left( \begin{pmatrix}
    T_1 \cdot (y^{(i)} - y) \\
    (y^{(i)} - y)
\end{pmatrix} \right) = \mathcal{F} \left( \begin{pmatrix}
    0 \\
    (y^{(i)} - y)
\end{pmatrix} \right) = 0
\]

- This is easy to recognize in the oil space description, since

\[
s^{(i)} - s = (v + o^{(i)}) - (v + o) = o^{(i)} - o \in O
\]

- One oil vector enables key recovery in polynomial time → next slide
Reduce Number of Needed Faulted Signatures

In fact, one can do even better

- The vector \( \begin{pmatrix} s_1^{(i)} \\ s_2^{(i)} \end{pmatrix} - \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} T_1 \cdot (y^{(i)} - y) \\ (y^{(i)} - y) \end{pmatrix} \) represents an oil vector, i.e.

  \[
  \mathcal{P} \left( \begin{pmatrix} T_1 \cdot (y^{(i)} - y) \\ (y^{(i)} - y) \end{pmatrix} \right) = \mathcal{F} \left( \begin{pmatrix} l_v & T_1 \\ 0 & l_m \end{pmatrix} \right) \begin{pmatrix} T_1 \cdot (y^{(i)} - y) \\ (y^{(i)} - y) \end{pmatrix} = \mathcal{F} \left( \begin{pmatrix} 0 \\ (y^{(i)} - y) \end{pmatrix} \right) = 0
  \]

- This is easy to recognize in the oil space description, since

  \[
  s^{(i)} - s = (v + o^{(i)}) - (v + o) = o^{(i)} - o \in O
  \]
Reduce Number of Needed Faulted Signatures

In fact, one can do even better

- The vector \( \begin{pmatrix} s_1^{(i)} \\ s_2^{(i)} \end{pmatrix} - \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} T_1 \cdot (y^{(i)} - y) \\ (y^{(i)} - y) \end{pmatrix} \) represents an oil vector, i.e.

\[
\mathcal{P} \left( \begin{pmatrix} T_1 \cdot (y^{(i)} - y) \\ (y^{(i)} - y) \end{pmatrix} \right) = \mathcal{F} \left( \begin{pmatrix} l_v & T_1 \\ 0 & l_m \end{pmatrix} \right) \begin{pmatrix} T_1 \cdot (y^{(i)} - y) \\ (y^{(i)} - y) \end{pmatrix} = \mathcal{F} \left( \begin{pmatrix} 0 \\ (y^{(i)} - y) \end{pmatrix} \right) = 0
\]

- This is easy to recognize in the oil space description, since

\[
s^{(i)} - s = (v + o^{(i)}) - (v + o) = o^{(i)} - o \in O
\]

- One oil vector enables key recovery in polynomial time \( \rightarrow \) next slide
Knowledge of an oil vector dramatically simplifies algebraic key recovery attacks

- For two oil vectors $o_1, o_2$ it holds

\[ P'(o_1, o_2) = P(o_1 + o_2) - P(o_1) - P(o_2) = 0 \in \mathbb{F}_q^m \]
Algebraic Attack

Knowledge of an oil vector dramatically simplifies algebraic key recovery attacks

- For two oil vectors \( o_1, o_2 \) it holds

\[
P'(o_1, o_2) = P(o_1 + o_2) - P(o_1) - P(o_2) = 0 \in \mathbb{F}_q^m
\]

→ If \( o_1 \) and \( o_2 \) are unknown, this is a quadratic system that is hard to solve

\[
\rightarrow
\]

Details can be found in [ACK+23] Aulbach, T., Campos, F., Krämer, J., Samardjiska, S., and Stöttinger, M.: Separating Oil and Vinegar with a Single Trace: Side-Channel Assisted Kipnis-Shamir Attack on UOV. IACR TCHES 2023
Algebraic Attack

Knowledge of an oil vector dramatically simplifies algebraic key recovery attacks

• For two oil vectors $o_1, o_2$ it holds

$$\mathcal{P}'(o_1, o_2) = \mathcal{P}(o_1 + o_2) - \mathcal{P}(o_1) - \mathcal{P}(o_2) = 0 \in \mathbb{F}_q^m$$

→ If $o_1$ and $o_2$ are unknown, this is a quadratic system that is hard to solve
→ If $o_1$ is known, this presents $m$ linear equations for $o_2$
Algebraic Attack

Knowledge of an oil vector dramatically simplifies algebraic key recovery attacks

- For two oil vectors $o_1, o_2$ it holds

$$\mathcal{P}'(o_1, o_2) = \mathcal{P}(o_1 + o_2) - \mathcal{P}(o_1) - \mathcal{P}(o_2) = 0 \in \mathbb{F}_q^m$$

→ If $o_1$ and $o_2$ are unknown, this is a quadratic system that is hard to solve
→ If $o_1$ is known, this presents $m$ linear equations for $o_2$

- With the given UOV parameters, this implies: If two oil vectors are known, the remaining oil space can be found in polynomial time
Knowledge of an oil vector dramatically simplifies algebraic key recovery attacks

- For two oil vectors $o_1, o_2$ it holds

$$P'(o_1, o_2) = P(o_1 + o_2) - P(o_1) - P(o_2) = 0 \in \mathbb{F}_q^m$$

$\rightarrow$ If $o_1$ and $o_2$ are unknown, this is a quadratic system that is hard to solve
$\rightarrow$ If $o_1$ is known, this presents $m$ linear equations for $o_2$

- With the given UOV parameters, this implies: If two oil vectors are known, the remaining oil space can be found in polynomial time

In fact, even one oil vector is enough, when using modified Kipnis-Shamir attack $^5$

$^5$Thanks to Ward Beullens for pointing out how this attack is possible
Algebraic Attack

Knowledge of an oil vector dramatically simplifies algebraic key recovery attacks

• For two oil vectors $o_1, o_2$ it holds

$$\mathcal{P}'(o_1, o_2) = \mathcal{P}(o_1 + o_2) - \mathcal{P}(o_1) - \mathcal{P}(o_2) = 0 \in \mathbb{F}^m$$

→ If $o_1$ and $o_2$ are unknown, this is a quadratic system that is hard to solve
→ If $o_1$ is known, this presents $m$ linear equations for $o_2$

• With the given UOV parameters, this implies: If two oil vectors are known, the remaining oil space can be found in polynomial time

In fact, even one oil vector is enough, when using modified Kipnis-Shamir attack\(^5\)

Details can be found in [ACK+23]\(^6\)

\(^5\)Thanks to Ward Beullens for pointing out how this attack is possible
Summary

- **Instruction skip** to reuse the vinegar variables
- **Number** of needed **faulted signatures** is reduced from $m$ to now only 1
- Distinguish between reuse and zero setting (analyzed in [SK20]$^7$ and [KKT22]$^8$)

---


Summary of the Fault Attack [SK20]

Summary

- **Instruction skip** to reuse the vinegar variables
- **Number** of needed **faulted signatures** is reduced from \( m \) to now only 1
- Distinguish between reuse and zero setting (analyzed in [SK20]^7 and [KKT22]^8)

Practicality

- Attack is simulated targeting Rainbow on an emulated ARM M4 architecture using QEMU in [AKK+22]^9

---


Summary of the Fault Attack [SK20]

Summary

- **Instruction skip** to reuse the vinegar variables
- **Number of needed faulted signatures** is reduced from $m$ to now only 1
- Distinguish between reuse and zero setting (is analyzed in [SK20] and [KKT22])

Practicality

- Attack is simulated targeting Rainbow on an emulated ARM M4 architecture using QEMU in [AKK+22]

Countermeasures

- ‘Verify before output’ is not possible, since faulted signature is valid
- Store old vinegar variables and only output signature if there are no large overlaps
Summary of the Fault Attack [SK20]

Summary

- **Instruction skip** to reuse the vinegar variables
- **Number** of needed **faulted signatures** is reduced from \( m \) to now only 1
- Distinguish between reuse and zero setting (is analyzed in [SK20] and [KKT22])

Practicality

- Attack is simulated targeting Rainbow on an emulated ARM M4 architecture using QEMU in [AKK+22]

Countermeasures

- ‘Verify before output’ is not possible, since faulted signature is valid
- Store old vinegar variables and only output signature if there are no large overlaps

Future work

- Execute instruction skip on a target device
- Apply to various modifications of UOV
Bit-Flip in Central Map

Fault model

- Introduce a fault that changes one coefficient $\alpha_{i,j}'^{(k)}$ in the central map $F$ (already discussed in [HTS11] and [KL19])
- Faulted coefficient is randomly chosen and attacker does not know its location

$$F'(k) = \begin{pmatrix}
\alpha_{1,1}^{(k)} & \ldots & \alpha_{1,v}^{(k)} & \alpha_{1,v+1}^{(k)} & \ldots & \alpha_{1,n}^{(k)} \\
0 & \ddots & \vdots & \vdots & \vdots & \alpha_{i,j}'^{(k)} & \vdots \\
0 & 0 & \alpha_{v,v}^{(k)} & \alpha_{v,v+1}^{(k)} & \ldots & \alpha_{v,n}^{(k)} \\
0 & \ldots & 0 & 0 & \ldots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \ldots & 0 & 0 & \ldots & 0
\end{pmatrix}$$
Bit-Flip in Central Map

Fault model

- Introduce a fault that changes one coefficient $\alpha'_{i,j}^{(k)}$ in the central map $\mathcal{F}$ (already discussed in [HTS11] and [KL19])
- Faulted coefficient is randomly chosen and attacker does not know its location

$$
\begin{pmatrix}
\tilde{v}_1 \\
\vdots \\
\tilde{v}_v \\
y_1 \\
\vdots \\
y_m
\end{pmatrix}^T
\begin{pmatrix}
\alpha^{(k)}_{1,1} & \cdots & \alpha^{(k)}_{1,v} & \alpha^{(k)}_{1,v+1} & \cdots & \alpha^{(k)}_{1,n} \\
0 & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \alpha^{(k)}_{v,v} & \alpha^{(k)}_{v,v+1} & \cdots & \alpha^{(k)}_{v,n} \\
0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 0 & \cdots & 0
\end{pmatrix}
\begin{pmatrix}
\tilde{v}_1 \\
\vdots \\
\tilde{v}_v \\
y_1 \\
\vdots \\
y_m
\end{pmatrix}
= l_1^{(k)} \cdot y_1 + \ldots + l_j^{(k)} \cdot y_j + \ldots + c^{(k)}$$
Bit-Flip in Central Map

Fault model

• Introduce a fault that changes one coefficient $\alpha'_{i,j}^{(k)}$ in the central map $F$ (already discussed in [HTS11] and [KL19])

• Faulted coefficient is randomly chosen and attacker does not know its location

\[
\begin{pmatrix}
\tilde{v}_1 \\
\vdots \\
\tilde{v}_v \\
y_1 \\
\vdots \\
y_m
\end{pmatrix}^T
\begin{pmatrix}
\alpha_{1,1}^{(k)} & \ldots & \alpha_{1,v}^{(k)} & \alpha_{1,v+1}^{(k)} & \ldots & \alpha_{1,n}^{(k)} \\
0 & \ddots & \vdots & \vdots & \vdots & \alpha'_{i,j}^{(k)} \\
0 & \ldots & \alpha_{v,v}^{(k)} & \alpha_{v,v+1}^{(k)} & \ldots & \alpha_{v,n}^{(k)} \\
y_1 & \ldots & 0 & 0 & \ldots & 0 \\
\vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\
y_m & \ldots & 0 & 0 & \ldots & 0
\end{pmatrix}
\begin{pmatrix}
\tilde{v}_1 \\
\vdots \\
\tilde{v}_v \\
y_1 \\
\vdots \\
y_m
\end{pmatrix}
= l_1^{(k)} \cdot y_1 + \ldots + l_j^{(k)} \cdot y_j + \ldots + c^{(k)}

• One coefficient in the $k$-th linear equation is altered
Bit-Flip in Central Map

Fault propagation

\[
\begin{align*}
t & \xrightarrow{\mathcal{F'}^{-1}} (v) \xrightarrow{T^{-1}} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = s'
\end{align*}
\]

- Faulted signature \( s' \) of \( t \) might deviate heavily from fault-free \( s = T^{-1} \circ \mathcal{F}^{-1}(t) \)
Fault propagation

\[
\begin{align*}
\mathbf{t} & \xrightarrow{\mathcal{F}'^{-1}} (v) \xrightarrow{T^{-1}} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = s' \\
\end{align*}
\]

- Faulted signature \(s'\) of \(\mathbf{t}\) might deviate heavily from fault-free \(s = T^{-1} \circ \mathcal{F}^{-1}(\mathbf{t})\)
- But \(\mathcal{P}(s')\) only deviates in one entry from \(\mathbf{t}\)

\[
\mathcal{P}(s') - \mathbf{t} = \mathcal{F} \circ T(s') - \mathcal{F}' \circ T(s') = (\mathcal{F} - \mathcal{F}') \circ T(s')
\]

\[
= (0, \ldots, 0, (\alpha_{i,j}^{(k)} - \alpha'_{i,j}^{(k)})(T(s')_i \cdot T(s')_j), 0, \ldots, 0)
\]
Bit-Flip in Central Map

Fault propagation

\[ t \xrightarrow{\mathcal{F}'^{-1}} \begin{pmatrix} v \\ y \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = s' \]

- Faulted signature \( s' \) of \( t \) might deviate heavily from fault-free \( s = T^{-1} \circ \mathcal{F}^{-1}(t) \)
- But \( \mathcal{P}(s') \) only deviates in one entry from \( t \)

\[
\mathcal{P}(s') - t = \mathcal{F} \circ T(s') - \mathcal{F}' \circ T(s') = (\mathcal{F} - \mathcal{F}') \circ T(s')
\]

\[
= (0, \ldots, 0, (\alpha_{i,j}^{(k)} - \alpha_{i,j}'^{(k)})(T(s')_i \cdot T(s')_j), 0, \ldots, 0)
\]

- This yields quadratic equations in the \( i \)-th and \( j \)-th row of \( T \)
Iterate the following steps to achieve key recovery (Details in [FKN+22]¹⁰)

1. Employ signing oracle to get $N = n(n + 1)/2$ message and faulted signature pairs
2. Obtain rows of the secret transformation $T$
3. Transform $\mathcal{P}$ to a smaller system by reducing the number of variables

¹⁰Furue, H., Kiyomura, Y., Nagasawa, T., and Takagi, T.: A New Fault Attack on UOV Multivariate Signature Scheme. PQCrypto 2022
Summary

- Randomization fault
- Attack needs $\approx 10 - 20$ iterations with $n^2/2$ queries to a signing oracle each round
Summary of the Fault Attack [FKN+22]

Summary

• Randomization fault
• Attack needs $\approx 10 - 20$ iterations with $n^2/2$ queries to a signing oracle each round

Practicality

• Purely theoretical $\rightarrow$ No execution of the fault attack yet
Summary of the Fault Attack [FKN+22]

Summary

- Randomization fault
- Attack needs $\approx 10 - 20$ iterations with $n^2/2$ queries to a signing oracle each round

Practicality

- Purely theoretical → No execution of the fault attack yet

Countermeasures

- Verify before returning the signature, since faulted signature is invalid
- Check if secret key is altered
Summary of the Fault Attack [FKN+22]

Summary

- Randomization fault
- Attack needs $\approx 10 - 20$ iterations with $n^2/2$ queries to a signing oracle each round

Practicality

- Purely theoretical $\rightarrow$ No execution of the fault attack yet

Countermeasures

- Verify before returning the signature, since faulted signature is invalid
- Check if secret key is altered

Future work

- Find a way to physically cause the randomization in exactly one entry
- Transfer the attack to implementation with compressed keys, where the central map is not stored
QuantumHammer: A Practical Hybrid Attack on the LUOV Signature Scheme

QuantumHammer: A Practical Hybrid Attack on the LUOV Signature Scheme\textsuperscript{11}

- Uses Rowhammer attack to introduce faults to the linear transformation $T$
  → Activate DRAM rows rapidly, to flip bits in neighboring rows (pushes voltage level above or below some threshold)

\textsuperscript{11}Mus, K., Islam, S., and Sunar, B. QuantumHammer: A Practical Hybrid Attack on the LUOV Signature Scheme. ACM SIGSAC Conference on Computer and Communications Security 2020
QuantumHammer: A Practical Hybrid Attack on the LUOV Signature Scheme

- Uses Rowhammer attack to introduce faults to the linear transformation $T$ → Activate DRAM rows rapidly, to flip bits in neighboring rows (pushes voltage level above or below some threshold)
- Software-induced hardware-fault attack

---

QuantumHammer: A Practical Hybrid Attack on the LUOV Signature Scheme

- Uses Rowhammer attack to introduce faults to the linear transformation $T$ 
  $\rightarrow$ Activate DRAM rows rapidly, to flip bits in neighboring rows (pushes voltage level above or below some threshold)
- Software-induced hardware-fault attack
- Applied the attack with $\approx 4$hrs of active Rowhammer with efficient post-processing to achieve full key recovery

---

QuantumHammer: A Practical Hybrid Attack on the LUOV Signature Scheme\textsuperscript{11}

• Uses Rowhammer attack to introduce faults to the linear transformation $T$
  → Activate DRAM rows rapidly, to flip bits in neighboring rows (pushes voltage level above or below some threshold)
• Software-induced hardware-fault attack
• Applied the attack with $\approx 4$hrs of active Rowhammer with efficient post-processing to achieve full key recovery
• Might be transferred to UOV

\textsuperscript{11}Mus, K., Islam, S., and Sunar, B. QuantumHammer: A Practical Hybrid Attack on the LUOV Signature Scheme. ACM SIGSAC Conference on Computer and Communications Security 2020
Side Channel Attacks
Horizontal SCA on Linear Transformation T

Main idea\(^{12}\)

\[
\begin{align*}
\text{t} & \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} v \\ y \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} v + T_1 \cdot y \\ y \end{pmatrix}
\end{align*}
\]

- Perform power analysis of matrix-vector multiplication

\[
\begin{pmatrix} I_v & T_1 \\ 0 & I_m \end{pmatrix} \cdot \begin{pmatrix} v \\ y \end{pmatrix} = \begin{pmatrix} v + T_1 \cdot y \\ y \end{pmatrix}
\]

\(^{12}\)Park, A., Shim, K. A., Koo, N., and Han, D. G.: Side-channel Attacks on Post-quantum Signature Schemes based on Multivariate Quadratic Equations:--Rainbow and UOV. IACR TCHES 2018
Horizontal SCA on Linear Transformation $T$

**Main idea**

$\mathbf{t} \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} \mathbf{v} \\ \mathbf{y} \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{v} + T_1 \cdot \mathbf{y} \\ \mathbf{y} \end{pmatrix}$

- Perform power analysis of matrix-vector multiplication

\[ \begin{pmatrix} I_v & T_1 \\ 0 & I_m \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{v} + T_1 \cdot \mathbf{y} \\ \mathbf{y} \end{pmatrix} \]

- Here, the vector $\mathbf{y}$ is known, and the matrix $T_1$ is the secret we want to obtain

---

12Park, A., Shim, K. A., Koo, N., and Han, D. G.: Side-channel Attacks on Post-quantum Signature Schemes based on Multivariate Quadratic Equations:-Rainbow and UOV. IACR TCHES 2018
Horizontal SCA on Linear Transformation $T$

Main idea\textsuperscript{12}

\[ t \xrightarrow{\mathcal{F}^{-1}} \begin{pmatrix} v \\ y \end{pmatrix} \xrightarrow{T^{-1}} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} v + T_1 \cdot y \\ y \end{pmatrix} \]

- Perform power analysis of matrix-vector multiplication

\[ \begin{pmatrix} l_v & T_1 \\ 0 & I_m \end{pmatrix} \cdot \begin{pmatrix} v \\ y \end{pmatrix} = \begin{pmatrix} v + T_1 \cdot y \\ y \end{pmatrix} \]

- Here, the vector $y$ is known, and the matrix $T_1$ is the secret we want to obtain
- Either obtain all entries of $T$ by SCA or identify certain rows and reduce the system \( P \) as shown in previous fault attack

\textsuperscript{12}Park, A., Shim, K. A., Koo, N., and Han, D. G.: Side-channel Attacks on Post-quantum Signature Schemes based on Multivariate Quadratic Equations:-Rainbow and UOV. IACR TCHES 2018
Matrix-Vector Multiplication

The vulnerable function in more detail

\[
\begin{pmatrix}
1 & 0 & 0 & t_{1,4} & t_{1,5} \\
0 & 1 & 0 & t_{2,4} & t_{2,5} \\
0 & 0 & 1 & t_{3,4} & t_{3,5} \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{pmatrix}
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
y_1 \\
y_2 \\
\end{pmatrix}
= 
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
y_1 \\
y_2 \\
\end{pmatrix}
+ 
\begin{pmatrix}
t_{1,4} \cdot y_1 + t_{1,5} \cdot y_2 \\
t_{2,4} \cdot y_1 + t_{2,5} \cdot y_2 \\
t_{3,4} \cdot y_1 + t_{3,5} \cdot y_2 \\
0 \\
0 \\
\end{pmatrix}
\]

Correlation power analysis

1. Guess intermediate values and map hypothetical value to hypothetical power consumption of the function under investigation
2. Measure the power consumption of the target device
3. Perform statistical comparison between hypothetical power consumption and measured power traces
The vulnerable function in more detail

\[
\begin{pmatrix}
I & T_1 \\
0 & I
\end{pmatrix} \cdot 
\begin{pmatrix}
v \\
y
\end{pmatrix} = 
\begin{pmatrix}
1 & 0 & 0 & t_{1,4} & t_{1,5} \\
0 & 1 & 0 & t_{2,4} & t_{2,5} \\
0 & 0 & 1 & t_{3,4} & t_{3,5} \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix} \cdot 
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
y_1 \\
y_2
\end{pmatrix} = 
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
y_1 \\
y_2
\end{pmatrix} + 
\begin{pmatrix}
0 \\
0 \\
0 \\
t_{1,4} \cdot y_1 + t_{1,5} \cdot y_2 \\
t_{2,4} \cdot y_1 + t_{2,5} \cdot y_2 \\
t_{3,4} \cdot y_1 + t_{3,5} \cdot y_2 \\
0 \\
0
\end{pmatrix}
\]

Correlation power analysis

1. Guess intermediate values and map hypothetical value to hypothetical power consumption of the function under investigation
Matrix-Vector Multiplication

The vulnerable function in more detail

\[
\begin{pmatrix}
I & T_1 \\
0 & I
\end{pmatrix}
\cdot
\begin{pmatrix}
v \\
y
\end{pmatrix}
= 
\begin{pmatrix}
1 & 0 & 0 & t_{1,4} & t_{1,5} \\
0 & 1 & 0 & t_{2,4} & t_{2,5} \\
0 & 0 & 1 & t_{3,4} & t_{3,5} \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
y_1 \\
y_2
\end{pmatrix}
= 
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
y_1 \\
y_2
\end{pmatrix}
+ 
\begin{pmatrix}
t_1,4 \cdot y_1 + t_1,5 \cdot y_2 \\
t_2,4 \cdot y_1 + t_2,5 \cdot y_2 \\
t_3,4 \cdot y_1 + t_3,5 \cdot y_2 \\
0 \\
0
\end{pmatrix}
\]

Correlation power analysis

1. Guess intermediate values and map hypothetical value to hypothetical power consumption of the function under investigation
2. Measure the power consumption of the target device
Matrix-Vector Multiplication

The vulnerable function in more detail

\[
\begin{pmatrix}
1 & 0 & 0 & t_{1,4} & t_{1,5} \\
0 & 1 & 0 & t_{2,4} & t_{2,5} \\
0 & 0 & 1 & t_{3,4} & t_{3,5} \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
\cdot
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
v_1 \\
v_2
\end{pmatrix}
= 
\begin{pmatrix}
1 & 0 & 0 & t_{1,4} & t_{1,5} \\
0 & 1 & 0 & t_{2,4} & t_{2,5} \\
0 & 0 & 1 & t_{3,4} & t_{3,5} \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
\cdot
\begin{pmatrix}
v_1 \\
v_2 \\
v_3 \\
v_1 \\
v_2
\end{pmatrix}
+ 
\begin{pmatrix}
t_{1,4} \cdot y_1 + t_{1,5} \cdot y_2 \\
t_{2,4} \cdot y_1 + t_{2,5} \cdot y_2 \\
t_{3,4} \cdot y_1 + t_{3,5} \cdot y_2 \\
0 \\
0
\end{pmatrix}
\]

Correlation power analysis

1. Guess intermediate values and map hypothetical value to hypothetical power consumption of the function under investigation
2. Measure the power consumption of the target device
3. Perform statistical comparison between hypothetical power consumption and measured power traces
Compute Correlation with Hypothetical Values

Example with clear separation between correct key elements and wrong key element

(a) Maximum correlation coefficients according to increased number of traces for $t_{45}$

Correlation coefficients for all possible field elements and the entry $t_{45}$ [PSK+18]
Compute Correlation with Hypothetical Values

Example with two possible candidate for the correct key element

(b) Maximum correlation coefficients according to increased number of traces for $\tilde{t}_{46}$

Correlation coefficients for all possible field elements and the entry $t_{46}$ [PSK+18]
Summary

• Correlation power analysis on field multiplication
• Around 30 – 100 power traces are needed to recover field elements
Summary of the Horizontal SCA

Summary

- Correlation power analysis on field multiplication
- Around 30 – 100 power traces are needed to recover field elements

Practicality

- Attack the matrix-vector product code on the ChipWhisperer-Lite evaluation platform
- Target board is an 8-bit Atmel XMEGA128 (might be more difficult on 32-bit devices)
- Parameters were strongly reduced ($n = 8$ and $m = 6$)
Summary of the Horizontal SCA

Summary

- Correlation power analysis on field multiplication
- Around 30 – 100 power traces are needed to recover field elements

Practicality

- Attack the matrix-vector product code on the ChipWhisperer-Lite evaluation platform
- Target board is an 8-bit Atmel XMEGA128 (might be more difficult on 32-bit devices)
- Parameters were strongly reduced ($n = 8$ and $m = 6$)

Countermeasures

- Masking or shuffling are classical countermeasures for this
- Randomization of the input value (since $T$ is linear)
Summary of the Horizontal SCA

Summary

- Correlation power analysis on field multiplication
- Around 30 – 100 power traces are needed to recover field elements

Practicality

- Attack the matrix-vector product code on the ChipWhisperer-Lite evaluation platform
- Target board is an 8-bit Atmel XMEGA128 (might be more difficult on 32-bit devices)
- Parameters were strongly reduced ($n = 8$ and $m = 6$)

Countermeasures

- Masking or shuffling are classical countermeasures for this
- Randomization of the input value (since $T$ is linear)

Future work

- Analyze efficiency impact of countermeasures
- Transfer the attack to modern and optimized implementations
Main idea

- Measure power consumption of $\mathcal{P}(v)$

---

Main idea\textsuperscript{13}

• Measure power consumption of $\mathcal{P}(\mathbf{v})$
• This operation boils down to computing $\mathbf{v}^\top \mathcal{P}(k) \mathbf{v}$ for $m$ known matrices $\mathcal{P}(k)$

\textsuperscript{13}Aulbach, T., Campos, F., Krämer, J., Samardjiska, S., and Stöttinger, M.: Separating Oil and Vinegar with a Single Trace: Side-Channel Assisted Kipnis-Shamir Attack on UOV. IACR TCHES 2023
Main idea\textsuperscript{13}

- Measure power consumption of $P(v)$
- This operation boils down to computing $v^T P^{(k)} v$ for $m$ known matrices $P^{(k)}$
- Consider the matrix-vector multiplication

$$P^{(k)} \cdot v = \begin{pmatrix} p_{1,1}^{(k)} & p_{1,2}^{(k)} & \cdots & p_{1,n}^{(k)} \\ p_{2,1}^{(k)} & p_{2,2}^{(k)} & \cdots & p_{2,n}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1}^{(k)} & p_{n,2}^{(k)} & \cdots & p_{n,n}^{(k)} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \text{ for } k \in \{1, \ldots, m\}$$

\textsuperscript{13}Aulbach, T., Campos, F., Krämer, J., Samardjiska, S., and Stöttinger, M.: Separating Oil and Vinegar with a Single Trace: Side-Channel Assisted Kipnis-Shamir Attack on UOV. IACR TCHES 2023
Main idea

- Measure power consumption of $P(v)$
- This operation boils down to computing $v^T P^{(k)} v$ for $m$ known matrices $P^{(k)}$
- Consider the matrix-vector multiplication

$$P^{(k)} \cdot v = \begin{pmatrix} p^{(k)}_{1,1} & p^{(k)}_{1,2} & \cdots & p^{(k)}_{1,n} \\ p^{(k)}_{2,1} & p^{(k)}_{2,2} & \cdots & p^{(k)}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p^{(k)}_{n,1} & p^{(k)}_{n,2} & \cdots & p^{(k)}_{n,n} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \text{ for } k \in \{1, \ldots, m\}$$

- Secret $v$ is multiplied with a considerable amount of known values

---

Template Attack

- Create a template by tracing the power consumption of

\[ p^{(k)} \cdot v = \begin{pmatrix} p_{1,1}^{(k)} & p_{1,2}^{(k)} & \cdots & p_{1,n}^{(k)} \\ p_{2,1}^{(k)} & p_{2,2}^{(k)} & \cdots & p_{2,n}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1}^{(k)} & p_{n,2}^{(k)} & \cdots & p_{n,n}^{(k)} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \text{ for } k \in \{1, \ldots, m\} \]

for \( v = 0, 1, 2, \ldots, q - 1 \in \mathbb{F}_q^m \)
Template Attack

- Create a template by tracing the power consumption of

\[
p^{(k)} \cdot v = \begin{pmatrix}
p_{1,1}^{(k)} & p_{1,2}^{(k)} & \cdots & p_{1,n}^{(k)} \\
p_{2,1}^{(k)} & p_{2,2}^{(k)} & \cdots & p_{2,n}^{(k)} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n,1}^{(k)} & p_{n,2}^{(k)} & \cdots & p_{n,n}^{(k)}
\end{pmatrix}
\begin{pmatrix}
v_1 \\
v_2 \\
\vdots \\
v_n
\end{pmatrix}
\] for \( k \in \{1, \ldots, m\} \)

for \( v = 0, 1, 2, \ldots, q - 1 \in \mathbb{F}_q^m \)

- Multiplication of field elements

\[
p_{1,1}^{(k)} \cdot v_1 \quad p_{2,2}^{(k)} \cdot v_2 \quad \cdots \quad p_{n,n}^{(k)} \cdot v_n
\]
Template Attack

- Create a template by tracing the power consumption of

\[ P^{(k)} \cdot v = \begin{pmatrix} p_{1,1}^{(k)} & p_{1,2}^{(k)} & \cdots & p_{1,n}^{(k)} \\
 p_{2,1}^{(k)} & p_{2,2}^{(k)} & \cdots & p_{2,n}^{(k)} \\
 \vdots & \vdots & \ddots & \vdots \\
 p_{n,1}^{(k)} & p_{n,2}^{(k)} & \cdots & p_{n,n}^{(k)} \end{pmatrix} \begin{pmatrix} v_1 \\
 v_2 \\
 \vdots \\
 v_n \end{pmatrix} \quad \text{for } k \in \{1, \ldots, m\} \]

for \( v = 0, 1, 2, \ldots, q - 1 \in \mathbb{F}_q^m \)

- Multiplication of field elements

\[ p_{1,1}^{(k)} \cdot v_1, p_{2,2}^{(k)} \cdot v_2, \ldots, p_{n,n}^{(k)} \cdot v_n \]

- For each field element, we need to run and trace the matrix-vector multiplication only once \( \rightarrow \) in total \( q = 256 \) profiling traces

Can be collected on another device (subtract some mean to erase the 'footprint' of the device)
Template Attack

- Create a template by tracing the power consumption of

\[
P^{(k)} \cdot v = \begin{pmatrix} p^{(k)}_{1,1} & p^{(k)}_{1,2} & \cdots & p^{(k)}_{1,n} \\ p^{(k)}_{2,2} & \cdots & p^{(k)}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p^{(k)}_{n,n} & \vdots & \cdots & \vdots \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}
\]

for \( k \in \{1, \ldots, m\} \)

- Multiplication of field elements

\[
p^{(k)}_{1,1} \cdot v_1 \quad p^{(k)}_{2,2} \cdot v_2 \quad \ldots \quad p^{(k)}_{n,n} \cdot v_n
\]

- For each field element, we need to run and trace the matrix-vector multiplication only once \( \rightarrow \) in total \( q = 256 \) profiling traces
- Can be collected on another device (subtract some mean to erase the 'footprint' of the device)
Record Power Traces

- Power traces are very distinctive

Consider the following example: Compare power traces with $v_i = 0xFF$ vs $v_i = 0xEB$.
Record Power Traces

- Power traces are very distinctive
- The vinegar variables are processed bitwise from LSB to MSB
Record Power Traces

- Power traces are very distinctive
- The vinegar variables are processed bitwise from LSB to MSB
- Consider the following example

Compare power traces with $v_i = 0xFF$ vs $v_i = 0xEB$
Compute Correlation

- Trace the matrix-vector multiplications with secret vinegar variables on the target device
- Compute correlation to templates for each entry of $v$
Compute Correlation

- Trace the matrix-vector multiplications with secret vinegar variables on the target device
- Compute correlation to templates for each entry of $v$

Correlation of the target trace with each of the 256 reference traces
Summary of the Template Attack

Summary

• Very high success probability (≈ 97%) for all vinegar variables
• Template attack with small number of profiling traces
• One single attack trace leads to a secret oil vector (key recovery)
Summary of the Template Attack

Summary

• Very high success probability (≈ 97%) for all vinegar variables
• Template attack with small number of profiling traces
• One single attack trace leads to a secret oil vector (key recovery)

Practicality

• Attack executed with the ChipWhisperer-Lite on an 32-bit STM32F3 target board
• Parameter set only slightly reduced, s.t. $\mathcal{P}$ fits on the target board
• Used modern UOV implementation

29
Summary of the Template Attack

Summary

- Very high success probability (≈ 97%) for all vinegar variables
- Template attack with small number of profiling traces
- One single attack trace leads to a secret oil vector (key recovery)

Practicality

- Attack executed with the ChipWhisperer-Lite on an 32-bit STM32F3 target board
- Parameter set only slightly reduced, s.t. $P$ fits on the target board
- Used modern UOV implementation

Countermeasures

- Masking or shuffling are classical countermeasures for this
Summary of the Template Attack

Summary

• Very high success probability (≈ 97%) for all vinegar variables
• Template attack with small number of profiling traces
• One single attack trace leads to a secret oil vector (key recovery)

Practicality

• Attack executed with the ChipWhisperer-Lite on an 32-bit STM32F3 target board
• Parameter set only slightly reduced, s.t. $\mathcal{P}$ fits on the target board
• Used modern UOV implementation

Countermeasures

• Masking or shuffling are classical countermeasures for this

Future work

• Analyze efficiency impact of countermeasures
• Apply the attack to M4 implementations or using a different setup
Takeaways
Takeaways

- Vinegar vectors and oil vectors should be equally secured
- With one of those, the secret key can be recovered in polynomial time
- Some physical attacks are still in a theoretical or simulated state
- Efficiency impact of countermeasures should be analyzed

Questions?
Contact: thomas.aulbach@ur.de

Aulbach, Campos, Krämer, Samardjiska, Stöttinger: Separating Oil and Vinegar with a Single Trace
https://ia.cr/2023/335


