Side-channel analysis in code-based cryptography

Tania RICHMOND

IMATH Laboratory
University of Toulon

SoSySec Seminar
Rennes, April 5, 2017
Outline

McEliece cryptosystem

Timing Attack

Power consumption Attack

Conclusion
Outline

McEliece cryptosystem

Timing Attack

Power consumption Attack

Conclusion
Communication

Once upon a time ...

a woman, Alice

and a man, Bob

who wanted to communicate together.
But, they did not want that anyone, could understand this message.
Cryptology

\textit{science of secret}

\textit{kryptos} = secret/hidden \hspace{1cm} \textit{logos} = science

Two concepts:

Cryptography
"Secret writing"

Good Man

Cryptanalysis
"Analysis of a secret"

Bad Man
Cryptography

Using a one-way function to do:

- Encryption
- Identification
- Signature
- Hash function
Cryptography

Using a one-way function to do:

- **Encryption**
- **Signature**
- **Identification**
- **Hash function**
Secret-Key Cryptosystem

Ceasar
Public-Key Cryptosystem (PKC)

[DH76]^{1}

---

Post-quantum cryptography

- Code-based cryptography
- Lattice-based cryptography
- Hash-based cryptography
- Multivariate-based cryptography

No solving in polynomial time, contrary to number theory problems [Sho97]$^2$

Post-quantum cryptography

- Code-based cryptography
- Lattice-based cryptography
- Hash-based cryptography
- Multivariate-based cryptography

No solving in polynomial time, contrary to number theory problems [Sho97]²

**Linear code**

**Definition (Linear code)**
Let $q = p^m$ be a power $m > 0$ of some prime $p$. Let $\mathbb{F}_q$ denoted the finite field of $q$ elements. A linear code $C$ of length $n$ and dimension $k$ is a $k$-dimensional subspace of $\mathbb{F}_q^n$.

**Definition (Generator matrix)**
Let $C$ be a $[n, k]_q$-linear code. Let $G \in \mathcal{M}_{k,n}(\mathbb{F}_q)$. We call $G$ a generator matrix of $C$ iff $G$-rows are basis vectors of $C$.
Syndrome Decoding (SD) problem [BMcEvT78]\(^3\)

**Inputs**

\( \mathcal{H} \) matrix of size \( r \times n \),

\( S \) binary vector of length \( r \),

\( t \) interger.

**Problem**

Does there exist a binary vector \( e \) of length \( n \) and weight \( t \) such that:

\[ r = n - k \]

\( S \) is called syndrome.

**Theorem**

\( SD \) is NP-complete.

---

McEliece PKC

Proposed in [McE78].

**Key generation:**

- sk: \((Q, G, P)\)
- pk: \((G', m, t)\)

\[ G' = Q \cdot G \cdot P \]

**Encryption:**

1. Message encoding into a codeword
2. Error vector adding to the codeword

**Decryption:**

1. Ciphertext permutation
2. Syndrome computation
3. Solving of the key equation
4. Error position finding

---

Cryptography
Theory vs. Practice

Mathematics VS. Implementations
Side-Channel Attack (SCA)

Definition (SCA)

Exploit the laws of physics phenomenons to obtain some information contained in channels associated to an implementation (software or hardware).

1st SCA in [Koc96]\(^5\)

---

Outline

McEliece cryptosystem

Timing Attack

Power consumption Attack

Conclusion
McEliece PKC

Proposed in [McE78] ⁶.

**Key generation:**

- **sk:** \((Q, G, P)\)
- **pk:** \((G', m, t)\)

\[ G' = Q \cdot G \cdot P \]

**Encryption:**

1. Message encoding into a codeword
2. Error vector adding to the codeword

**Decryption:**

1. Ciphertext permutation
2. Syndrome computation
3. **Solving of the key equation**
4. Error position finding

McEliece Decryption

Attack on the Extended Euclidean Algorithm (EEA) double call

1. Ciphertext permutation
2. Syndrome computation
3. Solving the key equation (with Patterson [Pat75]7):
   3.1 Syndrome inversion (via EEA)
   3.2 Square root computation
   3.3 Error locator polynomial (ELP) determination (via EEA)
4. Errors positions finding
   4.1 ELP evaluation
   4.2 Errors correction

Attack on the 2nd EEA call

Error locator polynomial (ELP) determination [Str10]

ELP determination attack [Str10] \(^8\) with error weight \(\varepsilon = 4 < t\):

\[
\sigma(X) = \sigma_4 \cdot X^4 \oplus \sigma_3 \cdot X^3 \oplus \sigma_2 \cdot X^2 \oplus \sigma_1 \cdot X \oplus \sigma_0
\]

\[
= a^2(X) \oplus X \cdot b^2(X)
\]

with:

\[
a(X) = \sigma_4 \cdot X^2 \oplus \sigma_2 \cdot X \oplus \sigma_0
\]

\[
b(X) = \sigma_3 \cdot X \oplus \sigma_1
\]

Two possible cases:

If \(\deg(b) = 0\): then \(\sigma_3 = 0\).

If \(\deg(b) = 1\): then \(\sigma_3 \neq 0\).

Attack on the 2nd EEA call
Error locator polynomial (ELP) determination [Str10]

\[ b(X) = \sigma_3 \cdot X \oplus \sigma_1 \]

Two possible cases:

If \( \deg(b) = 0 \): then

\[ \sigma_3 = \alpha \mathcal{P}(i_1) \oplus \alpha \mathcal{P}(i_2) \oplus \alpha \mathcal{P}(i_3) \oplus \alpha \mathcal{P}(i_4) = 0, \]

so max 2 iterations in EEA.

If \( \deg(b) = 1 \): then

\[ \sigma_3 = \alpha \mathcal{P}(i_1) \oplus \alpha \mathcal{P}(i_2) \oplus \alpha \mathcal{P}(i_3) \oplus \alpha \mathcal{P}(i_4) \neq 0, \]

so max 1 iteration in EEA.
Attack on the 1st EEA call

Syndrome inversion [Str13]

Syndrome inversion attack [Str13] \(^9\) with error weight \(\varepsilon = 4 < t\):

\[
S_E(X) \equiv \frac{\sigma'_E(X)}{\sigma_E(X)} \equiv \sum_{i=1}^{n} \frac{1}{X \oplus \alpha_i} \mod G(X)
\]

\[
S_E(X) \equiv \frac{\sigma_3 \cdot X^2 \oplus \sigma_1}{X^4 \oplus \sigma_3 \cdot X^3 \oplus \sigma_2 \cdot X^2 \oplus \sigma_1 \cdot X \oplus \sigma_0} \mod G(X).
\]

Attack on the 1st EEA call

Syndrome inversion [Str13]

Syndrome inversion attack [Str13] \(^9\) with error weight \(\varepsilon = 4 < t\):

\[
S_E(X) \equiv \frac{\sigma'_E(X)}{\sigma_E(X)} \equiv \sum_{i=1}^{n} \frac{1}{X \oplus \alpha_i} \mod G(X)
\]

\[
S_E(X) \equiv \frac{\sigma_3 \cdot X^2 \oplus \sigma_1}{X^4 \oplus \sigma_3 \cdot X^3 \oplus \sigma_2 \cdot X^2 \oplus \sigma_1 \cdot X \oplus \sigma_0} \mod G(X).
\]

Attack on the 1st EEA call

Syndrome inversion [Str13]

Attack with error weight $\varepsilon = 4 < t$:

$$\sigma'_E(X) = \sigma_3 \cdot X^2 \oplus \sigma_1,$$

with:

$$\sigma_3 = \alpha P(i_1) \oplus \alpha P(i_2) \oplus \alpha P(i_3) \oplus \alpha P(i_4).$$

Two possible cases:

- If $\sigma_3 = 0$: then $\deg(\sigma'_E) = 0$, so max 4 iterations in EEA.
- If $\sigma_3 \neq 0$: then $\deg(\sigma'_E) = 2$, so max 6 iterations in EEA.
Attack on the EEA double call

Relation between both EEA calls [BCDR17]

Relation between both EEA calls exploited in [BCDR17]¹⁰:
For an error weight $\varepsilon$ s.t. $4 \leq \varepsilon \leq t/2$ and $2|\varepsilon$:

If $\sigma_{\varepsilon-1} \neq 0$: then

1. Syndrome inversion: max $2\varepsilon - 2$ iterations.
2. ELP determination: max $\varepsilon/2$ iterations.

If $\sigma_{\varepsilon-1} = 0$ and $\sigma_{\varepsilon3} \neq 0$: then

1. Syndrome inversion: max $2\varepsilon - 4$ iterations.
2. ELP determination: max $\varepsilon/2 - 1$ iterations.

---

Attack on the EEA double call
Relation between both EEA calls [BCDR17]

Our proposition
Combine two previous attacks, [Str10] et [Str13], to obtain relations between support elements up to $t/2$ elements.

Countermeasure
Do additional iterations to make the iteration number constant in EEA (i.e. not depending on the error weight but only on the Goppa polynomial degree).
Attack on the EEA double call
Relation between both EEA calls [BCDR17]

Our proposition
Combine two previous attacks, [Str10] et [Str13], to obtain relations between support elements up to $t/2$ elements.

Countermeasure
Do additional iterations to make the iteration number constant in EEA (i.e. not depending on the error weight but only on the Goppa polynomial degree).

Remark

Possible attack by power analysis?
Or fault injection like the Square-and-Multiply algorithm?
Outline

McEliece cryptosystem

Timing Attack

Power consumption Attack

Conclusion
McEliece PKC

Proposed in [McE78] \(^{11}\).

**Key generation:**

sk: \((Q, G, P)\)

pk: \((G', m, t)\)

\[ G' = Q \cdot G \cdot P \]

**Encryption:**

1. Message encoding into a codeword
2. Error vector adding to the codeword

**Decryption:**

1. Ciphertext permutation
2. Syndrome computation
3. Solving of the key equation
4. Error position finding

Four profiles of implementation [HMP10]\(^{12}\) for ciphertext permutation and syndrome computation

<table>
<thead>
<tr>
<th>Profile I</th>
<th>Profile II</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tilde{C}_p = \tilde{C} \cdot P^{-1})</td>
<td>(\tilde{C}_p = \tilde{C} \cdot P^{-1})</td>
</tr>
<tr>
<td>Polynomial operations</td>
<td>Polynomial operations</td>
</tr>
<tr>
<td>for Goppa codes (\Gamma(\mathcal{L}, G)) with (\mathcal{L}) and (G)</td>
<td>(S = \tilde{C}_p \cdot H^T)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile III</th>
<th>Profile IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mathcal{L}_p \approx \mathcal{L} \cdot P)</td>
<td>(H_p^T = P^{-1} \cdot H^T)</td>
</tr>
<tr>
<td>Polynomial operations</td>
<td>(S = \tilde{C} \cdot H_p^T)</td>
</tr>
<tr>
<td>for Goppa codes (\Gamma(\mathcal{L}, G)) with (\mathcal{L}_p) and (G)</td>
<td></td>
</tr>
</tbody>
</table>

Attack platform scheme

ARM Cortex-M3 microprocessor
Simple Power Analysis (SPA) on the syndrome computation
Vector-matrix product

McEliece Decryption:
1. Ciphertext permutation
2. Syndrome computation: $S = \tilde{C}_p \cdot \mathcal{H}$
3. Solving of the key equation
4. Error position finding
Four profiles of implementation [HMP10] for ciphertext permutation and syndrome computation

Profile I
\[ \tilde{C}_p = \tilde{C} \cdot P^{-1} \]
Polynomial operations for Goppa codes \( \Gamma(\mathcal{L}, G) \) with \( \mathcal{L} \) and \( G \)

Profile II
\[ \tilde{C}_p = \tilde{C} \cdot P^{-1} \]
\[ S = \tilde{C}_p \cdot H^T \]

Profile III
\[ \mathcal{L}_p \approx \mathcal{L} \cdot P \]
Polynomial operations for Goppa codes \( \Gamma(\mathcal{L}, G) \) with \( \mathcal{L}_p \) and \( G \)

Profile IV
\[ H_p^T = P^{-1} \cdot H^T \]
\[ S = \tilde{C} \cdot H_p^T \]
Syndrome computation

Algorithm

Inputs: Permutated ciphertext $\tilde{C}_p \in \mathbb{F}_2^n$, parity-check matrix $H \in \mathbb{M}_{r,n}(\mathbb{F}_2)$.

For $i = 1$ to $n$

If $\tilde{C}_{p_i} = 1$

$S = S \oplus H_i$

EndIf

EndFor

Return $S$.

Output: Syndrome $S \in \mathbb{F}_2^r$ of $\tilde{C}_p$. 
Syndrome computation

Scheme

\[
\begin{array}{c}
1 & 1 & \tilde{C}_p & \cdots & 1 \\
\end{array}
\]

\[
\begin{array}{c}
r \\
\end{array}
\]

\[
\begin{array}{c}
\mathcal{H} & \cdots & \mathcal{H} \\
\end{array}
\]

\[
\begin{array}{c}
\mathcal{S} \\
\end{array}
\]

\[
\begin{array}{c}
n \\
\end{array}
\]
SPA on the syndrome computation [PRDCF15]\(^\text{13}\)

Toy example

SPA with Chosen Ciphertext Attack (CCA)

\(^{13}\text{M. Petrvalsk, T. Richmond, M. Drutarovsk, P.-L. Cayrel and V. Fischer, Countermeasure against the SPA attack on an embedded McEliece cryptosystem, IEEE, International Conference Radioelektronika 2015, pp. 462-466, 2015.}\)
Countermeasure [PRDCF15]
First algorithm

Inputs: Permuted ciphertext $\tilde{C}_p \in \mathbb{F}_2^n$, parity-check matrix $H \in \mathcal{M}_{r,n}(\mathbb{F}_2)$. 
words = $r / \text{sizeof}(S)$ Required number of bytes to store $S$

For $i = 1$ to $n$
   $\text{tmp} = \text{unsigned}(0 - \tilde{C}_{p_i})$
   For $j = 1$ to words
      $S_j = S_j \oplus H_{i,j} \& \text{tmp}$
   EndFor
EndFor

Return $S$.

Output: Syndrome $S \in \mathbb{F}_2^r$ of $\tilde{C}_p$. 
Countermeasure [PRDCF15]

Second algorithm

Inputs: Permuted ciphertext $\tilde{C}_p \in \mathbb{F}_2^n$, parity-check matrix $\mathcal{H} \in \mathcal{M}_{r,n}(\mathbb{F}_2)$.

$\text{words} = r / \text{sizeof}(S)$ Required number of bytes to store $S$

Syndrome masking

$\text{For } j = 1 \text{ to } \text{words}$

$S_j = S_j \& 0xAAAA$

$\text{EndFor}$

Syndrome computation

$\text{For } i = 1 \text{ to } n$

$\text{tmp} = \text{unsigned}(0 - \tilde{C}_{p_i})$

$\text{For } j = 1 \text{ to } \text{words}$

$S_j = S_j \oplus \mathcal{H}_{i,j} \& \text{tmp}$

$\text{EndFor}$

$\text{EndFor}$

Syndrome unmasking

$\text{For } j = 1 \text{ to } \text{words}$

$S_j = S_j \& 0xAAAA$

$\text{EndFor}$

Return $S$.

Output: Syndrome $S \in \mathbb{F}_2^r$ of $\tilde{C}_p$. 

Syndrome computation with countermeasure [PRDCF15]

Scheme

$S$ \Rightarrow \mathcal{H} \Rightarrow \hat{C}_p \Rightarrow 1 \Rightarrow S$

$i \Rightarrow 1 \Rightarrow \hat{C}_p \Rightarrow S$

$r \Rightarrow S$

$n \Rightarrow S$

$A A A A$

$S$

$\Rightarrow \mathcal{H}$

$\Rightarrow \hat{C}_p$

$1$

$S$

$\Rightarrow \mathcal{H}$

$\Rightarrow \hat{C}_p$

$1$

$S$
Differential Power Analysis (DPA) on the ciphertext permutation
For example vector-matrix product

Decryption:

1. Ciphertext permutation: $\tilde{C}_p = \tilde{C} \cdot P^{-1}$
2. Syndrome computation
3. Solving of the key equation
4. Error position finding
Four profiles of implementation [HMP10]
for ciphertext permutation and syndrome computation

<table>
<thead>
<tr>
<th>Profile I</th>
<th>Profile II</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$</td>
<td>$\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$</td>
</tr>
<tr>
<td>Polynomial operations</td>
<td>$S = \tilde{C}_p \cdot \mathcal{H}^T$</td>
</tr>
<tr>
<td>for Goppa codes $\Gamma(\mathcal{L}, G)$ with $\mathcal{L}$ and $G$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile III</th>
<th>Profile IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_p \approx L \cdot \mathcal{P}$</td>
<td>$H_p^T = \mathcal{P}^{-1} \cdot \mathcal{H}^T$</td>
</tr>
<tr>
<td>Polynomial operations</td>
<td>$S = \tilde{C} \cdot H_p^T$</td>
</tr>
<tr>
<td>for Goppa codes $\Gamma(\mathcal{L}, G)$ with $L_p$ and $G$</td>
<td></td>
</tr>
</tbody>
</table>
'Simple' permutation

Example

\[ \tilde{C} \]

\[ \begin{array}{cccccc}
1 & 2 & \cdots & j & \cdots & n \\
\end{array} \]

\[ \tilde{C}_p \]

\[ \begin{array}{cccccc}
1 & 2 & \cdots & i & \cdots & n \\
\end{array} \]
'Simple' permutation
Algorithm

**Inputs:** Private permutation matrix $\mathcal{P}^{-1} \in \mathcal{M}_{n,n}(\mathbb{F}_2)$ represented by a lookup table $t^{\mathcal{P}^{-1}}$, ciphertext $\tilde{C} \in \mathbb{F}_2^n$.

**For** $i = 0$ to $n - 1$

\[ j = t^i_{\mathcal{P}^{-1}} \]

\[ \tilde{C}_{\mathcal{P}i} = \tilde{C}_j \]

**Endfor**

**Return** $\tilde{C}_p$.

**Output:** Permuted ciphertext $\tilde{C}_p \in \mathbb{F}_2^n$. 
'Secure' permutation [STMOS08]\(^{14}\)

**Algorithm**

**Inputs:** Private permutation matrix \( \mathcal{P}^{-1} \in \mathcal{M}_{n,n}(\mathbb{F}_2) \) represented by a lookup table \( t^{\mathcal{P}^{-1}} \), ciphertext \( \tilde{C} \in \mathbb{F}_2^n \).

1. **For** \( i = 0 \) **to** \( n - 1 \)
2. \( j = t^{\mathcal{P}^{-1}}_i \)
3. \( \tilde{C}_{p_i} = 0 \)
4. **For** \( h = 0 \) **to** \( n - 1 \)
5. \( k = \tilde{C}_{p_i} \)
6. \( \mu = \tilde{C}_h \)
7. \( s = j \oplus h \)
8. \( s \triangleright= s \gg 1 \)
9. \( s \triangleright= s \gg 2 \)
10. \( s \triangleright= s \gg 4 \)
11. \( s \triangleright= s \gg 8 \)
12. \( s \triangleright= s \gg 16 \)
13. \( s \& = 1 \)
14. \( s = \sim (s - 1) \)
15. \( \tilde{C}_{p_i} = (s \& k) \mid ((\sim s) \& \mu) \)
16. **Endfor**
17. **Endfor**
18. **Return** \( \tilde{C}_p \)

**Output:** Permuted ciphertext \( \tilde{C}_p \in \mathbb{F}_2^n \).

---

'Secure' permutation [STMOS08]

Examples

<table>
<thead>
<tr>
<th>Steps</th>
<th>Test hypotheses</th>
</tr>
</thead>
<tbody>
<tr>
<td>7: $s = j \oplus h$</td>
<td>$100 \ldots 0$</td>
</tr>
<tr>
<td></td>
<td>$\underbrace{00 \ldots 0}_{31}$</td>
</tr>
<tr>
<td>8: $s</td>
<td>\leftarrow s \gg 1$</td>
</tr>
<tr>
<td></td>
<td>$\underbrace{00 \ldots 0}_{31}$</td>
</tr>
<tr>
<td>9: $s</td>
<td>\leftarrow s \gg 2$</td>
</tr>
<tr>
<td></td>
<td>$\underbrace{00 \ldots 0}_{31}$</td>
</tr>
<tr>
<td>10: $s</td>
<td>\leftarrow s \gg 4$</td>
</tr>
<tr>
<td></td>
<td>$\underbrace{00 \ldots 0}_{31}$</td>
</tr>
<tr>
<td>11: $s</td>
<td>\leftarrow s \gg 8$</td>
</tr>
<tr>
<td></td>
<td>$\underbrace{00 \ldots 0}_{31}$</td>
</tr>
<tr>
<td>12: $s</td>
<td>\leftarrow s \gg 16$</td>
</tr>
<tr>
<td></td>
<td>$\underbrace{00 \ldots 0}_{31}$</td>
</tr>
<tr>
<td>13: $s &amp; = 1$</td>
<td>$00 \ldots 0$</td>
</tr>
<tr>
<td></td>
<td>$\underbrace{00 \ldots 0}_{31}$</td>
</tr>
<tr>
<td>14: $s = \sim (s - 1)$</td>
<td>$11 \ldots 1$</td>
</tr>
<tr>
<td></td>
<td>$\underbrace{00 \ldots 0}_{31}$</td>
</tr>
</tbody>
</table>
Weakness [PRDCF16]^{15}

Leakage Step 15:

\[ \tilde{C}_{p_i} = \begin{cases} (s \& k) & \text{true only if } s = 11...1 \\ \text{else false} & \end{cases} \quad \begin{cases} ((\sim s) \& \mu) & \text{true only if } s = 00...0 \\ \text{else false} & \end{cases} \]

Giving:

\[ k = 0 \quad k = \tilde{C}_j \]

---

DPA on the ciphertext permutation [PRDCF16]

Known to attacker

1st measurement:
1 1 0 1

2nd measurement:
0 1 1 0

3rd measurement:
1 0 0 0

4th measurement:
0 1 0 1

Secret data

Permutation matrix

P⁻¹

Input random ciphertexts (CCA)

Leakage

Power consumption traces

Reconstruction of the P matrix

CA for 1st bit:

CA for 2nd bit:

CA for 3rd bit:

CA for 4th bit:

Correlation analyses (CAs) for each bit

Reconstructed P⁻¹ matrix by analysing CAs

Known to attacker

Permutation matrix

Bit permuted ciphertexts

DPA

Power consumption Attack

Traces

P⁻¹* =

0 0 1 0
1 0 0 0
0 0 1 1
0 1 0 0
Traces analysis [PRDCF16]

- Apply a Hamming weight of individual bits leakage model: $H_i \in \{0, 1\}$,
- Use correlation coefficient to test our hypotheses compared with measurements,
- Good hypothesis if the coefficient is (almost) 1 or -1,
- Average of 500 traces per ciphertext hypothesis to avoid noise,
- Chosen ciphertexts as every vectors of weight 1.
Trace example [PRDCF16]

Correlation peaks for 15th bit permuted to position 41 of 64

![Graph showing correlation coefficient over samples at 250 MS/s](image)
Countermeasure [PRDCF16]

Algorithm

**Inputs:** Private permutation matrix $\mathcal{P}^{-1} \in \mathcal{M}_{n,n}(\mathbb{F}_2)$ represented by a lookup table $t^{\mathcal{P}^{-1}}$, ciphertext $\tilde{\mathcal{C}} \in \mathbb{F}_2^n$ and private generator matrix.

1. Randomly choose $B \in \Gamma(\mathcal{L}, G)$
2. $B_p = B \cdot \mathcal{P}$
3. $\tilde{\mathcal{C}}' = \tilde{\mathcal{C}} \oplus B_p$
4. For $i = 0$ to $n - 1$
5. $j = t_i^{\mathcal{P}^{-1}}$
6. $\tilde{\mathcal{C}}_{p_i}' = 0$
7. For $h = 0$ to $n - 1$
8. $k = \tilde{\mathcal{C}}_{p_i}'$
9. $\mu = \tilde{\mathcal{C}}_h'$
10. $s = j \oplus h$
11. $s \vdash s \gg 1$
12. $s \vdash s \gg 2$
13. $s \vdash s \gg 4$
14. $s \vdash s \gg 8$
15. $s \vdash s \gg 16$
16. $s \& = 1$
17. $s = \sim (s - 1)$
18. $\tilde{\mathcal{C}}_{p_i}' = (s \& k) \mid ((\sim s) \& \mu)$
19. Endfor
20. Endfor
21. Return $\tilde{\mathcal{C}}_p'$

**Output:** Permuted ciphertext $\tilde{\mathcal{C}}_p' \in \mathbb{F}_2^n$ masked by a codeword.
Countermeasure [PRDCF16]

Scheme

Known to attacker

Input random ciphertexts (CCA)

1\textsuperscript{st} measurement: \begin{equation*} \begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix} \end{equation*}

2\textsuperscript{nd} measurement: \begin{equation*} \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \end{equation*}

3\textsuperscript{rd} measurement: \begin{equation*} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \end{equation*}

4\textsuperscript{th} measurement: \begin{equation*} \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \end{equation*}

Secret data

\begin{equation*} \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} \end{equation*}

\begin{equation*} \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \end{equation*}

\begin{equation*} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \end{equation*}

\begin{equation*} \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} \end{equation*}

\begin{equation*} \begin{bmatrix} pB_1 \ pB_2 \ pB_3 \ pB_4 \end{bmatrix} \end{equation*}

Modified Goppa codewords for all four ciphertexts

Bit permuted ciphertexts with added codewords
Countermeasure [PRDCF16]

Main idea

From masked ciphertext to masked permuted ciphertext:

\[
\tilde{C}_p' = \tilde{C}' \cdot \mathcal{P}^{-1} = (\tilde{C} \oplus B_p) \cdot \mathcal{P}^{-1} = \tilde{C} \cdot \mathcal{P}^{-1} \oplus (B \cdot \mathcal{P}) \cdot \mathcal{P}^{-1} = \tilde{C}_p \oplus B.
\]

From masked permuted ciphertext to the same syndrome than non-masked ciphertext:

\[
S = \tilde{C}_p' \cdot \mathcal{H}^T = (\tilde{C}_p \oplus B) \cdot \mathcal{H}^T = \tilde{C}_p \cdot \mathcal{H}^T \oplus B \cdot \mathcal{H}^T = 0 = \tilde{C}_p \cdot \mathcal{H}^T.
\]
Countermeasure [PRDCF16]

Trace example
Outline

McEliece cryptosystem

Timing Attack

Power consumption Attack

Conclusion
Conclusion

Two Chosen Ciphertext Attacks targeting the private permutation in the McEliece cryptosystem.

Timing attack (TA):

- analysis of both EEA calls in the Patterson’s algorithm during the McEliece decryption: syndrome inversion and error locator determination,
- study two previous attacks,
- combining both attacks to get more information,
- countermeasure proposal (by adding computations in the EEA),
- those TA attacks are depending on the code structure, so useless for others linear codes than Goppa codes.
Conclusion

Power consumption attack (PA):

- analysis of the two first steps in the McEliece decryption: ciphertext permutation and syndrome computation,
- SPA against the syndrome computation implemented on a microcontroller,
- masking countermeasure to avoid branches,
- DPA against a 'secure' permutation algorithm implemented on a microcontroller,
- masking countermeasure (with $n$ more bits and not a huge amount of additional computations),
- both PA attacks are not depending on the code structure, so possible for others linear codes than Goppa codes.
## Perspectives

Four profiles of implementation [HMP10] for ciphertext permutation and syndrome computation

<table>
<thead>
<tr>
<th>Profile I</th>
<th>Profile II</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tilde{C}_p = \tilde{C} \cdot P^{-1} )</td>
<td>( \tilde{C}_p = \tilde{C} \cdot P^{-1} )</td>
</tr>
<tr>
<td>Polynomial operations</td>
<td>( S = \tilde{C}_p \cdot H^T )</td>
</tr>
<tr>
<td>for Goppa codes ( \Gamma(\mathcal{L}, G) ) with ( \mathcal{L} ) and ( G )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile III</th>
<th>Profile IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{L}_p \approx \mathcal{L} \cdot P )</td>
<td>( \mathcal{H}^T_p = P^{-1} \cdot H^T )</td>
</tr>
<tr>
<td>Polynomial operations</td>
<td>( S = \tilde{C} \cdot H^T_p )</td>
</tr>
<tr>
<td>for Goppa codes ( \Gamma(\mathcal{L}, G) ) with ( \mathcal{L}_p ) and ( G )</td>
<td></td>
</tr>
</tbody>
</table>

Best choice ?!
Perspectives

- Make a power analysis and try a fault injection attack for the countermeasure against the TA,
- Try a higher-order power consumption or a template attack for the countermeasure against PA,
- Goppa polynomial recovering after getting the private permutation matrix and knowing the support elements order in the McEliece public key cryptosystem.
Side-channel analysis in code-based cryptography

Tania RICHMOND

Thank you for your attention!