Low-Complexity Power Analysis Countermeasure for Resource-Constrained Embedded McEliece Implementation

Martin Petrvalský, **Tania Richmond**, Miloš Drutarovský, Pierre-Louis Cayrel and Viktor Fischer

IMATH Laboratory
University of Toulon

CryptArchi
Smolenice, June 19, 2017

Outline

Outline

McEliece Cryptosystem

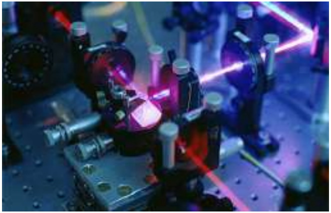
Power Consumption Attacks

Conclusion

Post-quantum cryptography

- ⋆ Code-based cryptography
- ⋆ Lattice-based cryptography
- ⋆ Hash-based cryptography
- ⋆ Multivariate-based cryptography
- ⋆ Isogeny-based cryptography

No solving in polynomial time,
contrary to number theory problems [Sho97][1]



[1] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing, 26(5), pp. 1484-1509, 1997.

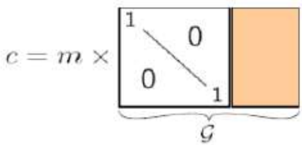
Linear code

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 $\mathscr{C}$ of length n and dimension k is a k-dimensional subspace of $\mathbb{F}_q^n$.

Definition (Generator matrix)

Let $\mathscr{C}$ be a $[n, k]_q$-linear code. Let $\mathcal{G} \in \mathbb{F}_q^{k \times n}$. We call $\mathcal{G}$ a generator matrix of $\mathscr{C}$ iff $\mathcal{G}$-rows are basis vectors of $\mathscr{C}$.



Linear code

Definition (Parity-check matrix)

Let $\mathscr{C}$ be a $[n, k]_q$-linear code. Let $\mathcal{H} \in \mathbb{F}_q^{(n-k)\times n}$. We call $\mathcal{H}$ a parity-check matrix of $\mathscr{C}$ if:

$$\forall C \in \mathbb{F}_q^n, \qquad C \in \mathscr{C} \Leftrightarrow C \cdot \mathcal{H}^T = 0 \quad (\in \mathbb{F}_q^{n-k})$$

Definition (Error-correction capacity)

Let $\mathscr{C}$ be a $[n, k, d]$-linear code and C a codeword. We call t the maximum weight of a corrigible error vector added to C:

$$\tilde{C} = C + E \quad \begin{cases} \tilde{C} \text{ corrigible if } w_H(E) \leqslant t \\ \tilde{C} \text{ incorrigible if } w_H(E) > t \end{cases}$$

Syndrome Decoding (SD) problem [BMcEvT78] [2]

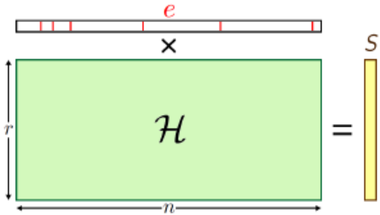
Inputs

$\mathcal{H}$ matrix of size $r \times n$,
S binary vector of length r,
t interger.

$r = n - k$
S is called syndrome.

Problem

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



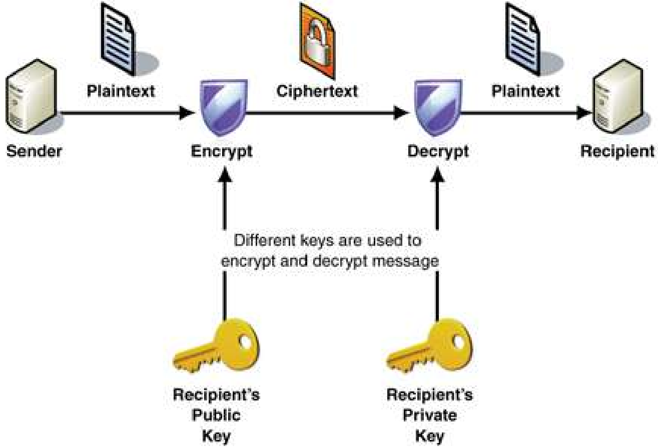
Theorem

SD is NP-complete.

[2] E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Transactions on Information Theory, 1978.

Public-Key Cryptosystem (PKC)

[DH76][3]



[3] W. Diffie and M. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory 1976.

McEliece PKC

Proposed in [McE78] [4].

Key generation: Given a (binary) t-error correcting $[n, k, d]$-linear code and a generator matrix $\mathcal{G}$.

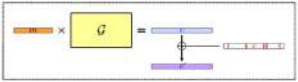
sk: $(\mathcal{Q}, \mathcal{G}, \mathcal{P})$

pk: $(\mathcal{G}', n, t)$

$$\mathcal{G}' = \mathcal{Q} \cdot \mathcal{G} \cdot \mathcal{P}$$

Encryption:

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



Decryption:

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

[4] R. J. McEliece, *A public-key cryptosystem based on algebraic coding theory*, California Inst. Technol., Pasadena, CA, Tech. Rep. 44, 1978.

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]

[5] P. C. Kocher, *Timing attacks on implementations of Diffie- Hellman, RSA, DSS, and other systems*, CRYPTO'96, Springer, LNCS, vol. 1109, pp. 104-113, 1996.

Outline

McEliece Cryptosystem

Power Consumption Attacks

Conclusion

How to implement the McEliece PKC?

Key generation: Given a (binary) t-error correcting $[n, k, d]$-linear code and a generator matrix $\mathcal{G}$.

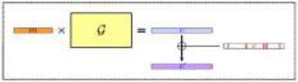
sk: $(\mathcal{Q}, \mathcal{G}, \mathcal{P})$

pk: $(\mathcal{G}', n, t)$

$$\mathcal{G}' = \mathcal{Q} \cdot \mathcal{G} \cdot \mathcal{P}$$

Encryption:

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



Decryption:

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

Four profiles of implementation [HMP10] [6]

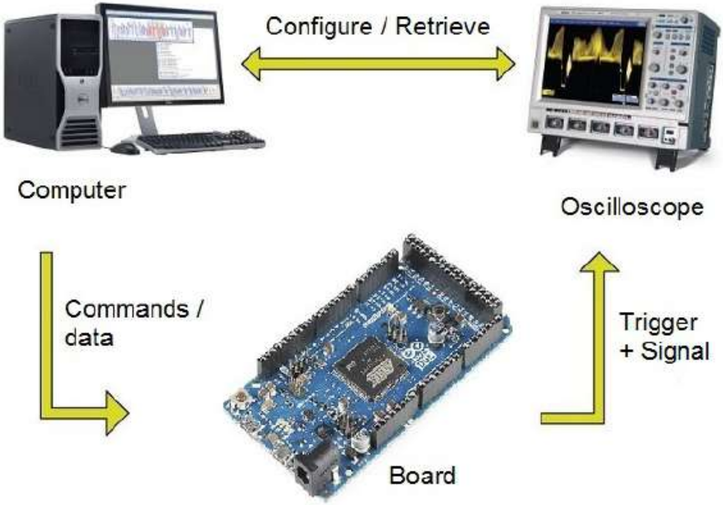
for ciphertext permutation and syndrome computation

Profile I	Profile II
$\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$ Polynomial operations for Goppa codes $\Gamma(\mathscr{L}, G)$ with $\mathscr{L}$ and G	$\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$ $S = \tilde{C}_p \cdot \mathcal{H}^T$
Profile III	**Profile IV**
$\mathscr{L}_p \approx \mathscr{L} \cdot \mathcal{P}$ Polynomial operations for Goppa codes $\Gamma(\mathscr{L}, G)$ with $\mathscr{L}_p$ and G	$\mathcal{H}_p^T = \mathcal{P}^{-1} \cdot \mathcal{H}^T$ $S = \tilde{C} \cdot \mathcal{H}_p^T$

[6] S. Heyse, A. Moradi and C. Paar, *Practical Power Analysis Attacks on Software Implementations of McEliece*, PQCrypto 2010.

UNIVERSITÉ DE TOULON

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 the key equation
4. Error position finding

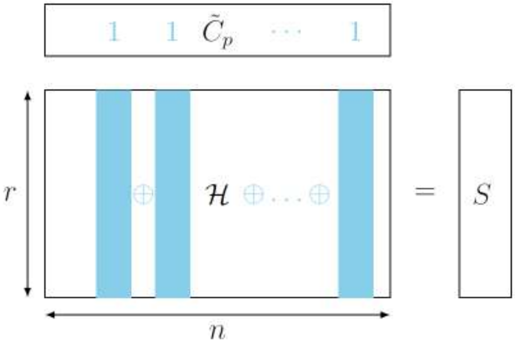
Four profiles of implementation [HMP10]

for ciphertext permutation and syndrome computation

Profile I	Profile II
$\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$ Polynomial operations for Goppa codes $\Gamma(\mathscr{L}, G)$ with $\mathscr{L}$ and G	$\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$ $S = \tilde{C}_p \cdot \mathcal{H}^T$
Profile III	**Profile IV**
$\mathscr{L}_p \approx \mathscr{L} \cdot \mathcal{P}$ Polynomial operations for Goppa codes $\Gamma(\mathscr{L}, G)$ with $\mathscr{L}_p$ and G	$\mathcal{H}_p^T = \mathcal{P}^{-1} \cdot \mathcal{H}^T$ $S = \tilde{C} \cdot \mathcal{H}_p^T$

Syndrome computation

Scheme



Syndrome computation

Algorithm

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

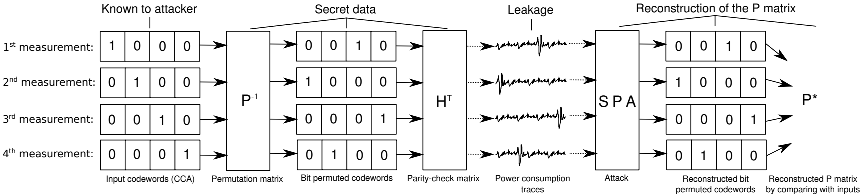
For $i = 1$ **to** n
  If $\tilde{C}_{p_i} = 1$
    $S = S \oplus \mathcal{H}_i$
  EndIf
EndFor
Return S.

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

SPA on the syndrome computation [PRDCF15][7]

Toy example

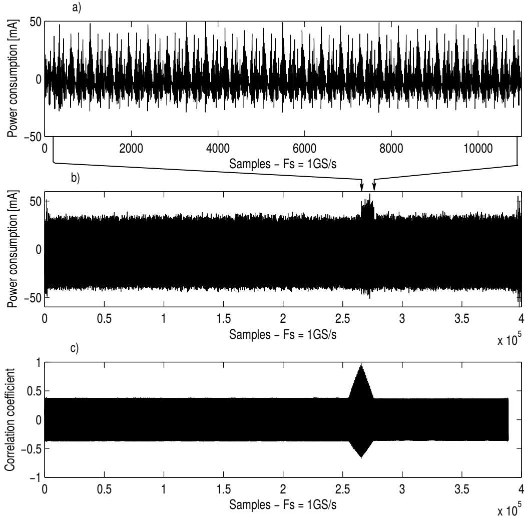
SPA with chosen single-one ciphertexts



Chosen Ciphertext Attack (CCA)

[7] 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.

Trace example [PRDCF15]



Countermeasure [PRDCF15]

First algorithm

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

$words = r/sizeof(S)$ *Required number of bytes to store* S

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

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

For $j = 1$ **to** $words$

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

EndFor

EndFor

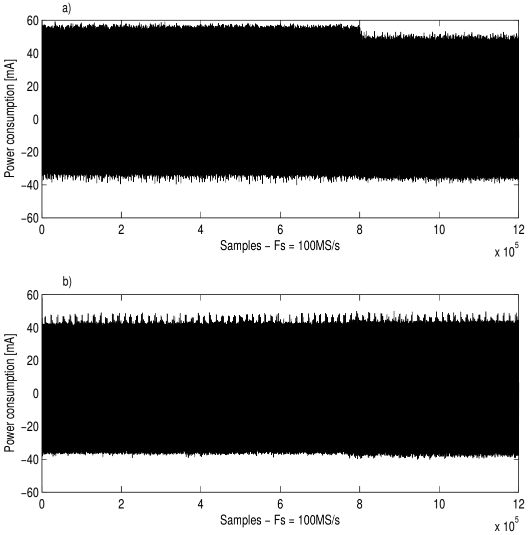
Return S.

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

UNIVERSITÉ DE TOULON

Countermeasure [PRDCF15]

Trace example



Countermeasure [PRDCF15]

Second algorithm

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

$\quad$ *words* $= r/$*sizeof*(S) *Required number of bytes to store* S

Syndrome masking

$\quad$ **For** $j = 1$ **to** *words*

$\quad\quad$ $S_j = S_j \ \& \ \mathrm{0xAAAA}$

$\quad$ **EndFor**

Syndrome computation

$\quad$ **For** $i = 1$ **to** n

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

$\quad\quad$ **For** $j = 1$ **to** *words*

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

$\quad\quad$ **EndFor**

$\quad$ **EndFor**

Syndrome unmasking

$\quad$ **For** $j = 1$ **to** *words*

$\quad\quad$ $S_j = S_j \ \& \ \mathrm{0xAAAA}$

$\quad$ **EndFor**

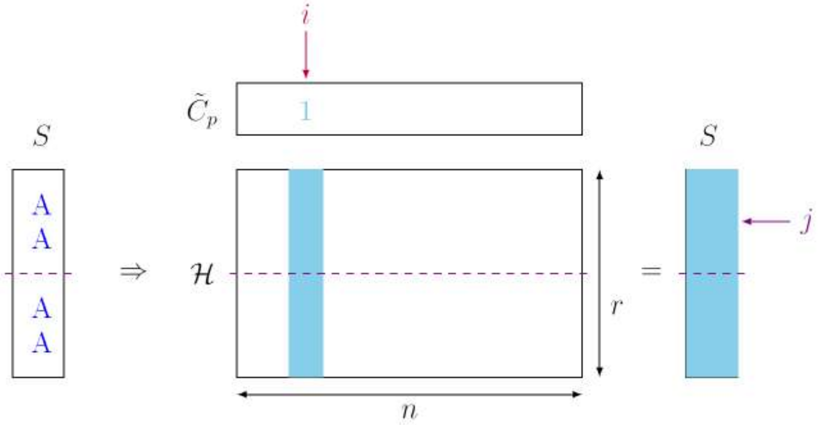
$\quad$ **Return** S.

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

UNIVERSITÉ DE TOULON

Syndrome computation with countermeasure [PRDCF15]

Scheme



Differential Power Analysis (DPA) on the ciphertext permutation

For example vector-matrix product

Decryption:

1. Ciphertext permutation : $\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$
2. Syndrome computation
3. Solving the key equation
4. Error position finding

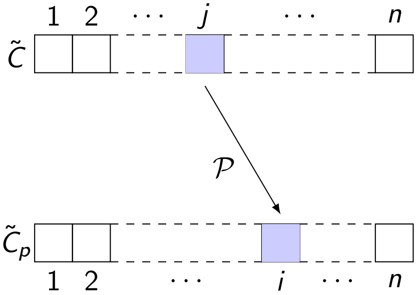
Four profiles of implementation [HMP10]

for ciphertext permutation and syndrome computation

Profile I $\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$ Polynomial operations for Goppa codes $\Gamma(\mathscr{L}, G)$ with $\mathscr{L}$ and G	Profile II $\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$ $S = \tilde{C}_p \cdot \mathcal{H}^T$
Profile III $\mathscr{L}_p \approx \mathscr{L} \cdot \mathcal{P}$ Polynomial operations for Goppa codes $\Gamma(\mathscr{L}, G)$ with $\mathscr{L}_p$ and G	Profile IV $\mathcal{H}_p^T = \mathcal{P}^{-1} \cdot \mathcal{H}^T$ $S = \tilde{C} \cdot \mathcal{H}_p^T$

Straightforward permutation

Example



Straightforward permutation

Algorithm

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

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

$$\begin{aligned} j &= t_i^{\mathcal{P}^{-1}} \\ \tilde{C}_{p_i} &= \tilde{C}_j \end{aligned}$$

Endfor

Return $\tilde{C}_p$.

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

'Secure' permutation [STMOS08][8]

Algorithm

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

For $i = 0$ **to** $n-1$
  $j = t_i^{\mathcal{P}^{-1}}$
  $\tilde{C}_{p_i} = 0$
  For $h = 0$ **to** $n-1$
    $k = \tilde{C}_{p_i}$
    $\mu = \tilde{C}_h$
    $s = j \oplus h$
    $s \mid= s \gg 1$
    $s \mid= s \gg 2$
    $s \mid= s \gg 4$
    $s \mid= s \gg 8$
    $s \mid= s \gg 16$
    $s \mathrel{\&}= 1$
    $s = \sim (s-1)$
    $\tilde{C}_{p_i} = (s \mathbin{\&} k) \mid ((\sim s) \mathbin{\&} \mu)$
  Endfor
Endfor
Return $\tilde{C}_p$

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

[8] F. Strenzke, E. Tews, H. G. Molter, R. Overbeck and A. Shoufan, *Side Channels in the McEliece PKC*, PQCrypto 2008.

'Secure' permutation [STMOS08]

Examples

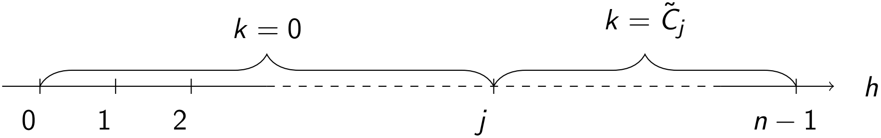
Steps	Test hypotheses			
7: $s = j \oplus h$	$1\,\underbrace{00\ldots0}_{31}$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{11\ldots1}_{32}$	$\underbrace{00\ldots0}_{32}$
8: $s \mid= s \gg 1$	$11\,\underbrace{00\ldots0}_{30}$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{11\ldots1}_{32}$	$\underbrace{00\ldots0}_{32}$
9: $s \mid= s \gg 2$	$1111\,\underbrace{00\ldots0}_{28}$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{11\ldots1}_{32}$	$\underbrace{00\ldots0}_{32}$
10: $s \mid= s \gg 4$	$\underbrace{11\ldots1}_{8}\underbrace{00\ldots0}_{24}$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{11\ldots1}_{32}$	$\underbrace{00\ldots0}_{32}$
11: $s \mid= s \gg 8$	$\underbrace{11\ldots1}_{16}\underbrace{00\ldots0}_{16}$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{11\ldots1}_{32}$	$\underbrace{00\ldots0}_{32}$
12: $s \mid= s \gg 16$	$\underbrace{11\ldots1}_{32}$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{11\ldots1}_{32}$	$\underbrace{00\ldots0}_{32}$
13: $s \mathrel{\&}= 1$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{00\ldots0}_{31}1$	$\underbrace{00\ldots0}_{32}$
14: $s = \sim(s-1)$	$\underbrace{11\ldots1}_{32}$	$\underbrace{11\ldots1}_{32}$	$\underbrace{11\ldots1}_{32}$	$\underbrace{00\ldots0}_{32}$

Weakness [PRDCF16][9]

Leakage Step 15:

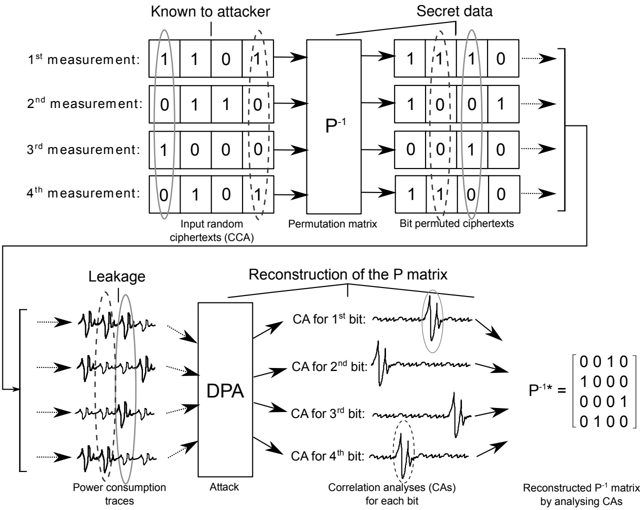
$$\tilde{C}_{p_i} = \underbrace{(s \ \& \ k)}_{\substack{\text{true only if } s=11\ldots1 \\ \text{else false}}} \quad | \quad \underbrace{((\sim s) \ \& \ \mu)}_{\substack{\text{true only if } s=00\ldots0 \\ \text{else false}}}$$

Giving:

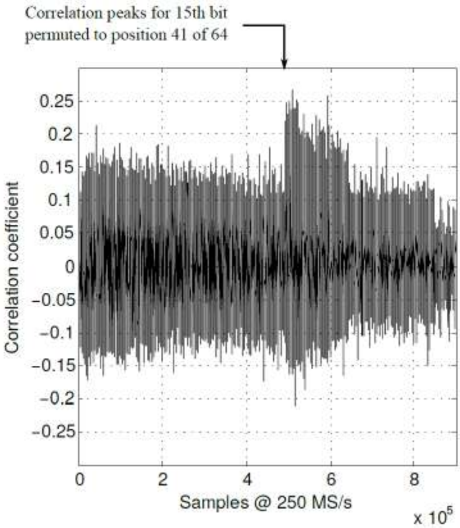


[9] M. Petrvalský, **T. Richmond**, M. Drutarovský, P.-L. Cayrel and V. Fischer, *Differential power analysis attack on the secure bit permutation in the McEliece cryptosystem*, IEEE Radioelektronika 2016.

DPA on the ciphertext permutation [PRDCF16]



Trace example [PRDCF16]



Countermeasure [PRDCF16]

Algorithm

Inputs: Private permutation matrix $\mathcal{P}^{-1} \in \mathbb{F}_2^{n \times n}$ represented by a lookup table $t^{\mathcal{P}^{-1}}$, ciphertext $\tilde{C} \in \mathbb{F}_2^n$ and private generator matrix $\mathcal{G}$ of $\Gamma(\mathscr{L}, G)$.

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

Output: Permuted ciphertext $\tilde{C}_p' \in \mathbb{F}_2^n$ masked by a codeword.

Countermeasure [PRDCF16]

Main idea

From masked ciphertext to masked permuted ciphertext:

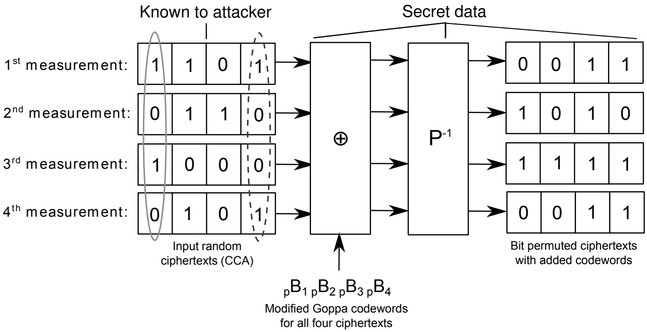
$$\begin{aligned}\tilde{C}'_p &= \tilde{C}' \cdot \mathcal{P}^{-1} \\ &= (\tilde{C} \oplus B_p) \cdot \mathcal{P}^{-1} \\ &= \tilde{C} \cdot \mathcal{P}^{-1} \oplus (B \cdot \mathcal{P}) \cdot \mathcal{P}^{-1} \\ &= \tilde{C}_p \oplus B.\end{aligned}$$

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

$$\begin{aligned}S &= \tilde{C}'_p \cdot \mathcal{H}^T \\ &= (\tilde{C}_p \oplus B) \cdot \mathcal{H}^T \\ &= \tilde{C}_p \cdot \mathcal{H}^T \oplus \underbrace{B \cdot \mathcal{H}^T}_{=0} \\ &= \tilde{C}_p \cdot \mathcal{H}^T.\end{aligned}$$

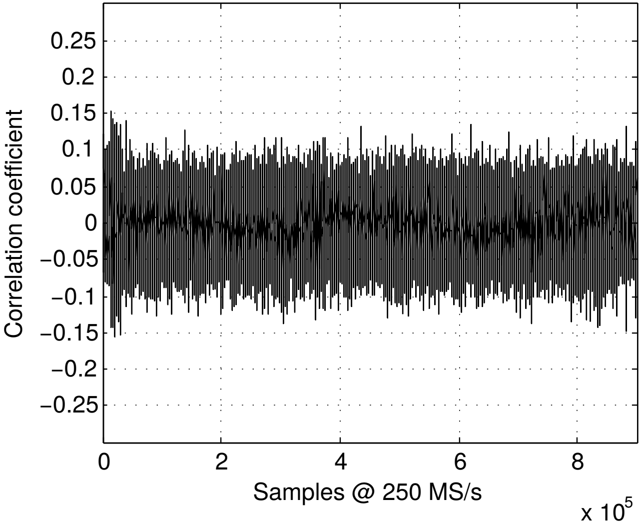
Countermeasure [PRDCF16]

Scheme



Countermeasure [PRDCF16]

Trace example



Outline

McEliece Cryptosystem

Power Consumption Attacks

Conclusion

Conclusion

Two power consumption attacks with chosen single-one ciphertexts targeting the private permutation in the McEliece cryptosystem:

- 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 the '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

Profile I	Profile II
$\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$ Polynomial operations for Goppa codes $\Gamma(\mathscr{L}, G)$ with $\mathscr{L}$ and G	$\tilde{C}_p = \tilde{C} \cdot \mathcal{P}^{-1}$ $S = \tilde{C}_p \cdot \mathcal{H}^T$
Profile III $\mathscr{L}_p \approx \mathscr{L} \cdot \mathcal{P}$ Polynomial operations for Goppa codes $\Gamma(\mathscr{L}, G)$ with $\mathscr{L}_p$ and G	**Profile IV** $\mathcal{H}_p^T = \mathcal{P}^{-1} \cdot \mathcal{H}^T$ $S = \tilde{C} \cdot \mathcal{H}_p^T$

Best choice!

Perspectives

- 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.

Low-Complexity Power Analysis Countermeasure for Resource-Constrained Embedded McEliece Implementation

Tania RICHMOND

Thank you for your attention!

UNIVERSITÉ DE TOULON

Remark on the last presented countermeasure

If we considere that we get the codeword without error at the end of the decoding, then we must keep the mask codeword to unmask, otherwise the error vector to remove it from the received ciphertext.

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.

UNIVERSITÉ DE TOULON

Pearson's correlation coefficient

We used for correlation analyses:

$$r_{H,X}(\eta) = \frac{\sum_{i=1}^{N}[(X_i(\eta) - \bar{X}(\eta))(H_i - \bar{H})]}{\sqrt{\sum_{i=1}^{N}[X_i(\eta) - \bar{X}(\eta)]^2 \sum_{i=1}^{N}(H_i - \bar{H})^2}}$$

where $r_{H,X}(\eta)$ is the Pearson's correlation coefficient for η-th sample (measured during execution of the cryptographic algorithm), N is a number of measured traces, $X_i(\eta)$ is a value of η-th sample measured during i-th measurement (i-th trace), $\bar{X}(\eta)$ is a mean value of corresponding η-th samples (from all traces), H_i is a hypothesis of power consumption for one bit of input data corresponding with i-th measurement (i-th trace) and $\bar{H}$ is a mean value of all hypotheses H_i.

UNIVERSITÉ DE TOULON