MINISTÈRE
DES ARMÉES

Liberté
Égalité
Fraternité

Implementing SCA Countermeasures for FrodoKEM is not Trivial

Jérémy METAIRIE, Cédric MURDICA, Karl TOURNIER

DGA
DIRECTION
GÉNÉRALE
DE L'ARMEMENT

20/11/2024

MINISTÈRE
DES ARMÉES

Liberté
Égalité
Fraternité

Project context

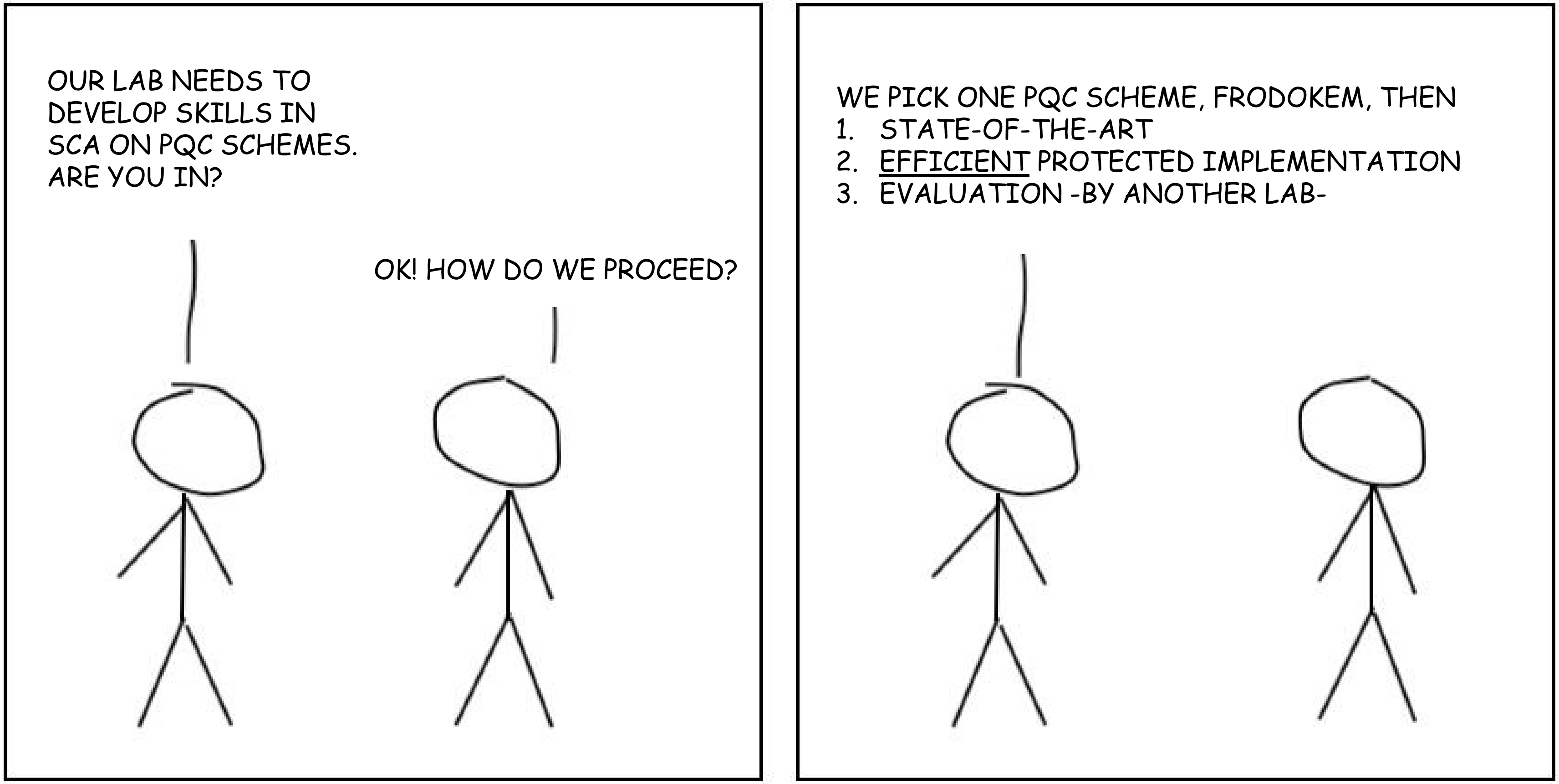
DGA

DIRECTION
GÉNÉRALE
DE L'ARMEMENT

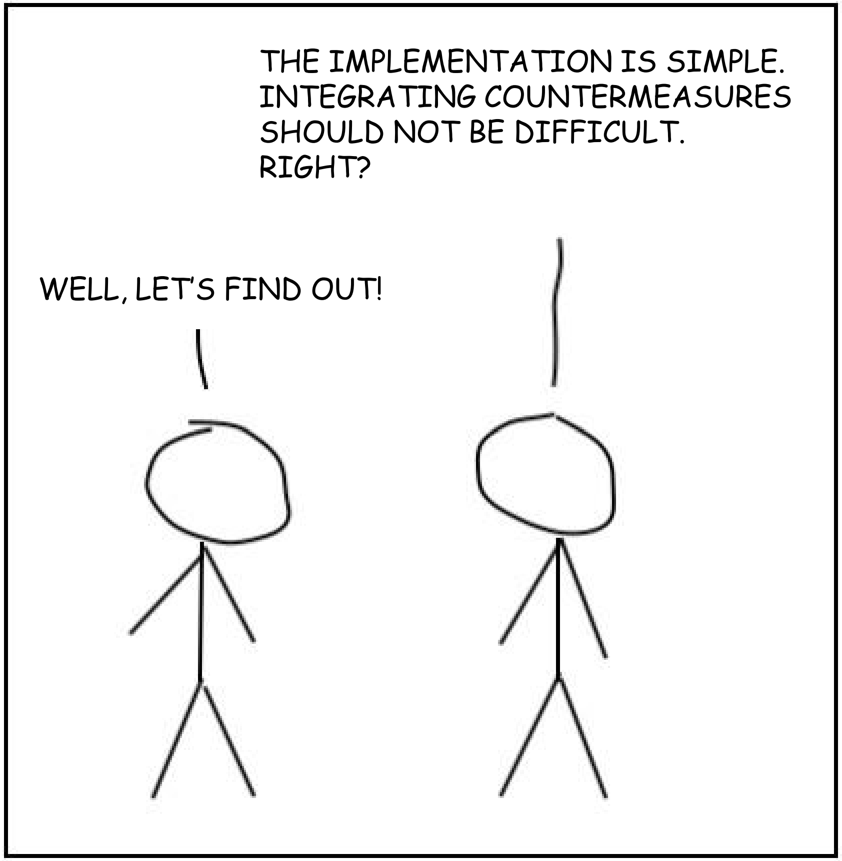
Project context

- Who are we?
  - Cryptography engineers at *DGA Maîtrise de l’Information*
  - Background in Side-Channel Attacks (PhD in SCA on Elliptic Curve Cryptography)

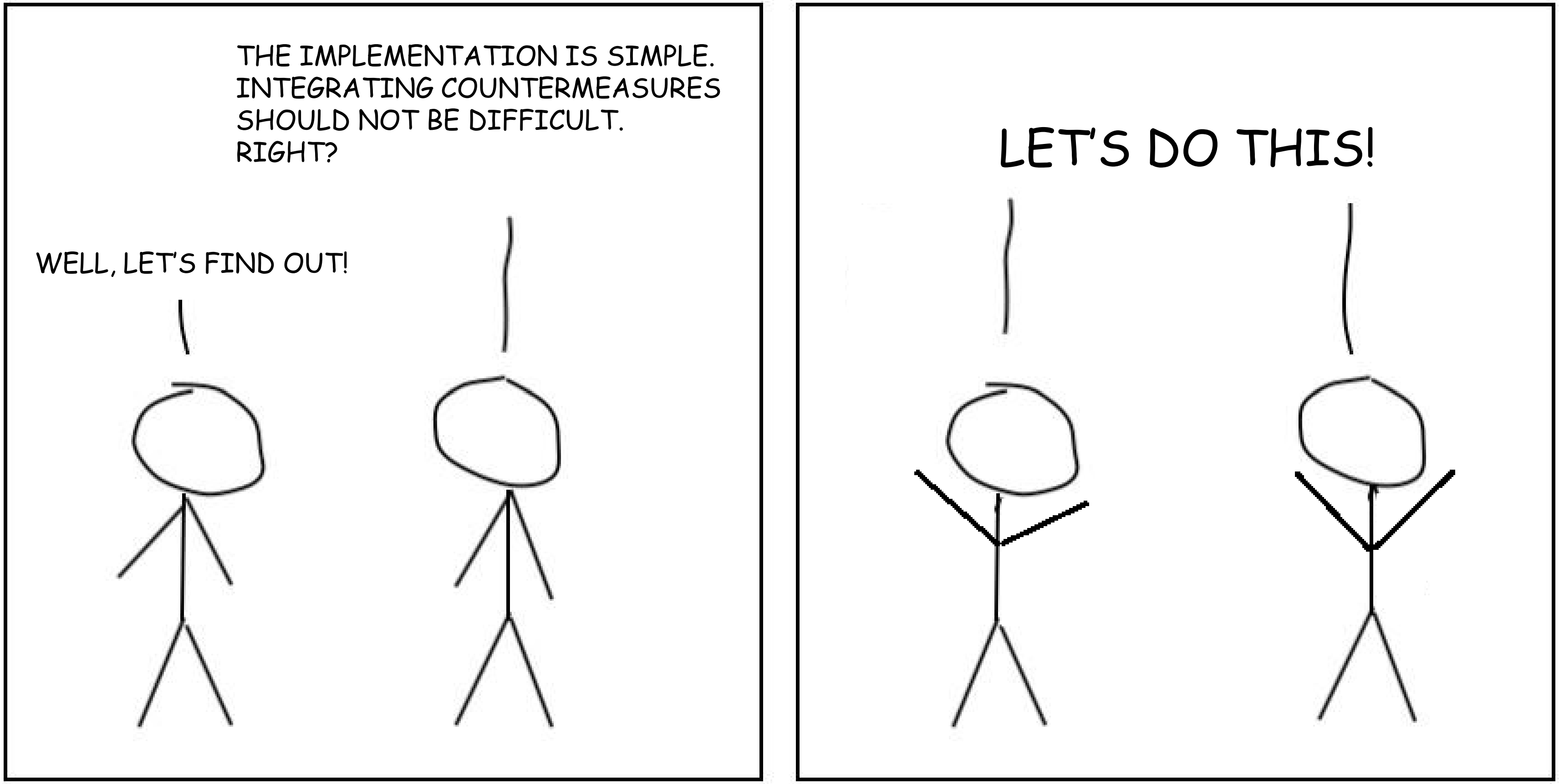
Project context


OUR LAB NEEDS TO DEVELOP SKILLS IN SCA ON PQC SCHEMES. ARE YOU IN?
OK! HOW DO WE PROCEED?
WE PICK ONE PQC SCHEME, FRODOKEM, THEN
1. STATE-OF-THE-ART
2. EFFICIENT PROTECTED IMPLEMENTATION
3. EVALUATION -BY ANOTHER LAB-

Project context


THE IMPLEMENTATION IS SIMPLE. INTEGRATING COUNTERMEASURES SHOULD NOT BE DIFFICULT. RIGHT?
WELL, LET'S FIND OUT!

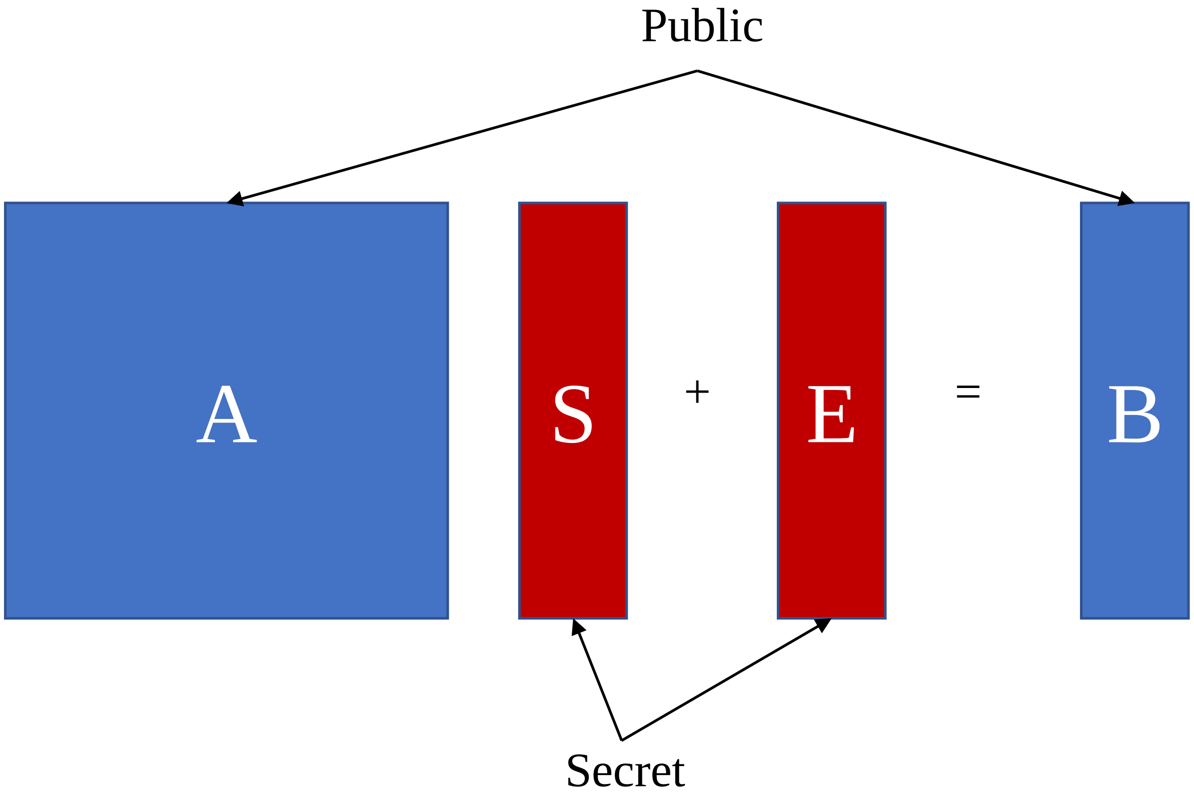
Project context


THE IMPLEMENTATION IS SIMPLE. INTEGRATING COUNTERMEASURES SHOULD NOT BE DIFFICULT. RIGHT?
WELL, LET'S FIND OUT!
LET'S DO THIS!

FrodoKEM

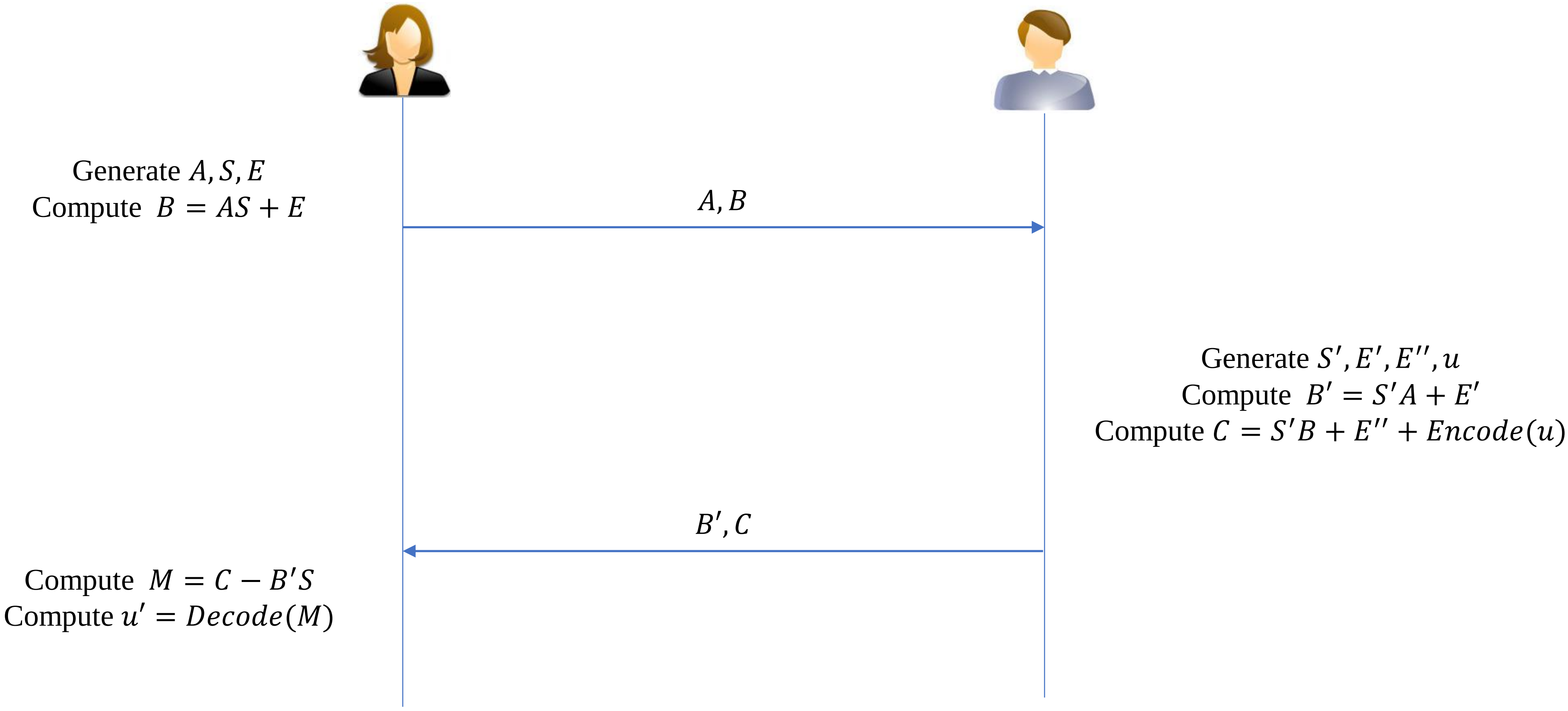
FrodoKEM

Learning With Error (LWE)



FrodoKEM

Encapsulation – Decapsulation (*Simplified*)



Generate A, S, E
Compute $B = AS + E$

A, B

Generate S', E', E'', u
Compute $B' = S'A + E'$
Compute $C = S'B + E'' + Encode(u)$

B', C

Compute $M = C - B'S$
Compute $u' = Decode(M)$

FrodoKEM

Size of elements

$$1344\left\{\overbrace{\begin{pmatrix} a_{0,0} & \cdots & \cdots & a_{0,n-1} \\ a_{1,0} & \ddots & & a_{1,n-1} \\ \vdots & & \ddots & \\ \vdots & & & \\ a_{n-1,0} & & & a_{n-1,n-1} \end{pmatrix}}^{1344}\right. \overbrace{\begin{pmatrix} s_{0,0} & \cdots & s_{0,\bar{n}-1} \\ s_{1,0} & & s_{1,\bar{n}-1} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ s_{n-1,0} & \cdots & s_{n-1,\bar{n}-1} \end{pmatrix}}^{8}$$

- Integers modulo $q = 2^{16}$
- 3.44Mb (= $1344 \times 1344 \times 16$ bits)

FrodoKEM

Generation of A

$$AES(seed_A, 0||j) \longrightarrow \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

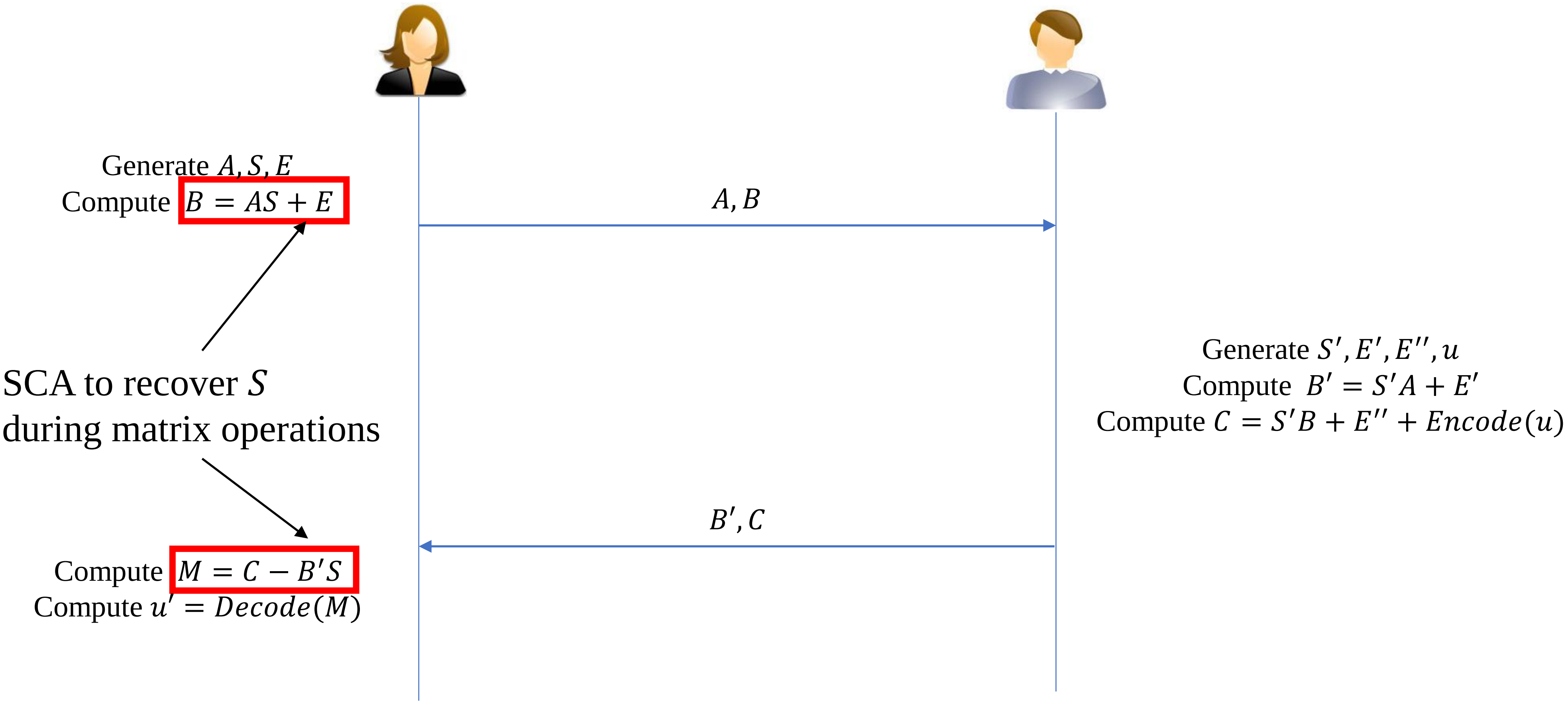
MINISTÈRE
DES ARMÉES

Liberté
Égalité
Fraternité

Horizontal Attack

DGA
DIRECTION
GÉNÉRALE
DE L'ARMEMENT

Horizontal Attack



Generate A, S, E
Compute $\boxed{B = AS + E}$

A, B

SCA to recover S
during matrix operations

Generate S', E', E'', u
Compute $B' = S'A + E'$
Compute $C = S'B + E'' + Encode(u)$

B', C

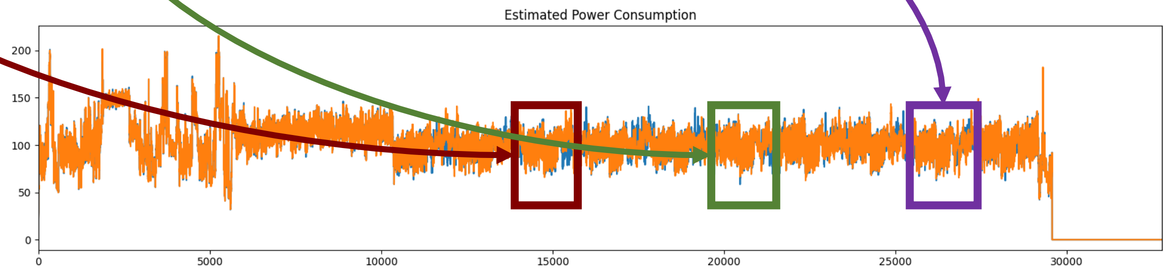
Compute $\boxed{M = C - B'S}$
Compute $u' = Decode(M)$

Horizontal Attack

- Computing the $A \times S$ matrix product:

$\boxed{a_{0,0} \times s_{0,0}}$ $\boxed{a_{1,0} \times s_{1,0}}$ $\boxed{a_{2,0} \times s_{0,0}}$ …

$$\begin{pmatrix} \boxed{\begin{matrix} a_{0,0} \\ a_{1,0} \\ \vdots \\ a_{n-2,0} \\ a_{n-1,0} \end{matrix}} & \cdots \end{pmatrix}$$



MINISTÈRE
DES ARMÉES

Liberté
Égalité
Fraternité

Countermeasures

DGA
DIRECTION
GÉNÉRALE
DE L'ARMEMENT

Countermeasures

- Additive masking – *Not presented here*
  - Not satisfactory: makes the attack harder but does not prevent it

- Multiplicative masking – *Not presented here*
  - Not satisfactory: makes the attack harder but does not prevent it (it could prevent it at an unsatisfactory cost)

- **Shuffling**

Shuffling

- Shuffling the rows
- Shuffling the columns

Shuffling the rows

$$\begin{pmatrix} s_{0,0} & \cdots & s_{0,\bar{n}-1} \\ s_{1,1} & & s_{0,\bar{n}-1} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ s_{n-1,0} & \cdots & s_{n-1,\bar{n}-1} \end{pmatrix}$$

$$AES(seed_A, 0||j) \longrightarrow \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

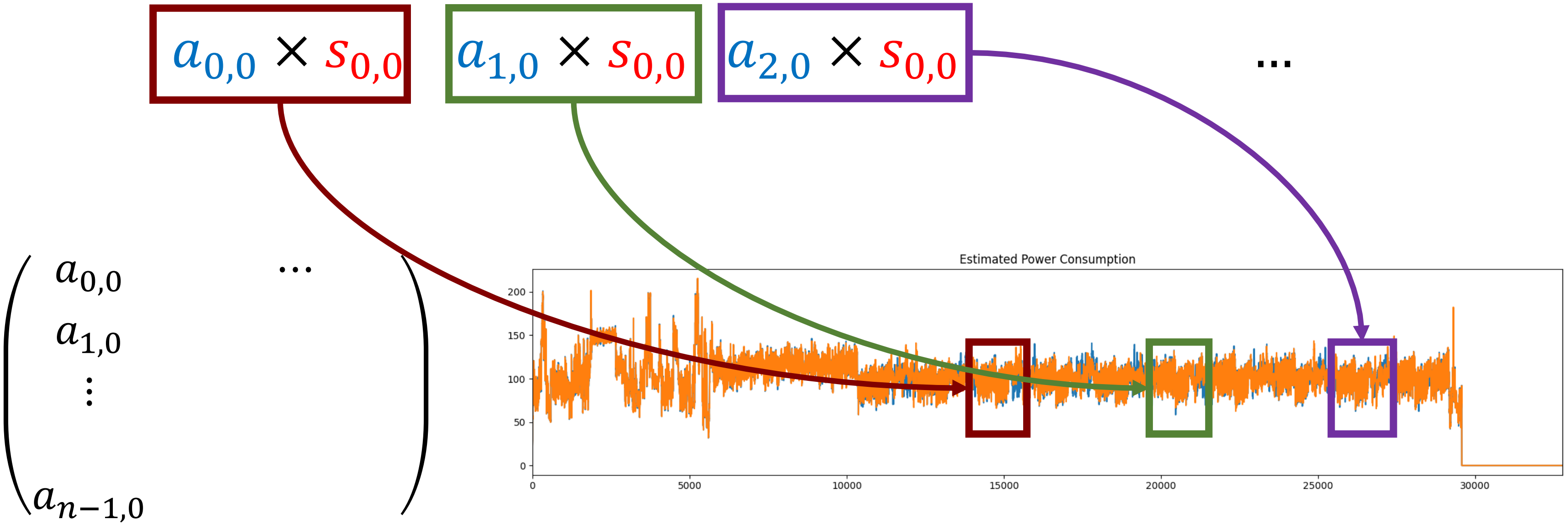
Shuffling the rows

$$\begin{pmatrix} s_{0,0} & \cdots & s_{0,\bar{n}-1} \\ s_{1,1} & & s_{0,\bar{n}-1} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ s_{n-1,0} & \cdots & s_{n-1,\bar{n}-1} \end{pmatrix}$$

$$AES(seed_A, r||j) \longrightarrow \begin{pmatrix} a_{r,0} & a_{r,1} & \cdots & a_{r,n-1} \end{pmatrix}$$

Horizontal Attack

Without rows permutation

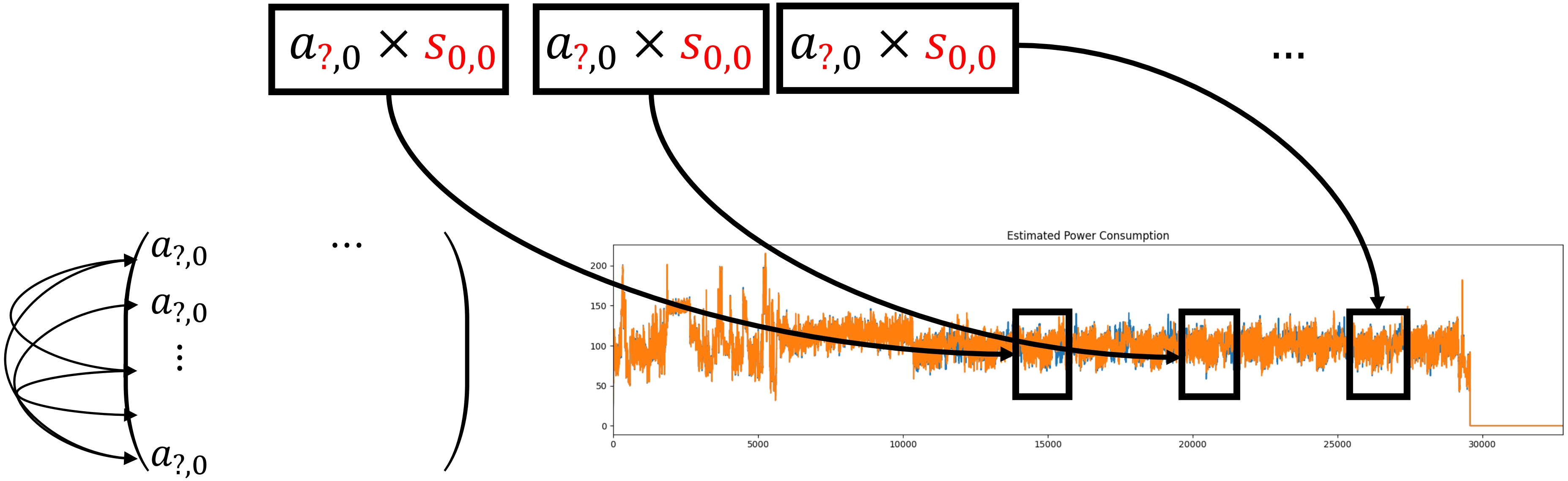


$a_{0,0} \times s_{0,0}$ $a_{1,0} \times s_{0,0}$ $a_{2,0} \times s_{0,0}$ …

$$\begin{pmatrix} a_{0,0} & \cdots \\ a_{1,0} & \\ \vdots & \\ a_{n-1,0} & \end{pmatrix}$$

Horizontal Attack

With rows permutation



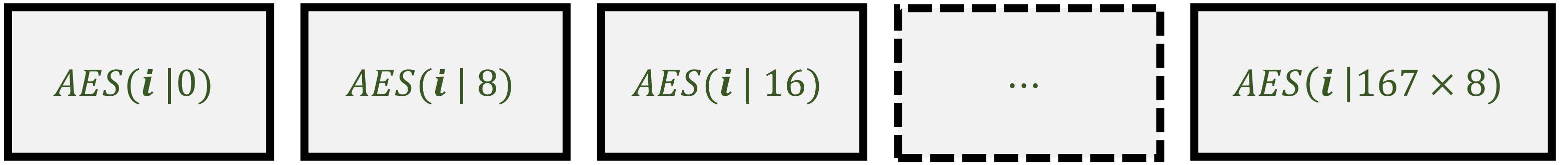
$a_{?,0} \times s_{0,0}$ $a_{?,0} \times s_{0,0}$ $a_{?,0} \times s_{0,0}$ …

$$\begin{pmatrix} a_{?,0} & \cdots \\ a_{?,0} & \\ \vdots & \\ a_{?,0} & \end{pmatrix}$$

Shuffling the rows: Horizontal Attack

- ➢ Shuffling the rows is not secure: we can recover the row index

Rows are generated on the fly based on the $AES(\ i \mid j\ )$ computation



$AES(\boldsymbol{i}\ |0)$ $AES(\boldsymbol{i} \mid 8)$ $AES(\boldsymbol{i} \mid 16)$ $\cdots$ $AES(\boldsymbol{i}\ |167 \times 8)$

- ○ **Key is Publicly Known ☹**
- ○ $\boldsymbol{i \in \{0, \cdots, 1343\}}$ **☹**
- ○ **Up to 168 AES with the same row index ☹**

➔ Should be easy to Recover *i* through SCA*

*It is!

Defeat the AES

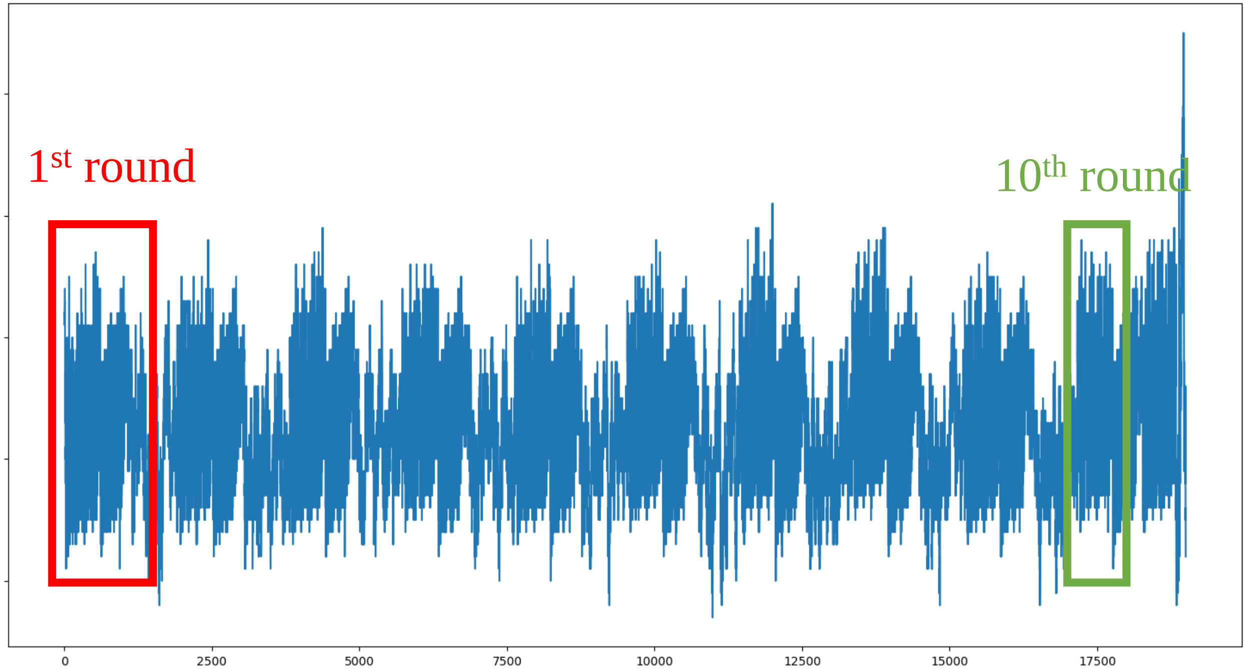
We want to extract i from:

- Tiny-AES
  - By-the-book implementation
  - 18,000 instructions per block
  - With generated traces and real traces (AESPTv2/STM32F411E-DISCO)

- AES from OpenSSL (version 3.3)
  - T-tables based implementation
  - 1,800 instructions per block (10 times as fast as tiny-AES)
  - With generated traces

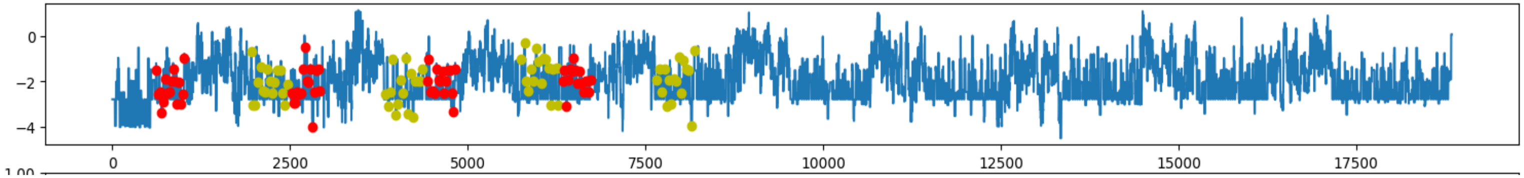
Defeat the AES

How do generated traces look? (Here tinyAES)



Defeat the AES

Leakage Assessment+CPA/Templates



Single trace attack:

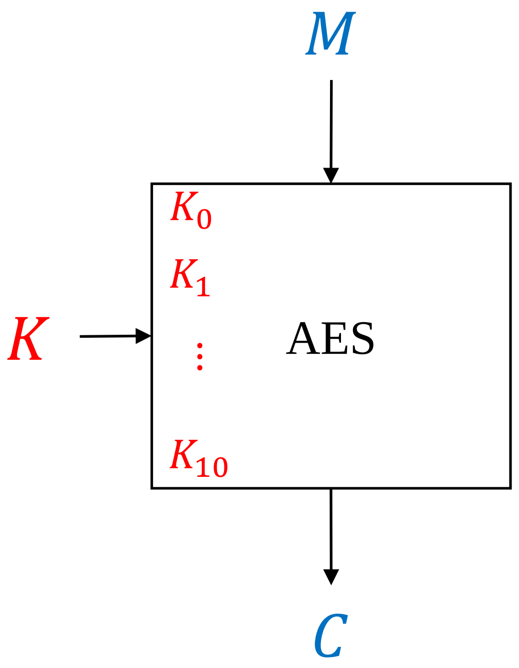
1. Extract POI
2. Correlate POI to Power Consumption Models
3. Highest correlation is the Right Hypothesis:
   - **True** for the tiny-AES with generated traces
   - **True** for the tiny-AES with real traces
   - **Almost True** for the OpenSSL implementation but…

- **Conclusion**: Row index can be recovered

Horizontal attack on AES

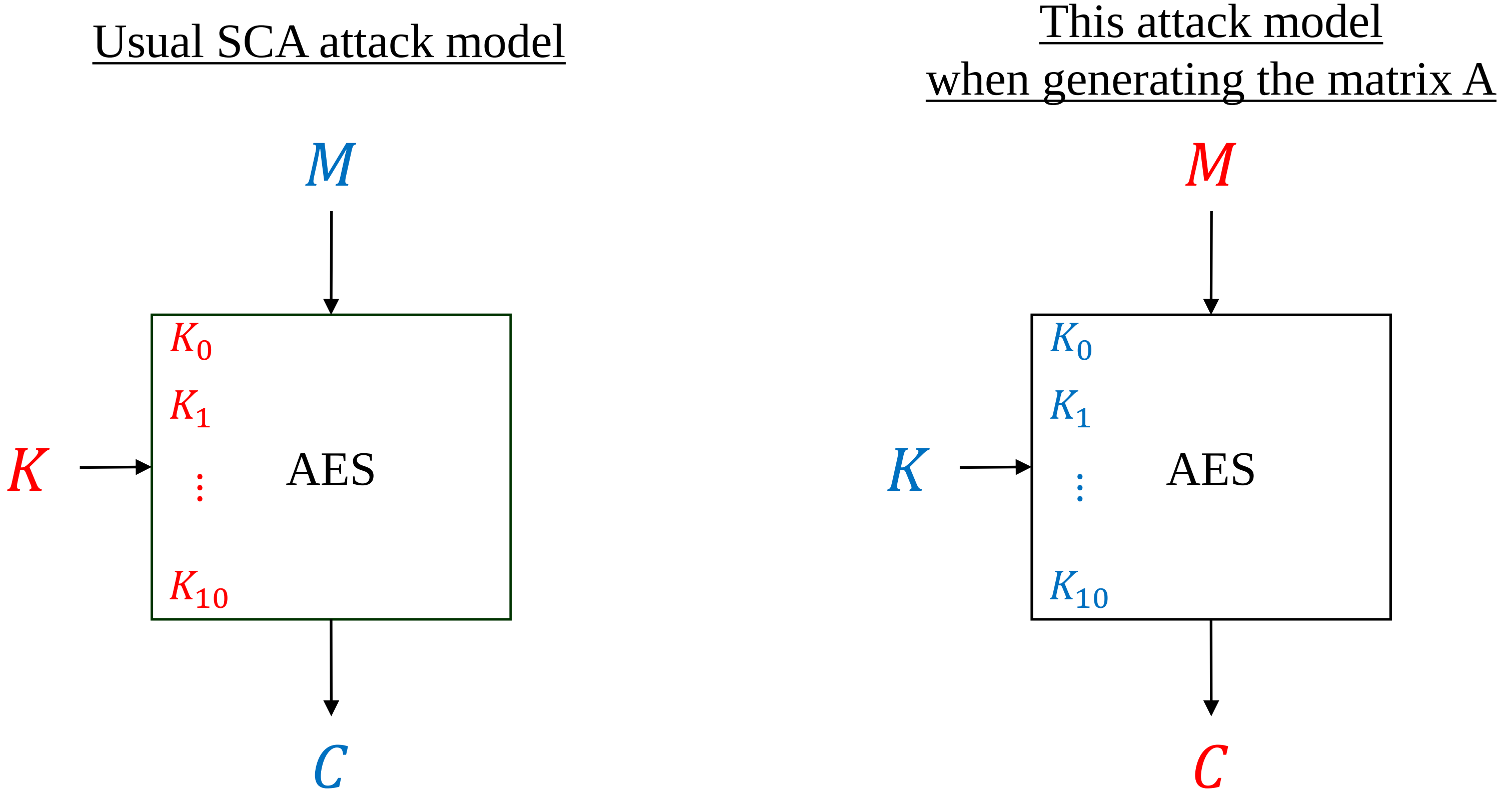
What about a secure implementation of AES?

Usual SCA attack model



Horizontal attack on AES

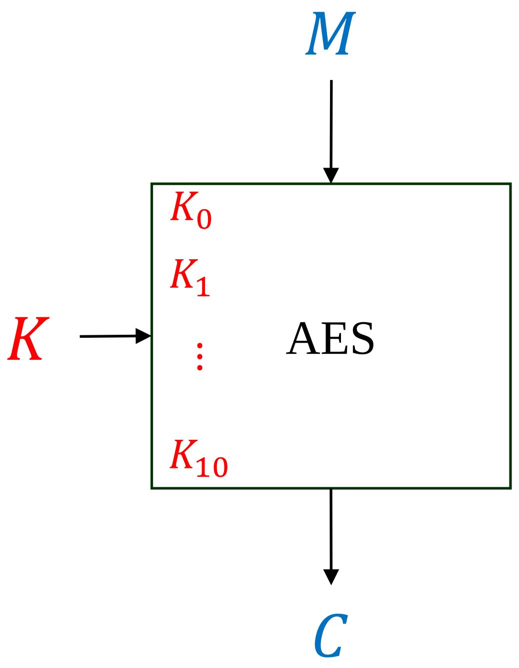
What about a secure implementation of AES?



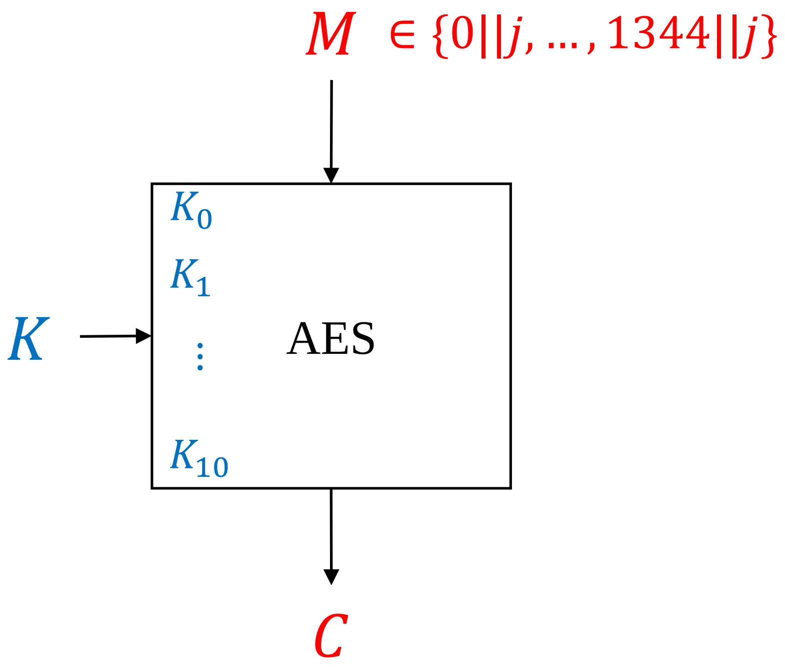
Horizontal attack on AES

What about a secure implementation of AES?

Usual SCA attack model



This attack model
when generating the matrix A



Horizontal attack on AES

- What about a secure implementation of AES?
  - **Unusual attack model**:
    - The **key** is **known**
    - The **message** is **unknown but the set of possible messages is small**

- What about SHAKE instead of AES?
  - **Unusual attack model**
    - The **input** is **unknown but the set of possible inputs is small**

=> It seems difficult to have protection against such attack model, for AES or SHAKE

Shuffling the columns

“Shuffling the columns” ≈ “Random permutation of elements of each row”

Shuffling the columns

$$\begin{pmatrix} s_{0,0} & \cdots & s_{0,\bar{n}-1} \\ \vdots & & \\ \boxed{s_{r,0}} & & \vdots \\ \vdots & & \\ s_{n-1,0} & \cdots & s_{n-1,\bar{n}-1} \end{pmatrix}$$

$$AES(seed_A, 0||j) \longrightarrow \begin{pmatrix} a_{0,0} & \cdots & \boxed{a_{0,r}} & \cdots & a_{0,n-1} \\ & & & & \\ & & & & \end{pmatrix}$$

MINISTÈRE
DES ARMÉES

Liberté
Égalité
Fraternité

Implementation and benchmark

DGA
DIRECTION GÉNÉRALE DE L'ARMEMENT

Implementation and benchmark

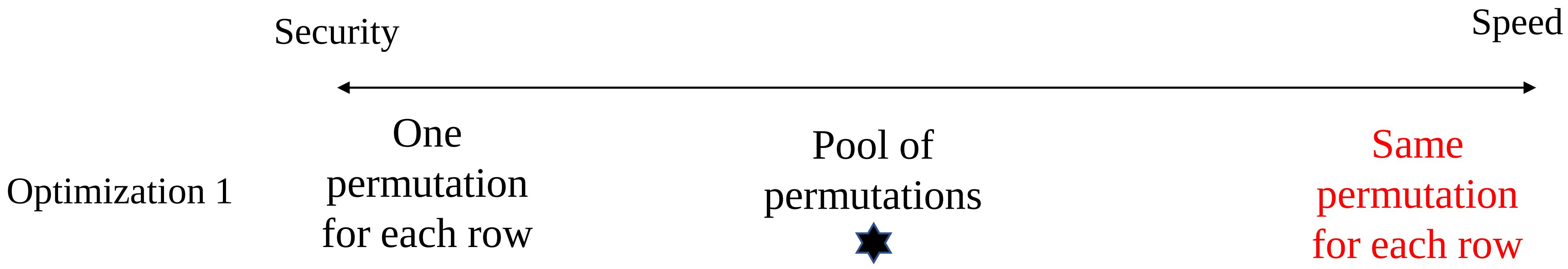
Naive implementation

- On Arm® Cortex®-M7 at 600MHz

Implementation	Execution time for one keygen	Additional Cost
No countermeasure (implementation as is)	**0,55s**	**-**
Shuffle Columns (naive implementation)	**0,75s**	**36%**

Implementation and benchmark

Security vs. Speed



Security

Speed

Optimization 1

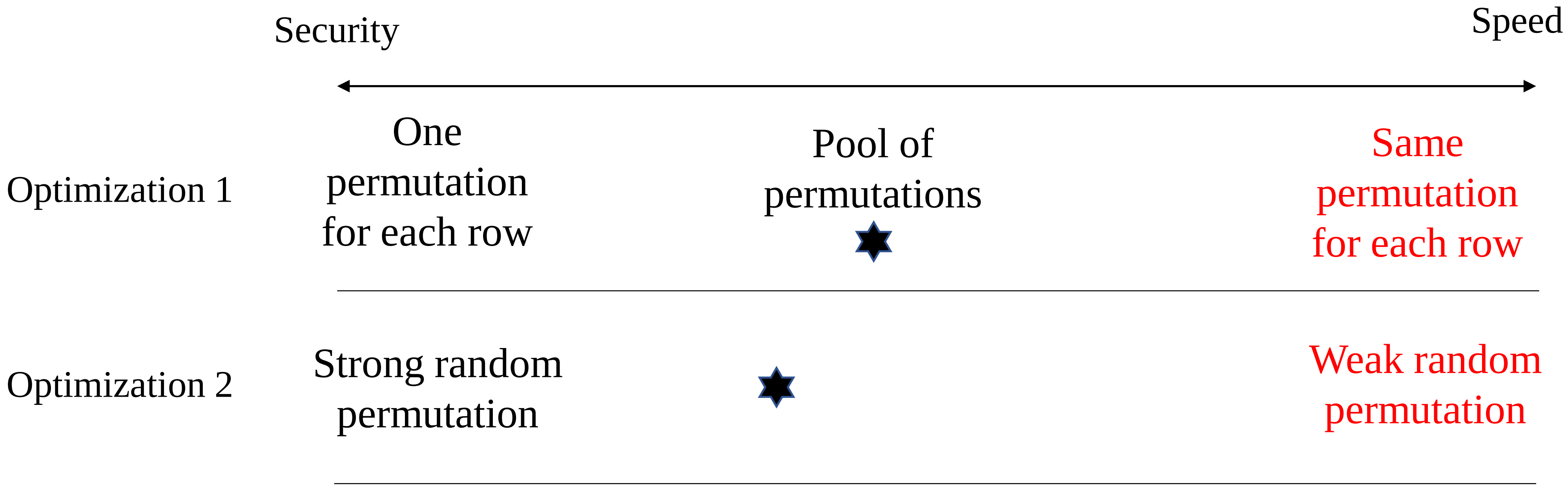
One permutation for each row

Pool of permutations ✶

Same permutation for each row

Implementation and benchmark

Security vs. Speed



Security

Speed

Optimization 1

One permutation for each row

Pool of permutations

✶

Same permutation for each row

Optimization 2

Strong random permutation

✶

Weak random permutation

…

Benchmark

Final implementation

- On Arm® Cortex®-M7 at 600MHz

Implementation	Execution time for one keygen	Additional Cost
No countermeasure (implementation as is)	**0,55s**	**-**
Shuffle Columns (naive implementation)	**0,75s**	**36%**
Shuffle Columns (final implementation)	**0,60s**	**7%**

MINISTÈRE
DES ARMÉES

Liberté
Égalité
Fraternité

Conclusion

DGA
DIRECTION
GÉNÉRALE
DE L'ARMEMENT

Conclusion

- What we achieved
  - Horizontal attack on AES with a very particular attack model
  - Secure implementation of FrodoKEM
- =>Not trivial…

MINISTÈRE
DES ARMÉES

Liberté
Égalité
Fraternité

Thank you

Any questions?

DGA
DIRECTION
GÉNÉRALE
DE L'ARMEMENT

MINISTÈRE DES ARMÉES

Liberté
Égalité
Fraternité

Implementing SCA countermeasures for FrodoKEM is not trivial

Jérémy METAIRIE, Cédric MURDICA, Karl TOURNIER

DGA
DIRECTION GÉNÉRALE DE L'ARMEMENT

20/11/2024

MINISTÈRE
DES ARMÉES

Liberté
Égalité
Fraternité

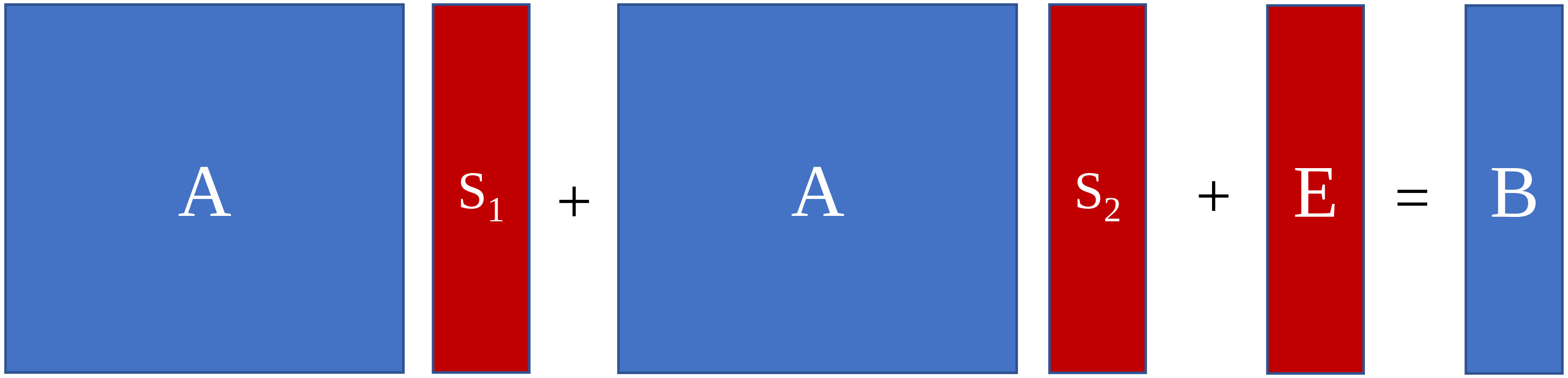
Additional content

DGA

DIRECTION
GÉNÉRALE
DE L'ARMEMENT

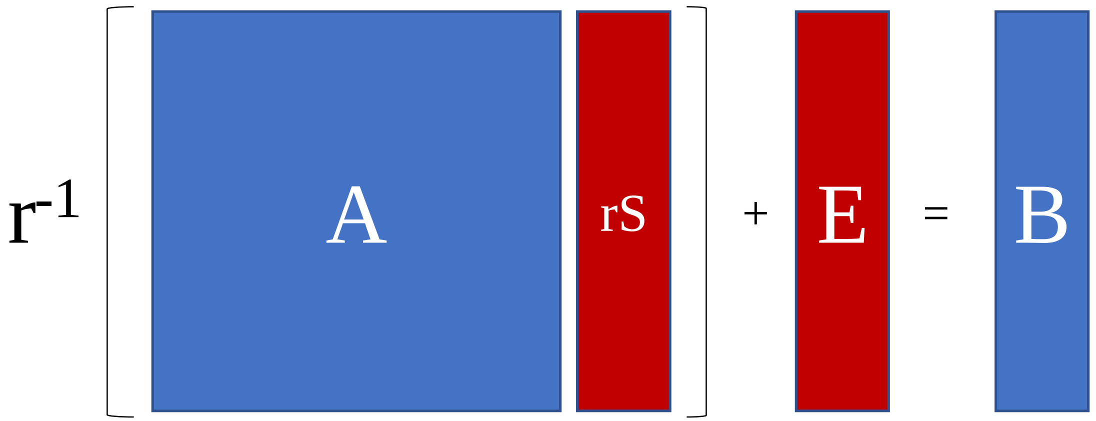
Additive masking

- $S = S_1+S_2$



Multiplicative masking

Randomization of S



Multiplicative masking

Randomization of A