Power side-channel leakage distinguishers on LESSv2.0

Exploiting sparse columns in Gaussian Elimination

Maciej Czuprynko[1] Rishub Nagpal[1] Tobias Schneider[2] Sujoy Sinha Roy[1]

[1]Graz University of Technology| [2]NXP Austria

28 August 2026, SAC

› isec.tugraz.at

Outline

Outline

Motivation

- NIST’s 2nd round standardization candidates for additional signature schemes:

Code-Based	Isogeny-Based	MPC-in-the-Head	Multivariate
CROSS	SQIsign	Mirath (MIRA/MiRitH)	UOV
LESS		MQOM	MAYO
		PERK	QR-UOV
Symmetric-Based	**Lattice-Based**	RYDE	SNOVA
FAEST	Hawk	SDitH	

Motivation

- NIST's 2nd round standardization candidates for additional signature schemes:

Code-Based	Isogeny-Based	MPC-in-the-Head	Multivariate
CROSS	SQIsign	Mirath (MIRA/MiRitH)	UOV
LESS		MQOM	MAYO
		PERK	QR-UOV
Symmetric-Based	**Lattice-Based**	RYDE	SNOVA
FAEST	Hawk	SDitH	

Motivation

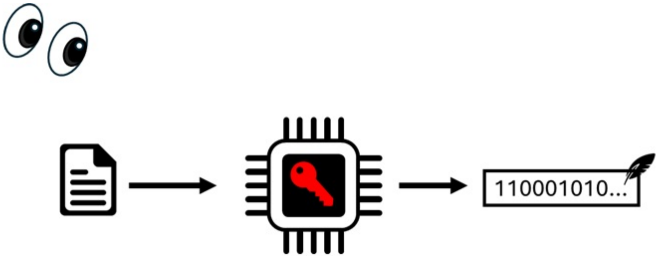
- NIST's 2nd round standardization candidates for additional signature schemes:

Code-Based	Isogeny-Based	MPC-in-the-Head	Multivariate
CROSS	SQIsign	Mirath (MIRA/MiRitH)	UOV
LESS		MQOM	MAYO
		PERK	QR-UOV
Symmetric-Based	**Lattice-Based**	RYDE	SNOVA
FAEST	Hawk	SDitH	

→ Interesting schemes to consider for side-channel analysis.

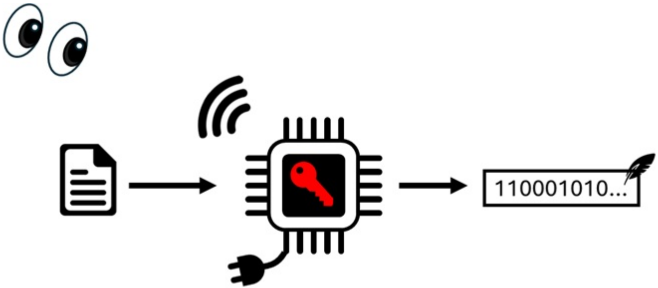
(Passive) Side channel attacks

- An attacker has access to the text and signature . . .



(Passive) Side channel attacks

- An attacker has access to the text and signature . . .
- . . . and also to the computation dependent leakage.



Outline

LESS scheme - Single iteration

$$\mathbf{G}^{(0)}$$

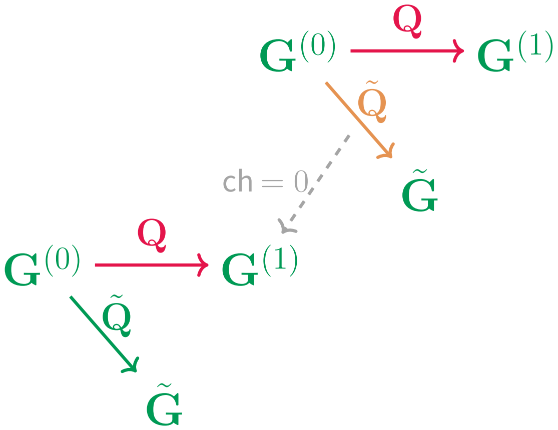
LESS scheme - Single iteration

$$\mathbf{G}^{(0)} \xrightarrow{\mathbf{Q}} \mathbf{G}^{(1)}$$

LESS scheme - Single iteration

$$\begin{tikzcd} \mathbf{G}^{(0)} \arrow[rr, "\mathbf{Q}"] \arrow[dr, "\tilde{\mathbf{Q}}"] & & \mathbf{G}^{(1)} \\ & \tilde{\mathbf{G}} & \end{tikzcd}$$

LESS scheme - Single iteration



LESS scheme - Single iteration

$$\begin{tikzcd} \mathbf{G}^{(0)} \arrow[r, "\mathbf{Q}"] \arrow[dr, "\tilde{\mathbf{Q}}"'] & \mathbf{G}^{(1)} \\ & \tilde{\mathbf{G}} \end{tikzcd}$$

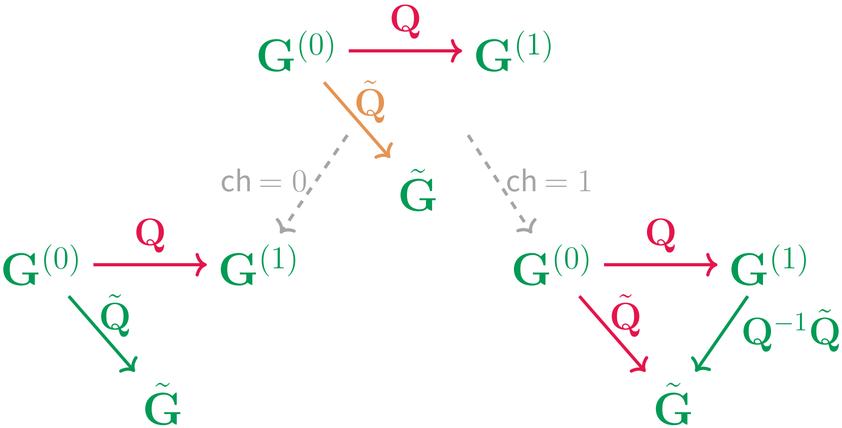
$\mathrm{ch} = 0$

$$\begin{tikzcd} \mathbf{G}^{(0)} \arrow[r, "\mathbf{Q}"] \arrow[dr, "\tilde{\mathbf{Q}}"'] & \mathbf{G}^{(1)} \\ & \tilde{\mathbf{G}} \end{tikzcd}$$

$\mathrm{ch} = 1$

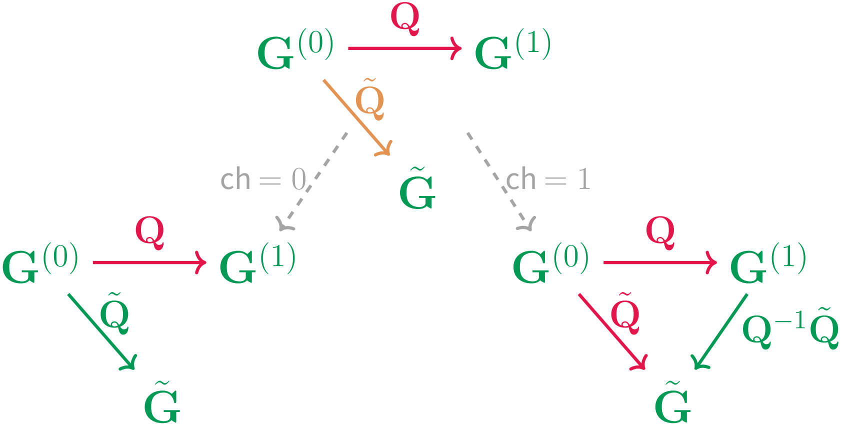
$$\begin{tikzcd} \mathbf{G}^{(0)} \arrow[r, "\mathbf{Q}"] \arrow[dr, "\tilde{\mathbf{Q}}"'] & \mathbf{G}^{(1)} \arrow[d, "\mathbf{Q}^{-1}\tilde{\mathbf{Q}}"] \\ & \tilde{\mathbf{G}} \end{tikzcd}$$

LESS scheme - Single iteration



➔ Side-channel attack only possible when $\mathsf{ch} = 1$.

LESS scheme - Single iteration



- Side-channel attack only possible when ch $= 1$.
- Soundness of $1/2 \rightarrow t$ iterations of the scheme with s public generators.

Computations in LESS - Signature generation

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \in \mathbb{F}_7^{4\times 8}$$

Computations in LESS - Signature generation

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \in \mathbb{F}_7^{4\times 8}$$

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 4 & 8 & 2 & 3 & 7 & 6 & 1 \end{pmatrix}$$

$$\mathbf{d} = \begin{pmatrix} 2 & 5 & 6 & 4 & 4 & 2 & 2 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Transformation $\mathbf{Q} \equiv (\pi, \mathbf{d})$ where:

- $\pi =$ permutation of columns,
- $\mathbf{d} =$ scaling of columns.

Computations in LESS - Signature generation

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \in \mathbb{F}_7^{4\times 8}$$

Transformation $\mathbf{Q} \equiv (\pi, \mathbf{d})$ where:

- $\pi =$ permutation of columns,
- $\mathbf{d} =$ scaling of columns.

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 4 & 8 & 2 & 3 & 7 & 6 & 1 \end{pmatrix}$$

$$\mathbf{d} = \begin{pmatrix} 2 & 5 & 6 & 4 & 4 & 2 & 2 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \xrightarrow{\text{Gaussian Elimination}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 5 & 3 \\ 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 1 & 6 \\ 0 & 0 & 0 & 1 & 6 & 1 & 2 & 6 \end{pmatrix}$$

Underlying problem(s) in LESS

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{(\pi, \mathbf{d}) \text{ and GE}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 5 & 3 \\ 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 1 & 6 \\ 0 & 0 & 0 & 1 & 6 & 1 & 2 & 6 \end{pmatrix} = \mathbf{G}'$$

Underlying problem(s) in LESS

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{(\pi, \mathbf{d}) \text{ and GE}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 5 & 3 \\ 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 1 & 6 \\ 0 & 0 & 0 & 1 & 6 & 1 & 2 & 6 \end{pmatrix} = \mathbf{G}'$$

Linear Equivalence Problem (LEP), informal

Given two matrices $\mathbf{G}$ and $\mathbf{G}'$ find $(\pi, \mathbf{d})$ that transforms one into the other.

Underlying problem(s) in LESS

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{(\pi, \mathbf{d}) \text{ and GE}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 5 & 3 \\ 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 1 & 6 \\ 0 & 0 & 0 & 1 & 6 & 1 & 2 & 6 \end{pmatrix} = \mathbf{G}'$$

Linear Equivalence Problem (LEP), informal

Given two matrices $\mathbf{G}$ and $\mathbf{G}'$ find $(\pi, \mathbf{d})$ that transforms one into the other.

Canonical form Linear Equivalence Problem (CF-LEP), informal

Given two matrices $\mathbf{G}$ and $\mathbf{G}'$, find a set J containing $k(=4)$ columns such that $\mathbf{G}'$ shares the Canonical Form with the matrix obtained from $\mathbf{G}$ by placing the columns in J in the first k positions.

Underlying problem(s) in LESS

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{(\pi, \mathbf{d}) \text{ and GE}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 5 & 3 \\ 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 1 & 2 & 1 & 6 \\ 0 & 0 & 0 & 1 & 6 & 1 & 2 & 6 \end{pmatrix} = \mathbf{G}'$$

Linear Equivalence Problem (LEP), informal

Given two matrices $\mathbf{G}$ and $\mathbf{G}'$ find $(\pi, \mathbf{d})$ that transforms one into the other.

Canonical form Linear Equivalence Problem (CF-LEP), informal

Given two matrices $\mathbf{G}$ and $\mathbf{G}'$, find a set J containing $k(= 4)$ columns such that $\mathbf{G}'$ shares the Canonical Form with the matrix obtained from $\mathbf{G}$ by placing the columns in J in the first k positions.

→ Sufficient to know which columns are mapped to the first k positions.

Outline

Gaussian elimination

Given a matrix $\mathbf{A} \in \mathbb{F}_q^{k \times n}$, the Gaussian elimination is performed as follows:

Gaussian elimination

Given a matrix $\mathbf{A} \in \mathbb{F}_q^{k \times n}$, the Gaussian elimination is performed as follows:

For i **from** 1 **to** k:

1. place the pivot row at the i-th position,

Gaussian elimination

Given a matrix $\mathbf{A} \in \mathbb{F}_q^{k \times n}$, the Gaussian elimination is performed as follows:

For i **from** 1 **to** k:

1. place the pivot row at the i-th position,
2. normalize the pivot row,

Gaussian elimination

Given a matrix $\mathbf{A} \in \mathbb{F}_q^{k \times n}$, the Gaussian elimination is performed as follows:

For i **from** 1 **to** k:

1. place the pivot row at the i-th position,
2. normalize the pivot row,
3. reduce the i-th column.

Attack target

Recall the matrix that needs to be reduced:

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 0$, Steps 1 + 2

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \xrightarrow[]{\substack{\text{1 swap } \mathbf{a}_0 \text{ with } \mathbf{a}_0 \\ \text{2 set } \mathbf{a}_0 = a_{0,0}^{-1} \cdot \mathbf{a}_0}} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 0$, Step 3

$$\rightarrow \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 0$, Step 3

$$\begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \\ \end{array} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} - 2 \cdot \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 0$, Step 3

$$\begin{array}{c c} \rightarrow & \\ & \\ \rightarrow & \\ & \end{array} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} - 6 \cdot \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 0$, Step 3

$$\begin{array}{c} \rightarrow \\ \\ \\ \rightarrow \end{array} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} - 2 \cdot \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 0$, Step 3

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 5 & 0 & 0 & 6 & 0 & 6 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 0$, Summery of Step 3

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 5 & 0 & 0 & 6 & 0 & 6 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 2 \\ 6 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 1$, Steps 1 + 2

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 5 & 0 & 0 & 6 & 0 & 6 & 0 \end{pmatrix} \xrightarrow[]{\substack{\boxed{1}\ \text{swap } \mathbf{a}_1 \text{ with } \mathbf{a}_3 \\ \boxed{2}\ \text{set } \mathbf{a}_1 = a_{1,1}^{-1} \cdot \mathbf{a}_1}} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \end{pmatrix}$$

Attack target (cont'd)

$i = 1$, Step 3

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \end{pmatrix}$$

Attack target (cont'd)

Step 3 for the whole procedure

$$i = 0 : \begin{pmatrix} 0 \\ 2 \\ 6 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \end{pmatrix} \quad i = 1 : \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \end{pmatrix}$$

$$i = 2 : \begin{pmatrix} 3 \\ 0 \\ 0 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 2 & 1 & 6 \end{pmatrix} \quad i = 3 : \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 1 & 6 & 1 & 2 & 6 \end{pmatrix}$$

Attack target (cont'd)

Step 3 for the whole procedure

$$i = 0 : \begin{pmatrix} 0 \\ 2 \\ 6 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \end{pmatrix}, i = 1 : \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \end{pmatrix}$$

$$i = 2 : \begin{pmatrix} 3 \\ 0 \\ 0 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 2 & 1 & 6 \end{pmatrix}, i = 3 : \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 1 & 6 & 1 & 2 & 6 \end{pmatrix}$$

Attack target (cont'd)

Step 3 for the whole procedure

$$i=0: \begin{pmatrix} 0 \\ 2 \\ 6 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \end{pmatrix}, i=1: \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 \end{pmatrix}$$

$$i=2: \begin{pmatrix} 3 \\ 0 \\ 0 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 2 & 1 & 6 \end{pmatrix}, i=3: \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 1 & 6 & 1 & 2 & 6 \end{pmatrix}$$

➔ Different computations depending on the nature of the columns.

Attack target (cont'd)

Step 3 for the whole procedure

$$i=0: \begin{pmatrix} 0 \\ \boxed{2} \\ \boxed{6} \\ \boxed{2} \end{pmatrix} \cdot \begin{pmatrix} \boxed{1} & 0 & \boxed{3} & 0 & \boxed{4} & 0 & \boxed{1} & 0 \end{pmatrix}, i=1: \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & \boxed{1} & 0 & 0 & \boxed{4} & 0 & \boxed{4} & 0 \end{pmatrix}$$

$$i=2: \begin{pmatrix} \boxed{3} \\ 0 \\ 0 \\ \boxed{3} \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & \boxed{1} & 0 & \boxed{1} & \boxed{2} & \boxed{1} & \boxed{6} \end{pmatrix}, i=3: \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & \boxed{1} & \boxed{6} & \boxed{1} & \boxed{2} & \boxed{6} \end{pmatrix}$$

- Different computations depending on the nature of the columns.
- Possible to count the number of sparse columns in the first k columns.

Column corruption

Goal: count the columns of the identity matrix that stay in the first k positions.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{(\pi, \mathbf{d})} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Column corruption

Goal: count the columns of the identity matrix that stay in the first k positions.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{(\pi, \mathbf{d})} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \xrightarrow{\text{iteration } i = 0} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 5 & 0 & 0 & 6 & 0 & 6 & 0 \end{pmatrix}$$

Column corruption

Goal: count the columns of the identity matrix that stay in the first k positions.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{(\pi, \mathbf{d})} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \xrightarrow{\text{iteration } i = 0} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 5 & 0 & 0 & 6 & 0 & 6 & 0 \end{pmatrix}$$

➔ The fifth columns becomes corrupted $\rightarrow$ count is not exact.

Column corruption

Goal: count the columns of the identity matrix that stay in the first k positions.

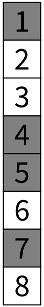
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 4 & 0 & 4 \\ 0 & 1 & 0 & 0 & 5 & 3 & 5 & 1 \\ 0 & 0 & 1 & 0 & 6 & 5 & 4 & 3 \\ 0 & 0 & 0 & 1 & 1 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{(\pi, \mathbf{d})} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 2 & 0 & 2 & 4 & 0 & 3 & 6 & 0 \\ 6 & 0 & 1 & 0 & 0 & 1 & 3 & 3 \\ 2 & 5 & 6 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \xrightarrow{\text{iteration } i = 0} \begin{pmatrix} 1 & 0 & 3 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 3 & 4 & 6 & 3 & 4 & 0 \\ 0 & 0 & 4 & 0 & 4 & 1 & 4 & 3 \\ 0 & 5 & 0 & 0 & 6 & 0 & 6 & 0 \end{pmatrix}$$

- The fifth columns becomes corrupted $\rightarrow$ count is not exact.
- Introduction of **algorithmic** noise to the SCA.

Key recovery

Secret:

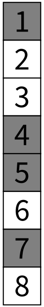


. . .

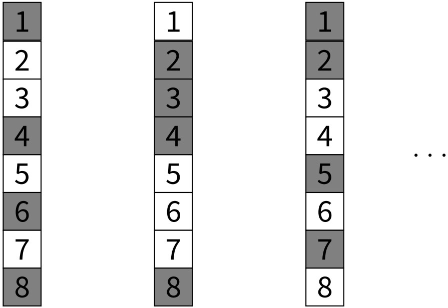
For $n = 8$
and $k = 4$

Key recovery

Secret:



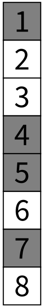
Signatures:



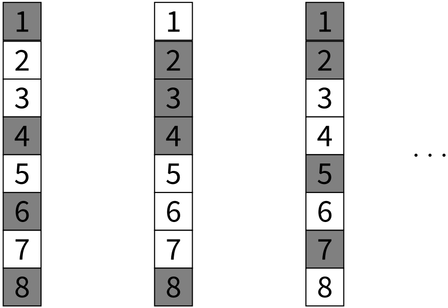
For $n = 8$
and $k = 4$

Key recovery

Secret:



Signatures:



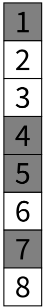
For $n = 8$
and $k = 4$

SCA:
shared values

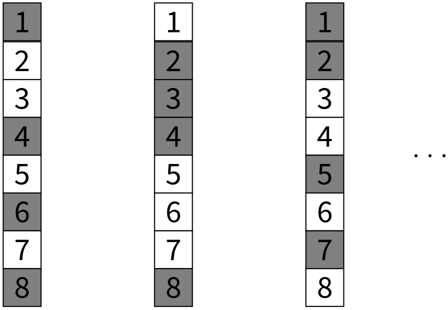
2 1 3

Key recovery

Secret:



Signatures:



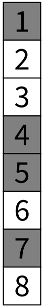
For $n = 8$
and $k = 4$

SCA:
shared values

$2 + e_1$ $\quad$ $1 + e_2$ $\quad$ $3 + e_3$

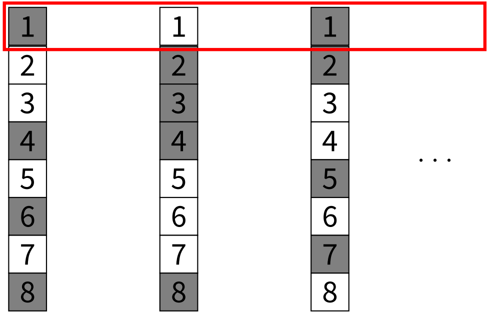
Key recovery

Secret:



Signatures:

+1 0 +1

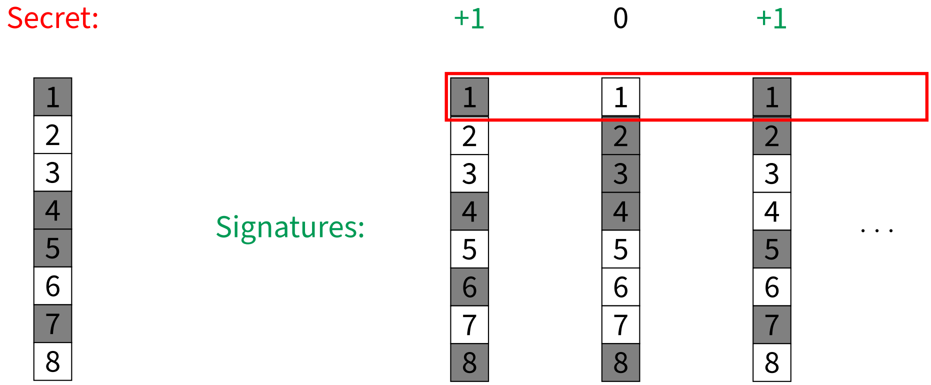


For $n = 8$
and $k = 4$

SCA:
shared values

$2 + e_1$ $1 + e_2$ $3 + e_3$

Key recovery



For $n = 8$
and $k = 4$

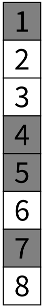
SCA:
shared values

$2 + e_1$ $1 + e_2$ $3 + e_3$

➔ Index in the set: **positive** correlation.

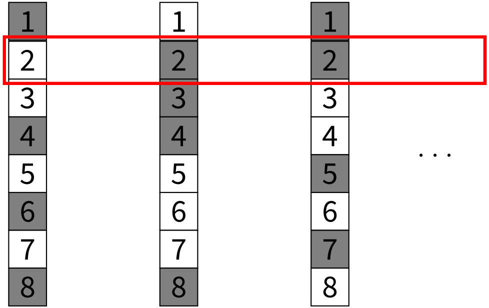
Key recovery

Secret:



Signatures:

0 -1 -1



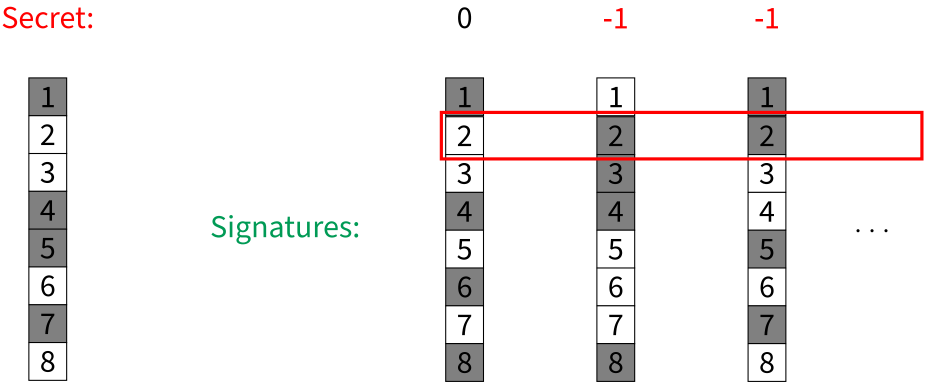
For $n = 8$
and $k = 4$

SCA:
shared values

2 + e_1 1 + e_2 3 + e_3

➔ Index in the set: **positive** correlation.

Key recovery



For $n = 8$ and $k = 4$

SCA: # shared values

2 + e_1 1 + e_2 3 + e_3

Index in the set: **positive** correlation. Index not in the set: **negative** correlation.

Outline

Countermeasures

- **Blinding:**
  - Multiply the matrix by a random invertible matrix.
  - The operation is invariant to the Gaussian elimination.
  - The result is an instance of the LEP.

Countermeasures

- **Blinding**:
  - Multiply the matrix by a random invertible matrix.
  - The operation is invariant to the Gaussian elimination.
  - The result is an instance of the LEP.
- **1st-order masking**:
  - Consists into splitting the computation into two parts that are random if observed one at a time.
  - Attacks still possible by recombining the parts.

Countermeasures

- **Blinding**:
  - Multiply the matrix by a random invertible matrix.
  - The operation is invariant to the Gaussian elimination.
  - The result is an instance of the LEP.
- **1st-order masking**:
  - Consists into splitting the computation into two parts that are random if observed one at a time.
  - Attacks still possible by recombining the parts.
- **Shuffling**:
  - Randomization of computation order.
  - Does not remove the leakage.

Outline

Simulation results

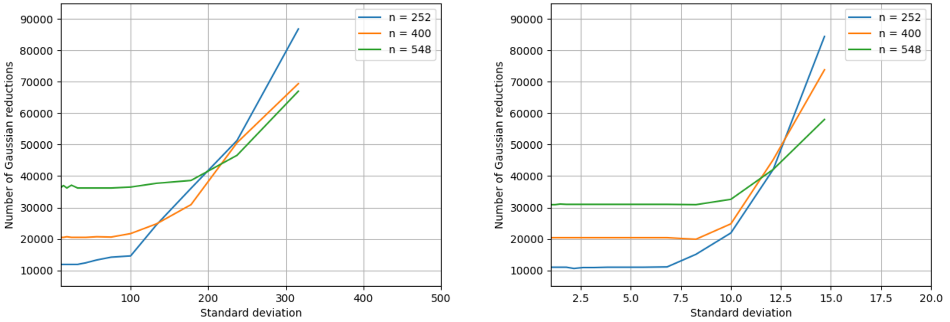


Figure: Required number of observed Gaussian eliminations to recover the secret for unprotected (left) and masked (right) implementations.

Simulation results

NIST Cat.	Parameter Set	Code Params n	k	q	Prot. Params t	w	s	Required signatures
1	LESS-252-192	252	126	127	192	36	2	318.1
	LESS-252-68				68	42	4	817.9
	LESS-252-45				45	34	8	2357.4
3	LESS-400-220	400	200	127	220	68	2	300.7
	LESS-400-102				102	61	4	1005.7
5	LESS-548-345	548	274	127	345	75	2	448.0
	LESS-548-137				137	79	4	1275.9

Table: Attack complexity estimations for all LESS parameter sets.

Simulation results

NIST Cat.	Parameter Set	Code Params n	k	q	Prot. Params t	w	s	Required signatures
1	LESS-252-192	252	126	127	192	36	2	318.1
	LESS-252-68				68	42	4	817.9
	LESS-252-45				45	34	8	2357.4
3	LESS-400-220	400	200	127	220	68	2	300.7
	LESS-400-102				102	61	4	1005.7
5	LESS-548-345	548	274	127	345	75	2	448.0
	LESS-548-137				137	79	4	1275.9

Table: Attack complexity estimations for all LESS parameter sets.

Leakage presence

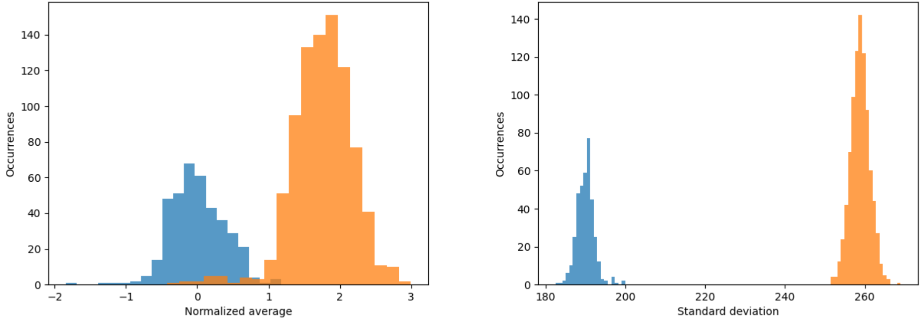


Figure: Distribution of the distinguisher values for unprotected (left) and masked (right) implementations. The results are for 10 Gaussian eliminations measurements on an Cortex-M4.

Outline

Conclusion

- We have shown a zero-value distinguisher for LESS.
- From simulation, the proposed attack would require between 301 and 2358 signatures.
- We discussed countermeasures such as shuffling, masking and blinding are discussed.

Conclusion

- We have shown a zero-value distinguisher for LESS.
- From simulation, the proposed attack would require between 301 and 2358 signatures.
- We discussed countermeasures such as shuffling, masking and blinding are discussed.

Future works:

- Explore the secure generation of the invertible matrix used in blinding.

IPCEI Microelectronics and
Communication Technologies

Acknowledgment

The SINFONIA project has received funding from the Recovery and Resilience Facility (RRF) as the centrepiece of NextGenerationEU via the Austrian Research Promotion Agency (FFG) and Austria Wirtschaftsservice Gesellschaft mbH (aws) in the frame of the IPCEI ME/CT – Important Project of Common European Interest on Microelectronic and Communication Technologies under FFG project No 917423 and AWS project No P2431566.

Funded by

Federal Ministry
Innovation, Mobility
and Infrastructure
Republic of Austria

Federal Ministry
Economy, Energy
and Tourism
Republic of Austria

aws austria wirtschaftsservice

FFG Forschung wirkt.

Funded by
the European Union
NextGenerationEU

SENSE THINK ACT COMMUNICATE

Power side-channel leakage distinguishers on LESSv2.0

Exploiting sparse columns in Gaussian Elimination

Maciej Czuprynko[1] Rishub Nagpal[1] Tobias Schneider[2] Sujoy Sinha Roy[1]

[1]Graz University of Technology| [2]NXP Austria

28 August 2026, SAC