

SESSION ID: CRYPT-W01A

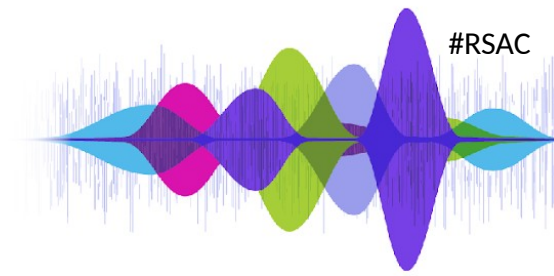
Masking Gaussian Elimination at Arbitrary Order *with Application to Multivariate- and Code-Based PQC*

Quinten Norga

PhD Candidate
COSIC, KU Leuven

Co-Authors: Suparna Kundu, Uttam Kumar Ojha, Anindya Ganguly, Angshuman Karmakar, Ingrid Verbauwhede

Disclaimer



Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

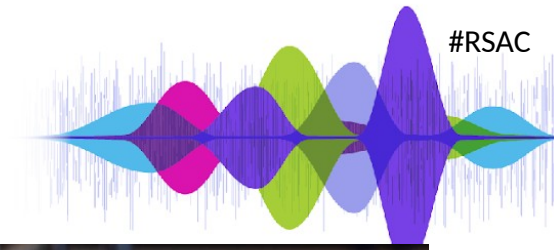
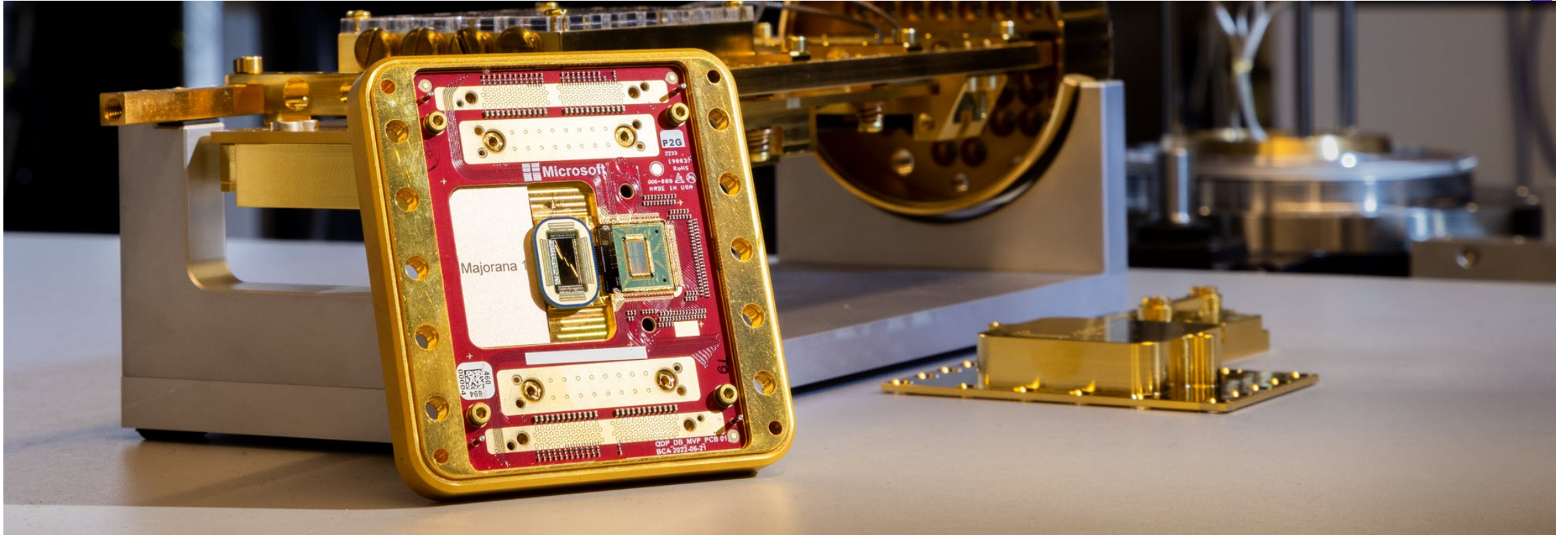
Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Outline

- * Introduction: Multivariate- and Code-Based PQC
- * Masking Gaussian Elimination w/ Back Substitution
- * Implementation & Evaluation on Arm Cortex-M4

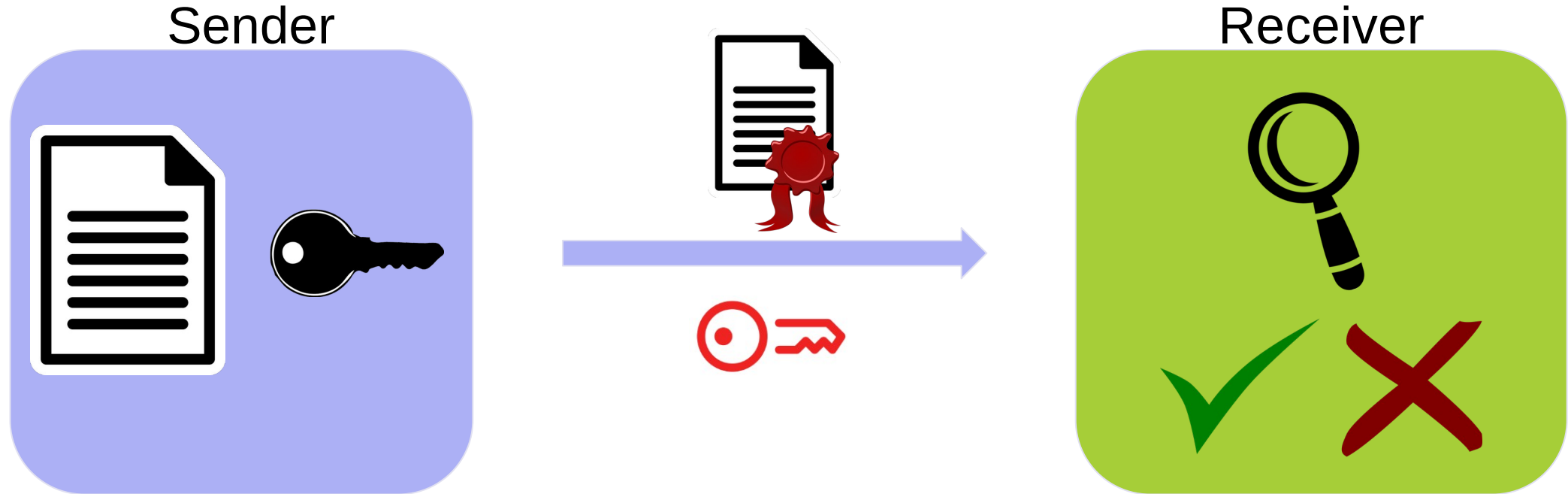
Post-Quantum Cryptography (PQC)



Majorana 1

[[Microsoft](#)]

Digital Signature Algorithm (DSA)



PQC DSA Standardization

NIST

FIPS 204

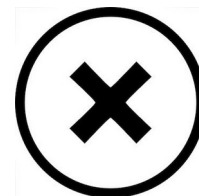
Federal Information Processing Standards Publication

Module-Lattice-Based Digital Signature Standard

Category: Computer Security

Subcategory: Cryptography

[FIPS 204]



and many more...

FIPS 205

Federal Information Processing Standards Publication

[FIPS 205]

Stateless Hash-Based Digital Signature Standard

Category: Computer Security

Subcategory: Cryptography

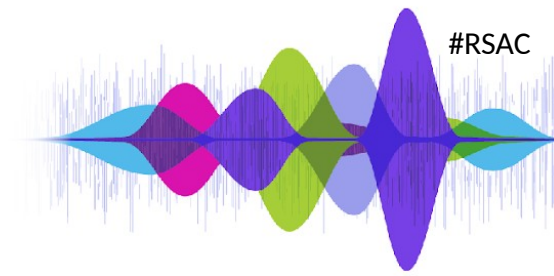


[NIST]



COSIC

Multivariate- and Code-Based DSA



$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

System of Linear Equations

Linear Algebra 101: Gaussian Elimination

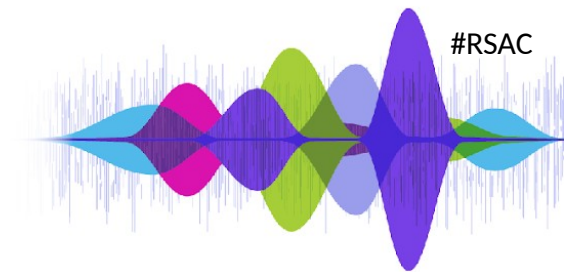
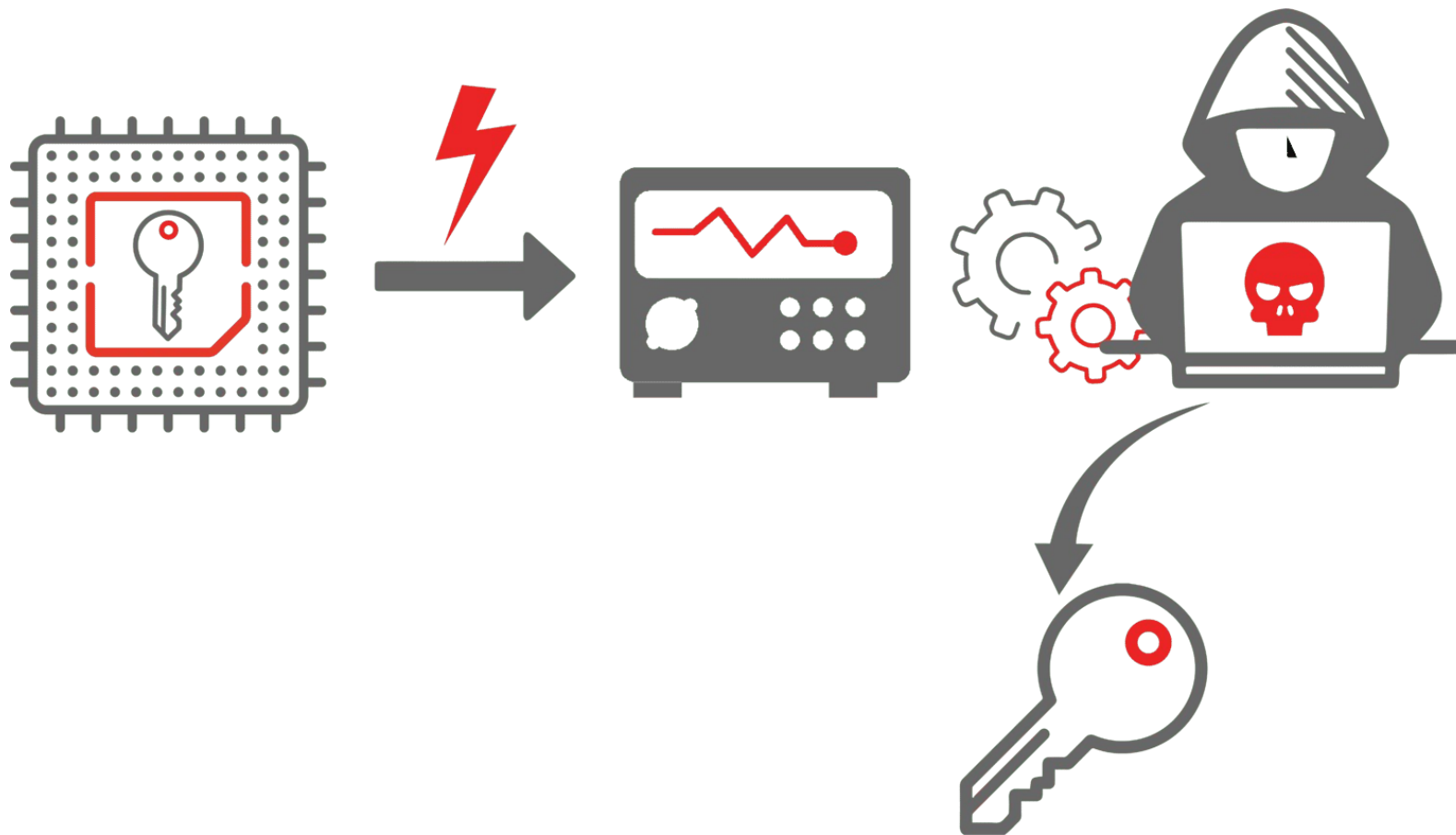
$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} & b_1 \\ a_{21} & a_{22} & \dots & a_{2m} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mm} & b_m \end{bmatrix}$$

Row-Echelon Form

$$\left[\begin{array}{cccc|c} 1 & a'_{12} & \dots & a'_{1m} & b'_1 \\ 0 & 1 & \dots & a'_{2m} & b'_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & b'_m \end{array} \right]$$

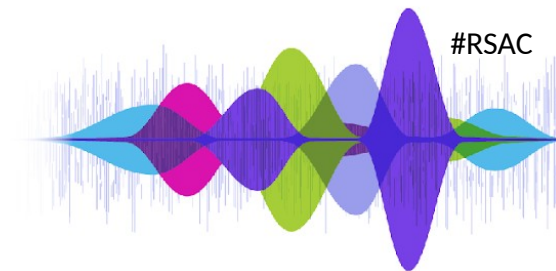
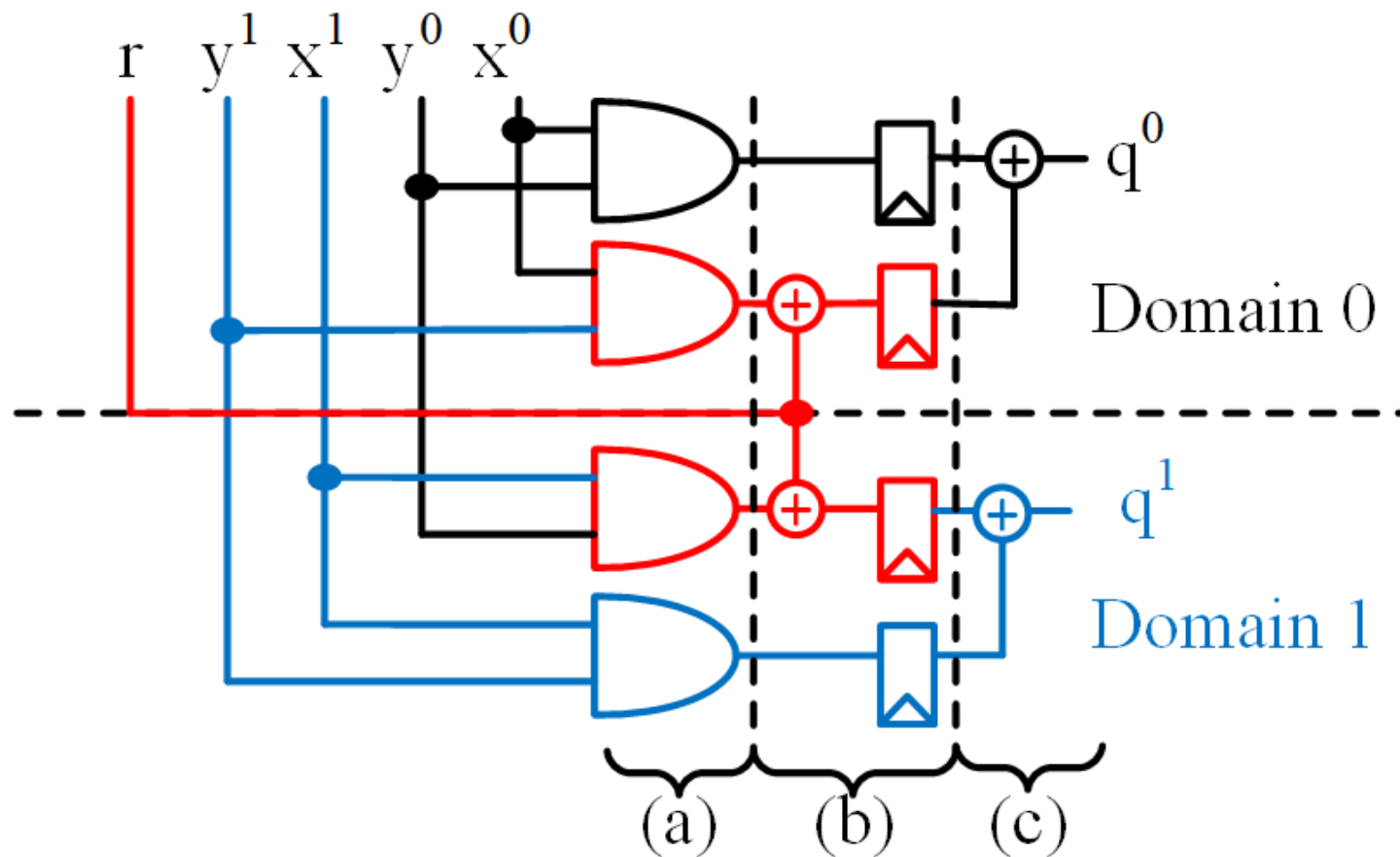
Row-Operations: Swap, Multiply, Add

Side-Channel Attacks (SCAs)



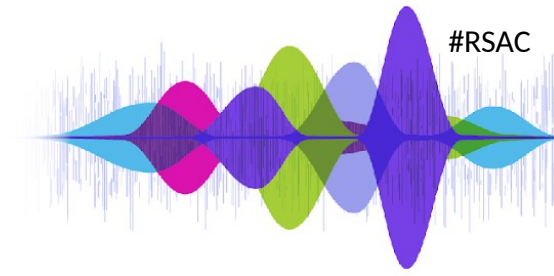
[[Secure-IC](#)]

Masking



[Gross et al.]

Masked System of Linear Equations



$$\begin{bmatrix} (a^1_{11}, a^2_{11}) & (a^1_{12}, a^2_{12}) & \dots & (a^1_{1m}, a^2_{1m}) & (b^1_1, b^2_1) \\ (a^1_{21}, a^2_{21}) & (a^1_{22}, a^2_{22}) & \dots & (a^1_{2m}, a^2_{2m}) & (b^1_2, b^2_2) \\ \dots & \dots & \dots & \dots & \dots \\ (a^1_{m1}, a^2_{m1}) & (a^1_{m2}, a^2_{m2}) & \dots & (a^1_{mm}, a^2_{mm}) & (b^1_m, b^2_m) \end{bmatrix}$$

$$a^1_{xy} * a^2_{xy} = a_{xy}$$

Multiplicative Sharing

Masking Gaussian Elimination

A decorative graphic at the bottom of the slide. It features a horizontal line of fine, vertical, light blue lines that resemble a noise or signal pattern. Overlaid on this pattern are several large, overlapping, teardrop-shaped or petal-like forms in various colors: light blue, purple, magenta, green, and light purple. These shapes are arranged in a symmetrical, star-like pattern that tapers towards the center.

Many Voices.
One Community.

Step 1: Make Pivot Non-Zero

$$\begin{bmatrix} (a_{11}^1, a_{11}^2) & (a_{12}^1, a_{12}^2) & \dots & (a_{1m}^1, a_{1m}^2) & (b_1^1, b_1^2) \\ (a_{21}^1, a_{21}^2) & (a_{22}^1, a_{22}^2) & \dots & (a_{2m}^1, a_{2m}^2) & (b_2^1, b_2^2) \\ \dots & \dots & \dots & \dots & \dots \\ (a_{m1}^1, a_{m1}^2) & (a_{m2}^1, a_{m2}^2) & \dots & (a_{mm}^1, a_{mm}^2) & (b_m^1, b_m^2) \end{bmatrix}$$

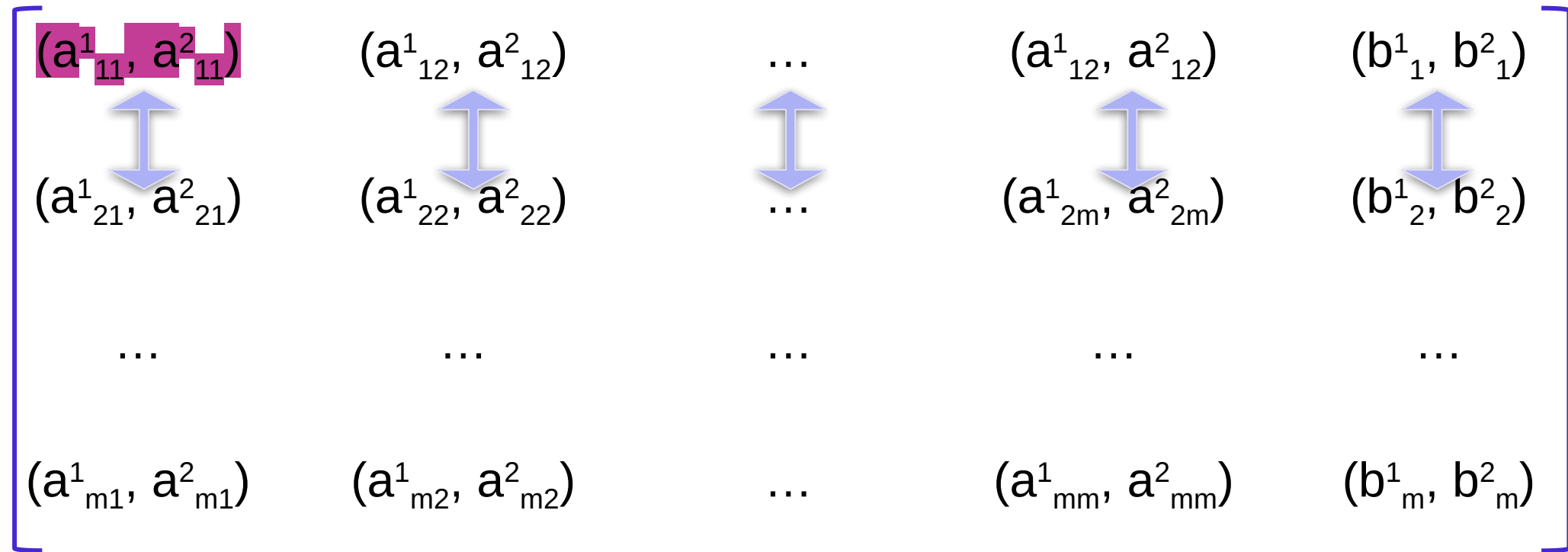
$$a_{11}^1 + a_{11}^2 = a_{11}^? \stackrel{?}{=} 0$$

SecNonzero [CC24]

$$a_{11}^1 + a_{11}^2 = a_{11} \stackrel{?}{=} 0$$

- * Returns masked bool (1/0)
- * Unmasking *leaks* information

Step 1: Make Pivot Non-Zero



$a^1 + a^2 = a_1 = 0$
Not Constant Time!

SecCondAdd

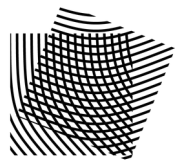
* Conditional Addition

— Add rows to Pivot row...

$$a_{11}^1 + a_{11}^2 = a_{11} = 0$$

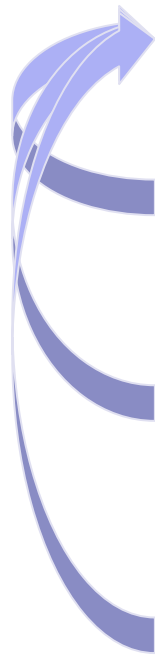
— or add 0 to Pivot row

$$a_{11}^1 + a_{11}^2 = a_{11} \neq 0$$



— Details in paper

Step 1: Make Pivot Non-Zero



$$\begin{bmatrix} (a^1_{11}, a^2_{11}) & (a^1_{12}, a^2_{12}) & \dots & (a^1_{1m}, a^2_{1m}) & (b^1_1, b^2_1) \\ (a^1_{21}, a^2_{21}) & (a^1_{22}, a^2_{22}) & \dots & (a^1_{2m}, a^2_{2m}) & (b^1_2, b^2_2) \\ \dots & \dots & \dots & \dots & \dots \\ (a^1_{m1}, a^2_{m1}) & (a^1_{m2}, a^2_{m2}) & \dots & (a^1_{mm}, a^2_{mm}) & (b^1_m, b^2_m) \end{bmatrix}$$

SecCondAdd (CT)

Step 1: Make Pivot Non-Zero



$$\begin{bmatrix} (a^1_{11}, a^2_{11}) & (a^1_{12}, a^2_{12}) & \dots & (a^1_{1m}, a^2_{1m}) & (b^1_1, b^2_1) \\ (a^1_{21}, a^2_{21}) & (a^1_{22}, a^2_{22}) & \dots & (a^1_{2m}, a^2_{2m}) & (b^1_2, b^2_2) \\ \dots & \dots & \dots & \dots & \dots \\ (a^1_{m1}, a^2_{m1}) & (a^1_{m2}, a^2_{m2}) & \dots & (a^1_{mm}, a^2_{mm}) & (b^1_m, b^2_m) \end{bmatrix}$$

Early Stop (details in paper)

Step 2: Check Pivot

$$\begin{bmatrix} (a^1_{11}, a^2_{11}) & (a^1_{12}, a^2_{12}) & \dots & (a^1_{1m}, a^2_{1m}) & (b^1_1, b^2_1) \\ (a^1_{21}, a^2_{21}) & (a^1_{22}, a^2_{22}) & \dots & (a^1_{2m}, a^2_{2m}) & (b^1_2, b^2_2) \\ \dots & \dots & \dots & \dots & \dots \\ (a^1_{m1}, a^2_{m1}) & (a^1_{m2}, a^2_{m2}) & \dots & (a^1_{mm}, a^2_{mm}) & (b^1_m, b^2_m) \end{bmatrix}$$

SecNonzero

SecNonzero [CC24] + Unmask

$$a_{11}^1 + a_{11}^2 = a_{11} \stackrel{?}{=} 0$$

- * Leak if pivot is (non-)zero
- * No unique solution → OK!

Step 2: Check Pivot

$$\left[\begin{array}{ccccc} \boxed{a_{11} = 0} & (a_{12}^1, a_{12}^2) & \dots & (a_{12}^1, a_{12}^2) & (b_1^1, b_1^2) \\ (a_{21}^1, a_{21}^2) & (a_{22}^1, a_{22}^2) & \dots & (a_{2m}^1, a_{2m}^2) & (b_2^1, b_2^2) \\ \dots & \dots & \dots & \dots & \dots \\ (a_{m1}^1, a_{m1}^2) & (a_{m2}^1, a_{m2}^2) & \dots & (a_{mm}^1, a_{mm}^2) & (b_m^1, b_m^2) \end{array} \right]$$

Start over!

Step 3: Make Pivot 1

$$\left[\begin{array}{ccccc} (a_{11}^1, a_{11}^2) & (a_{12}^1, a_{12}^2) & \dots & (a_{1m}^1, a_{1m}^2) & (b_1^1, b_1^2) \\ (a_{21}^1, a_{21}^2) & (a_{22}^1, a_{22}^2) & \dots & (a_{2m}^1, a_{2m}^2) & (b_2^1, b_2^2) \\ \dots & \dots & \dots & \dots & \dots \\ (a_{m1}^1, a_{m1}^2) & (a_{m2}^1, a_{m2}^2) & \dots & (a_{mm}^1, a_{mm}^2) & (b_m^1, b_m^2) \end{array} \right]$$

$$a_{11}^1 * a_{11}^{2-1} = 1$$

B2Minv

- * Multiplication prefers multiplicative masking
 - B2M conversion [[GenProQui11](#)]

$$(a^1_{11} + a^2_{11}) = (a'^1_{11} * a'^2_{11})$$

SecScalarMult

- * Strictly* linear operations

$$(a_{11}^1 + a_{11}^2) * a_{11}'^1 * a_{11}'^2$$

- * Total Cost:
 - 1 B2M conversion
 - NO M2B conversion

Step 3: Make Pivot 1

$$\begin{bmatrix} (a_{11}^1, a_{11}^2) & (a_{12}^1, a_{12}^2) & \dots & (a_{1m}^1, a_{1m}^2) & (b_1^1, b_1^2) \\ (a_{21}^1, a_{21}^2) & (a_{22}^1, a_{22}^2) & \dots & (a_{2m}^1, a_{2m}^2) & (b_2^1, b_2^2) \\ \dots & \dots & \dots & \dots & \dots \\ (a_{m1}^1, a_{m1}^2) & (a_{m2}^1, a_{m2}^2) & \dots & (a_{mm}^1, a_{mm}^2) & (b_m^1, b_m^2) \end{bmatrix}$$

$$(a_{11}^1 + a_{11}^2) = 1$$

Step 3: Make Pivot 1

$$\begin{bmatrix} (a_{11}^1, a_{11}^2) & (a_{12}^1, a_{12}^2) & \dots & (a_{1m}^1, a_{1m}^2) & (b_1^1, b_1^2) \\ (a_{21}^1, a_{21}^2) & (a_{22}^1, a_{22}^2) & \dots & (a_{2m}^1, a_{2m}^2) & (b_2^1, b_2^2) \\ \dots & \dots & \dots & \dots & \dots \\ (a_{m1}^1, a_{m1}^2) & (a_{m2}^1, a_{m2}^2) & \dots & (a_{mm}^1, a_{mm}^2) & (b_m^1, b_m^2) \end{bmatrix}$$

SecScalarMult on Row 1

Step 4: Make Lower Elements 0

$$\begin{bmatrix}
 (a^1_{11}, a^2_{11}) & (a^1_{12}, a^2_{12}) & \dots & (a^1_{1m}, a^2_{1m}) & (b^1_1, b^2_1) \\
 (a^1_{21}, a^2_{21}) & (a^1_{22}, a^2_{22}) & \dots & (a^1_{2m}, a^2_{2m}) & (b^1_2, b^2_2) \\
 \dots & \dots & \dots & \dots & \dots \\
 (a^1_{m1}, a^2_{m1}) & (a^1_{m2}, a^2_{m2}) & \dots & (a^1_{mm}, a^2_{mm}) & (b^1_m, b^2_m)
 \end{bmatrix}$$

Subtract first-row from rows below: SecMultSub

Step 4: Make Lower Elements 0

$$\begin{bmatrix}
 (a_{11}^1, a_{11}^2) & (a_{12}^1, a_{12}^2) & \dots & (a_{1m}^1, a_{1m}^2) & (b_1^1, b_1^2) \\
 (a_{21}^1, a_{21}^2) & (a_{22}^1, a_{22}^2) & \dots & (a_{2m}^1, a_{2m}^2) & (b_2^1, b_2^2) \\
 \dots & \dots & \dots & \dots & \dots \\
 (a_{m1}^1, a_{m1}^2) & (a_{m2}^1, a_{m2}^2) & \dots & (a_{mm}^1, a_{mm}^2) & (b_m^1, b_m^2)
 \end{bmatrix}$$

Subtract first-row from rows below: SecMultSub

New Pivot: Repeat Step 1-4

$$\left[\begin{array}{ccccc} (a_{11}^1, a_{11}^2) & (a_{12}^1, a_{12}^2) & \dots & (a_{1m}^1, a_{1m}^2) & (b_1^1, b_1^2) \\ (a_{21}^1, a_{21}^2) & (a_{22}^1, a_{22}^2) & \dots & (a_{2m}^1, a_{2m}^2) & (b_2^1, b_2^2) \\ \dots & \dots & \dots & \dots & \dots \\ (a_{m1}^1, a_{m1}^2) & (a_{m2}^1, a_{m2}^2) & \dots & (a_{mm}^1, a_{mm}^2) & (b_m^1, b_m^2) \end{array} \right]$$

$$a_{22}^1 + a_{22}^2 = a_{22}^? = 0$$

Resulting Matrix

$$\begin{bmatrix} (a^1_{11}, a^2_{11}) & (a^1_{12}, a^2_{12}) & \dots & (a^1_{1m}, a^2_{1m}) & (b^1_1, b^2_1) \\ (a^1_{21}, a^2_{21}) & (a^1_{22}, a^2_{22}) & \dots & (a^1_{2m}, a^2_{2m}) & (b^1_2, b^2_2) \\ \dots & \dots & \dots & \dots & \dots \\ (a^1_{m1}, a^2_{m1}) & (a^1_{m2}, a^2_{m2}) & \dots & (a^1_{mm}, a^2_{mm}) & (b^1_m, b^2_m) \end{bmatrix}$$

Final step: back-substitution

SecBackSub

- * See paper for details
- * Reduce cost through early unmasking

Application to PQC DS Schemes

- * Cost determined by
 - **m** (matrix dimension)
 - **q** (modulus)
 - **n** (# shares)

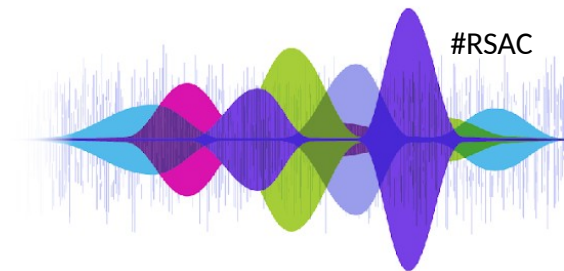
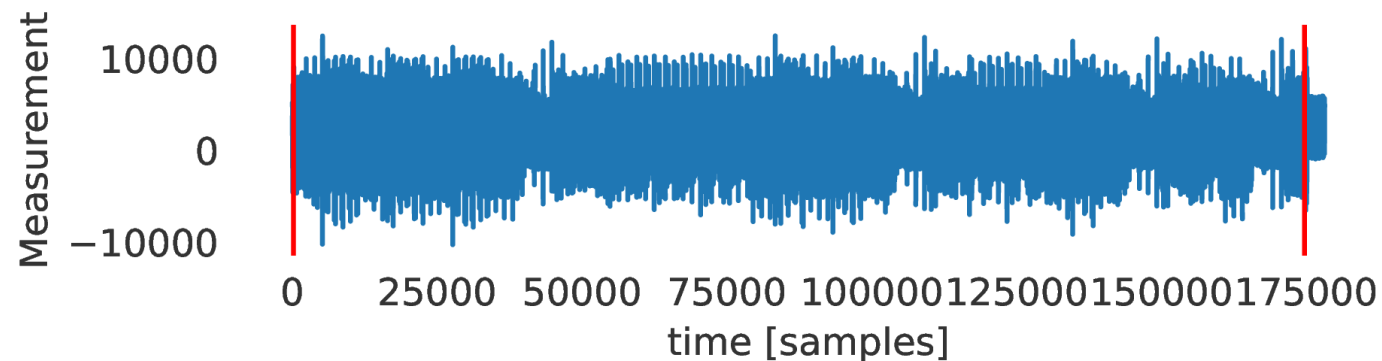
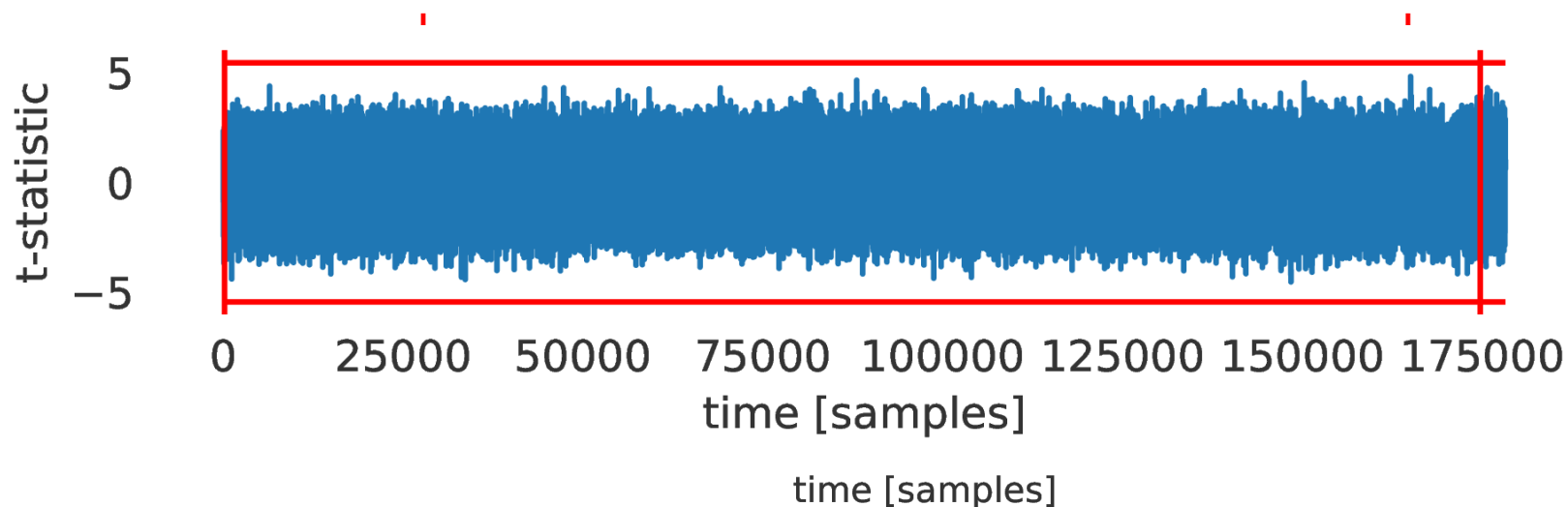
- * NIST Additional DS Round 2 & Korean PQC Candidates
 - UOV, MAYO, SNOVA, QR-UOV, MQ-Sign

UOV Implementation on Arm Cortex-M4

Scheme	Operation	Unmasked	Masked		
			1st-order	2nd-order	3rd-order
UOV-I (m=44 q=256)	GE (Total)	1034	15649 (15.1×)	39587 (38.3×)	69186 (66.9×)
	SecRowEch	999	15575 (15.6×)	39469 (39.5×)	69021 (69.1×)
	Step 1	80	1998 (25×)	5642 (70.5×)	9901 (123.8×)
	Step 2		39	108	193
	Step 3		598	1335	2426
	Step 4		12938	32383	56498
	SecBackSub	32	74 (2.3×)	118 (3.7×)	164 (5.1×)
UOV-III	GE (Total)	4117	62680 (15.2×)	157901 (38.4×)	275768 (67×)
UOV-V	GE (Total)	9336	143361 (15.4×)	360590 (38.6×)	1025942 (109.9×)

- * STM32 Discovery
- * C code (arm-none-eabi-gcc)
- * On-the-fly TRNG

Security Evaluation in COSIC Lab



Countermeasure
ON

Test Vector Leakage Assessment (TVLA)

Mitigation of micro-architectural leakages

Key Takeaways

How to Apply our results?



Many Voices.
One Community.

Conclusion

- * **Real-World** requires **Secure & Efficient** implementations
 - Design of PQC schemes
 - Implementation of PQC schemes
- * PQC transition
 - Crypto inventory
 - Agility: many standards!

RSAC[™] | 2025
Conference

Many Voices.
One Community.

Thank You!

Questions?