Quasi-linear masking against SCA and FIA, with cost amortization

Speaker: **Sylvain Guilley** (*Secure-IC S.A.S.*)

Co-authors:

- **Claude Carlet** (*Paris 8 & 13, Bergen U.*),
- **Abderrahman Daif** (*Bull SAS - ATOS*), and
- **Cédric Tavernier** (*Hensoldt France*)

Date: September 2024

Place: Halifax, Nova Scotia, Canada

Presentation of following paper: Claude Carlet, Abderrahman Daif, Sylvain Guilley, Cédric Tavernier: Quasi-linear masking against SCA and FIA, with cost amortization. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2024(1): 398-432 (2024)

Up-to-date version: https://eprint.iacr.org/2023/406

Version 3

AGENDA

SECURE-iC
THE SECURITY SCIENCE COMPANY

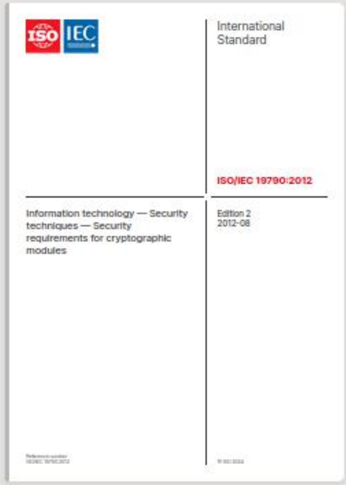
AGENDA

SECURE-iC THE SECURITY SCIENCE COMPANY

Security (SCA*) vs PPA

Common Criteria

Eurosmart PP 0117



ISO/IEC 19790:2012

Information technology — Security techniques — Security requirements for cryptographic modules

Published (Edition 2, 2012)

This standard was last reviewed and confirmed in 2018. Therefore this version remains current.

→ Expected to be replaced by ISO/IEC FDIS 19790 within the coming months.

FIPS 140-3™

UNCLASSIFIED / NON CLASSIFIÉ

ISO/IEC WD 19790:2022(E)

Annex F
(normative)

Approved non-invasive attack mitigation test metrics

Purpose

This Annex provides a list of the ISO/IEC approved non-invasive attack mitigation test metrics applicable to this document. This list is not exhaustive.

This does not preclude the use of approval authority approved non-invasive attack mitigation test metrics.

An approval authority may supersede this Annex in its entirety with its own list of approved non-invasive attack mitigation test metrics.

F.1.1 Non-invasive attack mitigation test metrics

a) ISO/IEC 17825 *Information technology – Security techniques – Testing methods for the mitigation of non-invasive attack classes against cryptographic modules.*

The TOE shall avert the threat "Inherent Information Leakage (T.Leak-Inherent)", as follows:

T.Leak-Inherent	Inherent Information Leakage An attacker may exploit information , as user data or TSF data, which is leaked from the TOE and/or the SoC interfaces while being stored and/or processed by the TOE.

Leakage may occur through emanations, variations in power consumption, response times, clock frequency, or similar variations in the behaviour, based on the data processed by the TOE. This leakage is related to measurement of operating parameters, which may be derived either from measurements of internal and/or external supply signals and/or measurement of emanations and/or IO signal. These operating parameters can then be matched to the specific operations inside the TOE. Examples of such attacks are Differential Power Analysis and Timing Attacks (8 in Figure 5), or analysis of emanation (7 in Figure 5).

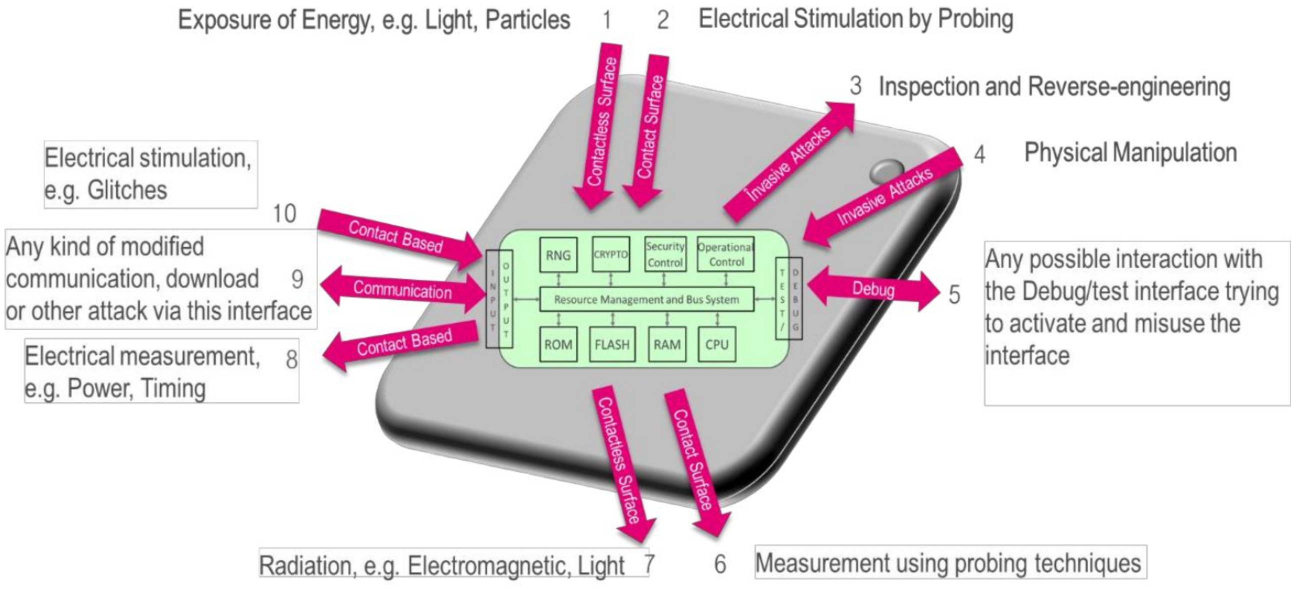


Figure 5: Attacks against the TOE

*** SCA: Side-Channel Analysis**

SECURE-IC
THE SECURITY SCIENCE COMPANY

Security (FIA*) vs PPA

Common Criteria®

Eurosmart PP 0117

Table 1 - Summary of security requirements

	Security Level 1	*Security Level 2*	*Security Level 3*	*Security Level 4*
Physical Security	Production-grade components.	Tamper evidence. Opaque covering or enclosure.	Tamper detection and response for covers and doors. Strong enclosure or coating. Protection from direct probing. EFP or EFT.	Tamper detection and response envelope. EFP. Fault injection mitigation.
Mitigation of other attacks	Specification of mitigation of attacks for which no testable requirements are currently available.			Specification of mitigation of attacks with testable requirements.

FIPS 140-3™

Fault injection mitigation

Mitigation of other attacks

Environment Failure Testing (EFT)

Environment Failure Protection (EFP)

The TOE shall avert the threat “Forced Information Leakage (T.Leak-Forced)“ as specified below.

T.Leak-Forced	Forced Information Leakage An attacker may disclose user data or TSF data, which is leaked from the TOE when such data is processed or stored by the TOE even if the information leakage is not inherent but caused by the attacker by influencing the TOE or the hosting SoC.

This threat pertains to attacks where environmental stress or physical manipulation is applied to the TOE or the hosting SoC to cause leakage from signals which do not compromise user data or TSF data during normal operation. This threat pertains to attacks where methods described in “Malfunction due to Environmental Stress” (see T.Malfunction) and/or “Physical Manipulation” (see T.Phys-Manipulation) are used to cause leakage from signals (Numbers 5, 6, 7, 8 or 9 in Figure 5) that normally do not contain significant information about secrets.

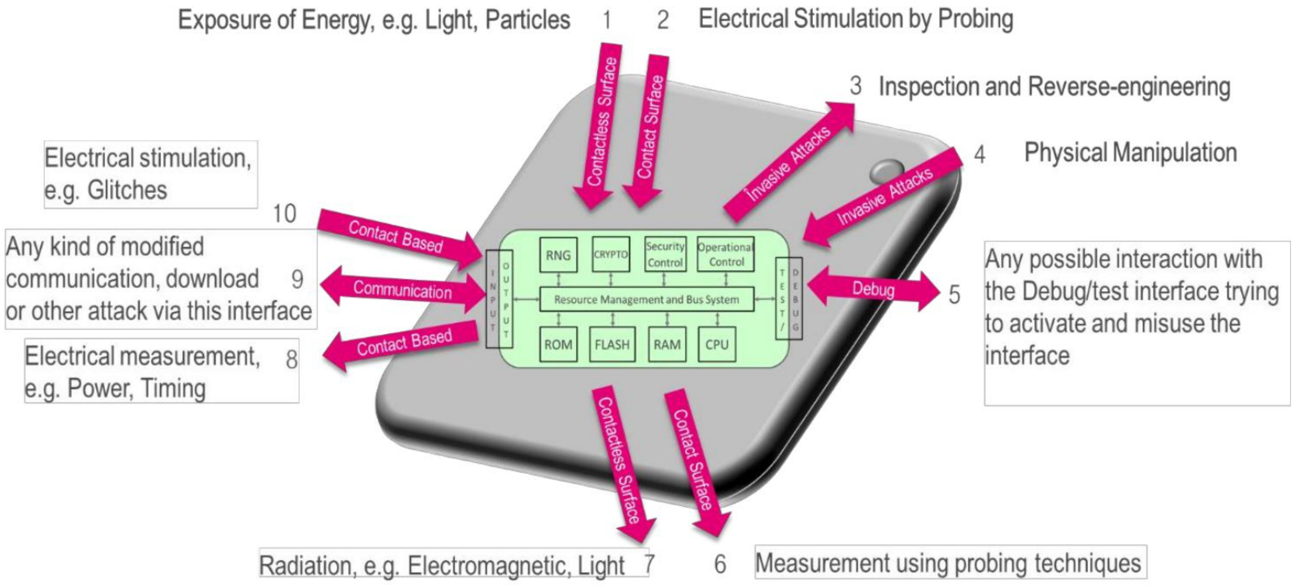


Figure 5: Attacks against the TOE

* FIA: Fault Injection Analysis

AGENDA

1. Introduction
2. Protecting Cryptographic Algorithms with Random Masking
3. Fourier Transform
4. Application to AES
5. Conclusions and Perspectives

Masking for Cryptographic Operations (for instance AES)

Universal computation as extrapolation in ($\mathbf{F}_{256}$, +, *): For example, `SubBytes`(x)

$$63 + 8f\, x^{127} + b5\, x^{191} + 01\, x^{223} + f4\, x^{239} + 25\, x^{247} + f9\, x^{251} + 09\, x^{253} + 05\, x^{254}$$ [1]

Masking = computing on random variables:

- Sharing each byte into *d* bytes
  - $x \rightarrow (x_0, x_1, \ldots, x_{n-1})$
- Linear operations are trivially computed masked:
  - $x+y \rightarrow (x_0+y_0, x_1+y_1, \ldots, x_{n-1}+y_{n-1})$, and $x^2 \rightarrow (x_0^2, x_1^2, \ldots, x_{n-1}^2)$ when in $\mathbf{F}_q$ with $q = 2^m$
- Non-linear operations, such as multiplication, are harder to compute, and make up the bulk of the computation time

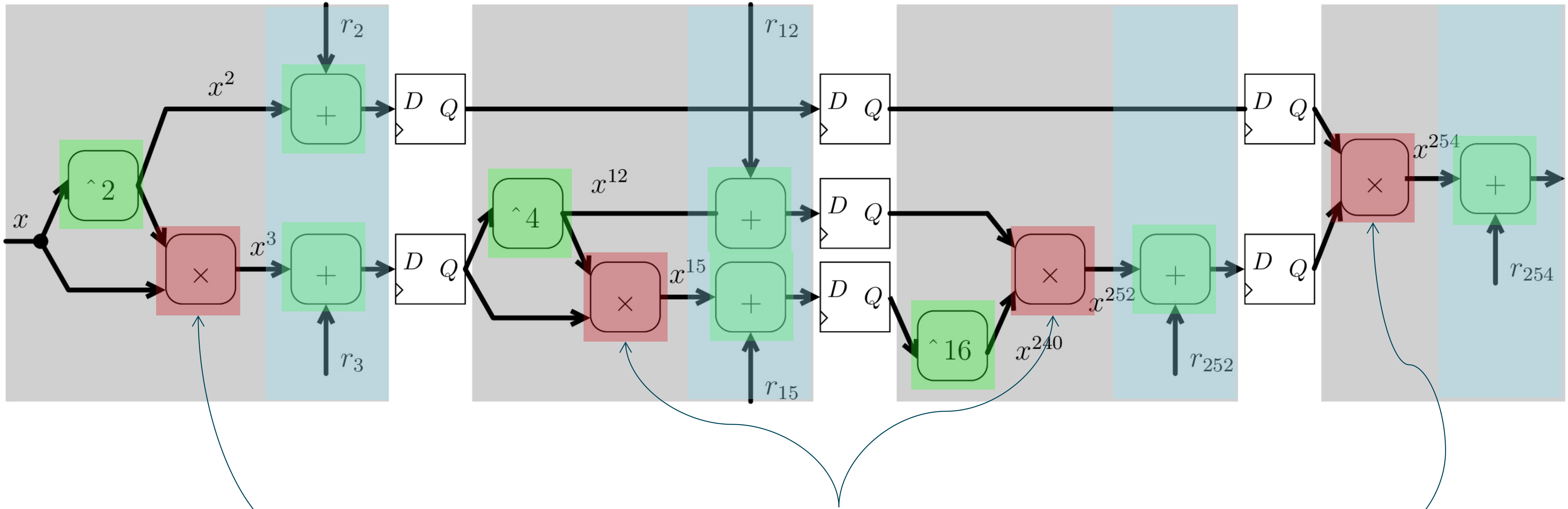
[1] Joan Daemen, Vincent Rijmen: The Rijndael Block Cipher -- AES Proposal, Document version 2, dated 03/09/99

Masking the multiplication on F_{256}

Caption:

- Linear operations
- Non-linear operations
- Refresh stage (addition of 0)

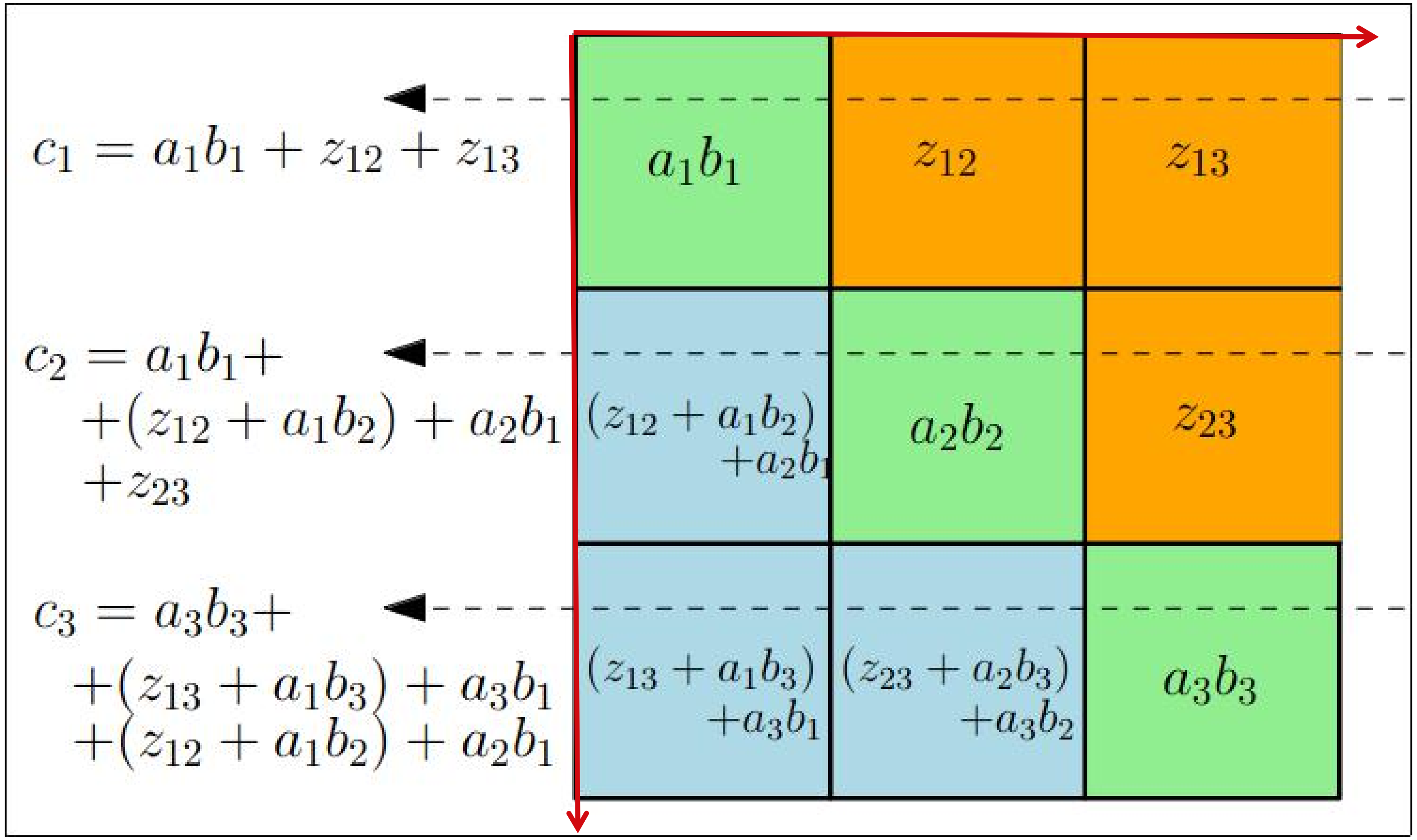
Function $x \rightarrow x^{-1} = x^{254}$:



Bottleneck is complexity of the multiplication operations

State of the art: Quadratic Complexity

Implementation: all cross-product terms are computed, hence a **quadratic** complexity



ISW algorithm:

Require: s-shares $\mathbf{a}$ and $\mathbf{b}$
Ensure: s-shares $\mathbf{c}$ satisfying $c = ab$
  for i from 1 to s **do**
    for j from $i + 1$ to s **do**
      $z_{ij} \leftarrow \texttt{rnd()}$
      $z_{ji} \leftarrow (z_{ij} \oplus a_i b_j) \oplus a_j b_i$
    end for
  end for
  for i from 1 to s **do**
    $c_i \leftarrow a_i b_i$
    for j from 1 to s, $j \neq i$ **do**
      $c_i \leftarrow c_i \oplus z_{ij}$
    end for
  end for

Reparaz, O., Bilgin, B., Nikova, S., Gierlichs, B., Verbauwhede, I. (2015). Consolidating Masking Schemes. In: Gennaro, R., Robshaw, M. (eds) Advances in Cryptology -- CRYPTO 2015. CRYPTO 2015. Lecture Notes in Computer Science(), vol 9215. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-47989-6_37

AGENDA

Discrete Fourier Transform (DFT) Matrix

- Base field is $\mathbf{F}_q$, where $q = p^m$
  - *e.g., p=2 and m=8 for AES, or p=3329 and m=1 for ML-KEM, etc.*
- We are interested in n-point transform
- Naïve multiplication by M (Fourier transform) or M^{-1} (inverse Fourier transform) has complexity n^2
- Shape of M is:

$$M = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{pmatrix}$$

  - where $\omega^n = 1$.

Roots of Unity in Finite Fields

Let $\omega \in \mathbb{F}_q$ an nth root of unity. This means that $\omega^n = 1$.

Lemma 1. *Let M the $n \times n$ matrix of element $M_{i,j}$ equal to ω^{ij}. Then M^{-1} is equal to $M^{-1}_{i,j} = \frac{1}{n}\omega^{-ij}$.*

Proof. Let $\tilde{M}$ be the matrix of elements $\frac{1}{n}\omega^{-ij}$ at position (i,j). The element at position (i,j) of matrix $M\tilde{M}$ is:

$$\sum_{k=0}^{n-1} M_{i,k}\tilde{M}_{k,j} = \sum_{k=0}^{n-1} \omega^{ik}\frac{1}{n}\omega^{-kj} = \frac{1}{n}\sum_{k=0}^{n-1} \left(\omega^{i-j}\right)^k.$$

Then there are two situations:

1. $i = j$, then the sum is equal to 1;
2. $i \neq j$, then $\omega^{i-j} \neq 0$, and the sum is that of a geometric series, equal to $\frac{1}{n}\frac{1-\left(\omega^{i-j}\right)^n}{1-\left(\omega^{i-j}\right)} = 0$, because $\left(\omega^{i-j}\right)^n = \left(\omega^n\right)^{i-j} = 1^{i-j} = 1$.

Thus $\tilde{M}$ coincides with M^{-1}. □

Fourier Transform Matrices, Foreward & Backward

$$M = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{pmatrix}.$$

Vandermonde matrix

$$M^{-1} = \frac{1}{n} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \dots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \dots & \omega^{-(n-1)^2} \end{pmatrix}.$$

Inverse is also a Vandermonde matrix

Fast Fourier Transform Methodology

- We make use of special algebraic structures of field fields

Let $\vec{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n$. Then $DFT(\vec{a}) = \vec{b} = (b_0, \ldots, b_{n-1}) \in \mathbb{F}_q^n$ is such that:

$$b_j = \sum_{i=0} a_i \omega^{ij} = \sum_{i=0} a_i \left(\omega^j\right)^i .$$

Let $P(X) = \sum_{i=0}^{n-1} a_i X^i$. We notice that b_j, for $0 \leq j < n$, can be rewritten as:

$$b_j = P(\omega^j) = P(X) \bmod (X - \omega^j) .$$

The complexity of computing n polynomial modular divisions is still quadratic. Indeed, a polynomial Euclidean division of a polynomial of degree $n-1$ by a polynomial of unitary degree is n, as shown below:

$$\begin{array}{cccccc|c}
a_{n-1}X^{n-1} & + & a_{n-2}X^{n-2} & + & a_{n-3}X^{n-3} & + \ldots + a_0 & X - \omega^j \\
\hline
-a_{n-1}X^{n-1} & + & a_{n-1}\omega^j X^{n-2} & & & & a_{n-1}X^{n-2} \quad \backslash \\
 & - & (a_{n-2} + a_{n-1}\omega^j)X^{n-2} & + & (a_{n-2} + a_{n-1}\omega^j)\omega^j X^{n-3} & & +(a_{n-2} + a_{n-1}\omega^j)X^{n-3} \quad \backslash \\
 & & & - & \ddots & \ddots & + \vdots \quad \backslash
\end{array}$$

Fast Fourier Transform Complexity Analysis

The colors indicate the number of products in $\mathbb{F}_q$. In the case of the division of one dividend of degree $n-1$ by a divisor of degree 1, one field multiplication (colored above) is needed per reduction step, hence in total $n-1$ multiplications are required.

Let us recall in general how Euclidean division works:

- in the case of the division of a dividend of degree d_1 by a divisor of degree $d_2 \leq d_1$ (with unitary leading coefficient),
- the result is a quotient of degree $d_2 - d_1$ and a remainder of degree $d_1 - 1$;
- the number of steps in the Euclidean division is $d_2 - d_1$,
- and involves $d_1 - 1$ (since the leading coefficient is $= 1$) multiplications,
- hence a total of $(d_2 - d_1)(d_1 - 1)$ field multiplications.

This is still quadratic!

Fast Fourier Transform Rationale

- We make use of special algebraic structures of field fields

If the divisor has only one coefficient which is nonzero and not equal to one, then the number of field multiplication per Euclidean division step becomes only one, and therefore the overall complexity of the division is $(d_2 - d_1)$.

Hence we try to compute the reductions hierarchically, by noting that:

$$(X-0)\prod_{i=0}^{n-1}(X-\omega^i) = X^n - X \ ,$$

and that:

$$\big(P(X) \bmod (X-\omega^i)\underbrace{(X-\omega^j)}\big) \bmod \underbrace{(X-\omega^j)} = P(X) \bmod \underbrace{X-\omega^j} \ .$$

For instance, let us consider: $n = 3$, in the field $\mathbb{F}_4$. We denote:

- $q_{2,0}(X) = X^4 - X = (X - 0)(X - \omega)(X - \omega^2)(X - \omega^3)$,
- $q_{1,0}(X) = X(X - \omega)$ and $q_{1,1}(X) = (X - \omega^2)(X - \omega^3)$,
- $q_{0,0}(X) = X$, $q_{0,1}(X) = X - \omega$, $q_{0,2}(X) = X - \omega^2$ and $q_{0,3}(X) = X - \omega^3$.

Then, the computation of b_j from $P(X) = \sum_{i=0}^{n-1} a_i X^i$ can be achieved as:

- $P_{2,0}(X) = P(X) \bmod q_{2,0}(X) = P(X)$,
- $P_{1,0}(X) = P_{2,0}(X) \bmod q_{1,0}(X)$ and $P_{1,1}(X) = P_{2,0}(X) \bmod q_{1,1}(X)$,
- $P_{0,0} = P_{1,0} \bmod q_{0,0}(X)$, $P_{0,1} = P_{1,0} \bmod q_{0,1}(X)$, $P_{0,2}(X) = P_{1,1} \bmod q_{0,2}(X)$ and $P_{0,3}(X) = P_{1,1} \bmod q_{0,3}(X)$,

where we end by the following affectation:

- $b_0 = P_{0,0}$,
- $b_1 = P_{0,1}$,
- $b_2 = P_{0,2}$,
- $b_3 = P_{0,3}$.

Fast Fourier transform

Algorithm 1: Quasi-linear (i.e., fast) Discrete Fourier Transform

Data: Pre-computed binary tree $q_{i,j}$

Input: $a = (a_0, a_1, \ldots, a_{n-1})$

Output: $(b_0, b_1, \ldots, b_{n-1})$ the DFT of a

$P_{\lceil \log_2(n) \rceil, 0} \leftarrow \sum_{i=0}^{n-1} a_i X^i$

for $i \in \{\lceil \log_2(n) \rceil - 1, \lceil \log_2(n) \rceil - 2, \ldots, 0\}$ **do** // Depth i of $\log_2(n)$

  for $j \in \{1, \ldots, 2^{\lceil \log_2(n) \rceil - i}\}$ **do** // Breadth of $n/2^i$

    $P_{i,j} \leftarrow P_{i+1, \lfloor j/2 \rfloor} \bmod q_{i,j}$ // Complexity of 2^i

return $(P_{0,j})_{0 \le j \le n-1} = (b_0, b_1, \ldots, b_{n-1})$

Fast Fourier Transform: Quasi-Linear Complexity

Now, it is possible to reorder the leafs so that the $q_{i,j}(X)$ polynomials are *linearized* or *affine*, with all coefficients but one in $\mathbb{F}_q$ (all others belonging to $\mathbb{F}_p = \mathbb{F}_2$) [1, main theorem, page 513].

The complexity of this hierarchical computation is given below:

Level	Number of reductions	Degree of dividend	Degree of divisor	Complexity
Generic	r	d_1	d_2	$r \times (d_1 - d_2)$
$\lceil \log_2(n) \rceil$	1	$n-1$	n	0
$\lceil \log_2(n) - 1 \rceil$	2	$n-1$	$n/2-1$	$2 \times n/2 = n$
$\lceil \log_2(n) - 2 \rceil$	4	$n/2-1$	$n/4-1$	$4 \times n/4 = n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	$2^{\lceil \log_2(n) \rceil}$	$\frac{n}{2^{\lceil \log_2(n)-1 \rceil -1}} - 1$	$\frac{n}{2^{\lceil \log_2(n) \rceil}} - 1$	n
Total	....			$n \times \lceil \log_2(n) \rceil$

[1] R. E. Blahut, Theory and Practice of Error Control Codes. Reading, MA: Addison-Wesley, 1983.

AGENDA

Settings

Let ν be a primitive element of $\mathbb{F}_q$, that is a generator of the multiplicative group $\mathbb{F}_q^*$. Let n be a positive integer. We assume that n divides $q-1$, then we have that the field element $\omega = \nu^{\frac{q-1}{n}}$ is a root of the unity (i.e. $\omega^n = 1$). By construction, n is odd with q is power of two. We denote $n = 2d + 1$.

Verification in MAGMA

```
F<alpha> := PolynomialRing(GF(2));
P := F ! alpha^8+alpha^4+alpha^3+alpha+1;
GF256<X> := ext< GF(2) | P >;

nu := PrimitiveElement(GF256); // X+1
omega := nu^85; // 85 = 255/3  // X^7+X^5+X^4+X^3+X^2+1

Order( nu    ); // 255 = 3*5*17 = 3*85
Order( omega ); // 3 = 255/85
```

Fast Fourier Transform: Data Preparation

- Representation:

$$n = 2d + 1 :$$

Cleartext:

$$\begin{aligned}\vec{a} &= (a_0, \; a_1, \; \ldots, \; a_d, \; a_{d+1}, \; \ldots, \; a_{2d}) \\ &= (x, \; r_0, \; \ldots, \; r_{d-1}, \; 0, \; \ldots, \; 0)\end{aligned}$$

1   d   d

x M   x M^{-1}

Masked:

$$\mathtt{mask}(x) := \mathrm{DFT}(\vec{a}) = \left(\sum_{i=0}^{d} a_i \omega^{ij}\right)_{j\in\{0,\ldots,2d\}} = \vec{a}\cdot M = \vec{z}$$

$$\mathtt{unmask}(\vec{z}) = \mathrm{IDFT}(\vec{z})_0 = \left(\vec{z}\cdot M^{-1}\right)_0 = x$$

- Addition, scaling,
- Multiplication by a constant
- Multiplication

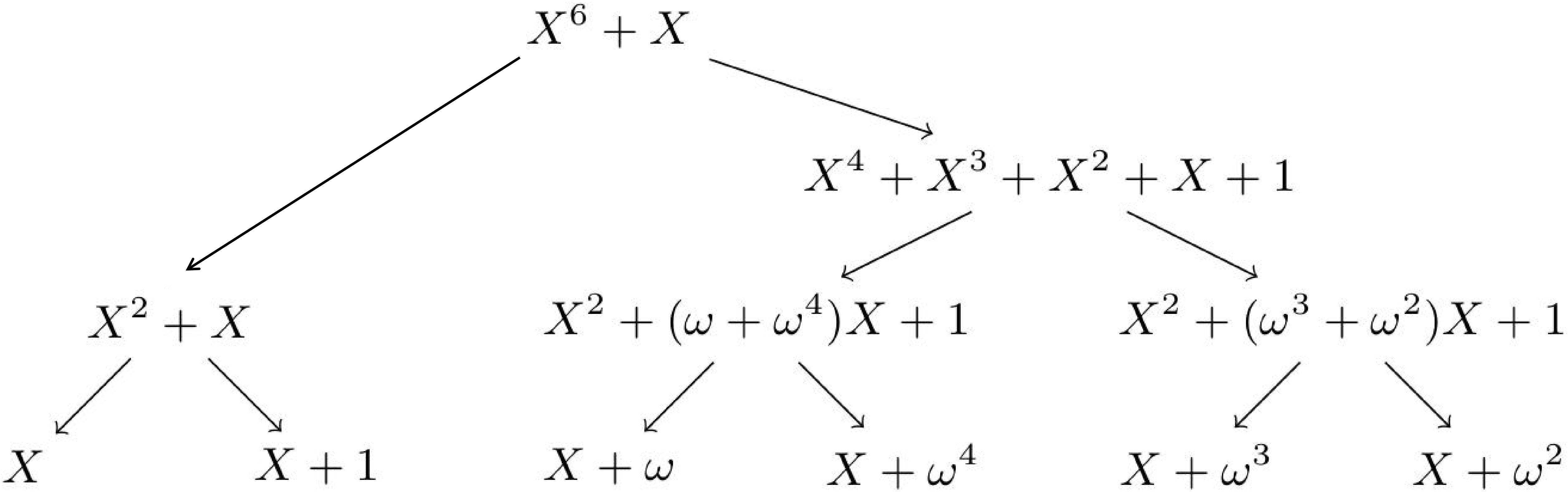
Parameters range

The symmetric encryption algorithm AES is a byte-oriented block cipher. Its design leverages the irreducible polynomial $X^8+X^4+X^3+X+1$. The Sbox is based on the inverse function defined over the finite field $\mathbb{F}_{2^8} = \frac{\mathbb{F}_2[X]}{(X^8+X^4+X^3+X+1)}$. The canonical basis is given by $\alpha = \overline{X}$ in $\mathbb{F}_{2^8}$ and $1+\alpha$ is a primitive element of this field. Then $X^{256} - X = X(X^{255} - 1)$ and $255 = 3 \times 5 \times 17$. We can consider DFT with $n = 3, 5, 15, 17, 51, 85$.

We note that we have not a large choice for n if we keep this method. We will see in the next section that we can construct a DFT and its associate inverse by observing the different trees.

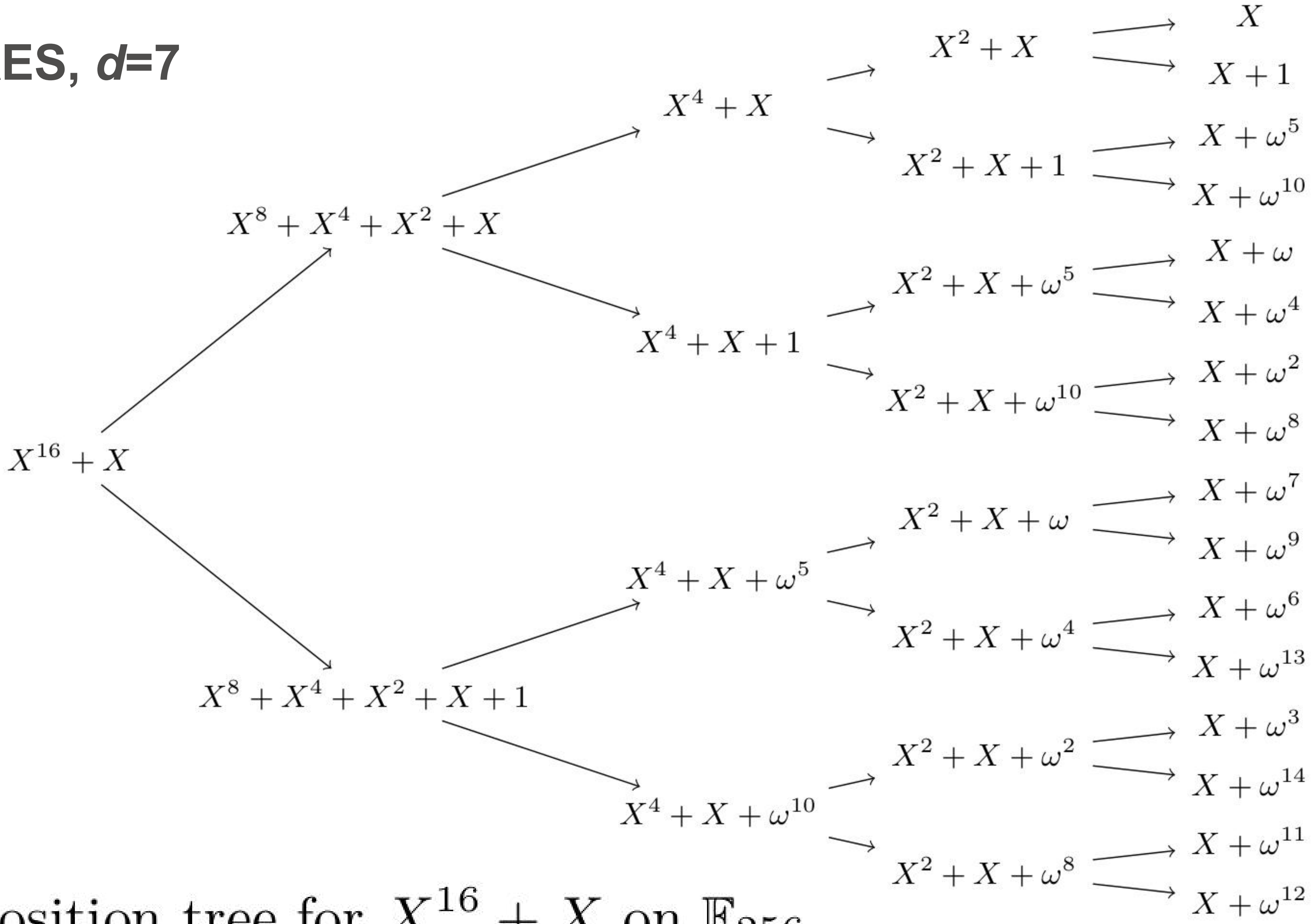
The `SAGE` code and the executable source code in C language are provided in a GitHub: https://github.com/daif-abde/FFT_masking.git.

AES, d=2



Polynomial decomposition tree for $X^6 + X$ on $\mathbb{F}_{256}$.

AES, *d*=7



Polynomial decomposition tree for $X^{16} + X$ on $\mathbb{F}_{256}$.

Homomorphic operations:

- Addition
- Scaling

Let us denote: $\vec{z} = \mathtt{mask}(x)$ and $\vec{z}\,' = \mathtt{mask}(x')$. The following properties are satisfied:

- $\mathtt{mask}(x + x') = \vec{z} + \vec{z}\,'$,
- $\mathtt{mask}(\lambda x) = \lambda \cdot \vec{z} \quad$ for any $\lambda \in \mathbb{F}_q$.

Quasi-Linear Masking without Cost Amortization

Variable	Cost
$\vec{y}$	n
$\vec{r}\,''$	$n + n\log(n)$
$\mathtt{mask}(xx')$	$2n(1+\log(n))$

Multiplication operation:

- Commutative diagram:

Cleartext: **Masked:**

$$(x, r_0, \ldots, r_{d-1}, 0, \ldots, 0) \xrightarrow{\text{DFT}} \vec{z} = \mathtt{mask}(x)$$

$$(x', r'_0, \ldots, r'_{d-1}, 0, \ldots, 0) \longrightarrow \vec{z}\,' = \mathtt{mask}(x')$$

$$\vec{y} = (z_0 z'_0, \ldots, z_{2d} z'_{2d})$$

$$\begin{aligned}\mathtt{mask}(xx') &= \vec{y} - \text{DFT}\left(0, 0, \ldots, 0, \frac{1}{n}\sum_{i=0}^{2d} y_i \omega^{-i(d+1)}, \ldots, \frac{1}{n}\sum_{i=0}^{2d} y_i \omega^{-i(2d)}\right) \\ &= \vec{y} - \text{DFT}\left(0, 0, \ldots, 0, \text{IDFT}(\vec{y})_{d+1,\ldots,2d}\right)\end{aligned}$$

$$(xx', r''_0, \ldots, r''_{d-1}, 0, \ldots, 0) \xleftarrow{\text{IDFT}} \mathtt{mask}(xx')$$

Security Proof

- By reduction from Code-Based Masking (CBM):
  - Weijia Wang, Pierrick Méaux, Gaëtan Cassiers, and François-Xavier Standaert.
    Efficient and private computations with code-based masking. IACR Trans. Cryptogr. Hardw. Embed. Syst., 2020(2):128–171, 2020.

- No such assumption as:

Hypothesis 1 (FFT Probing Security). *The circuits processing*

$$\mathsf{FFT}_{\boldsymbol{\alpha}} : (\boldsymbol{x} \parallel \mathbf{0}) \mapsto \boldsymbol{r} \quad \textit{and} \quad \mathsf{FFT}_{\boldsymbol{\alpha}}^{-1} : \boldsymbol{u}' \mapsto \boldsymbol{t}$$

are t_n^{FFT}*-probing secure w.r.t. the* $\boldsymbol{v}_\omega$*-encoding and the* $\boldsymbol{v}'_\omega$*-encoding respectively.*

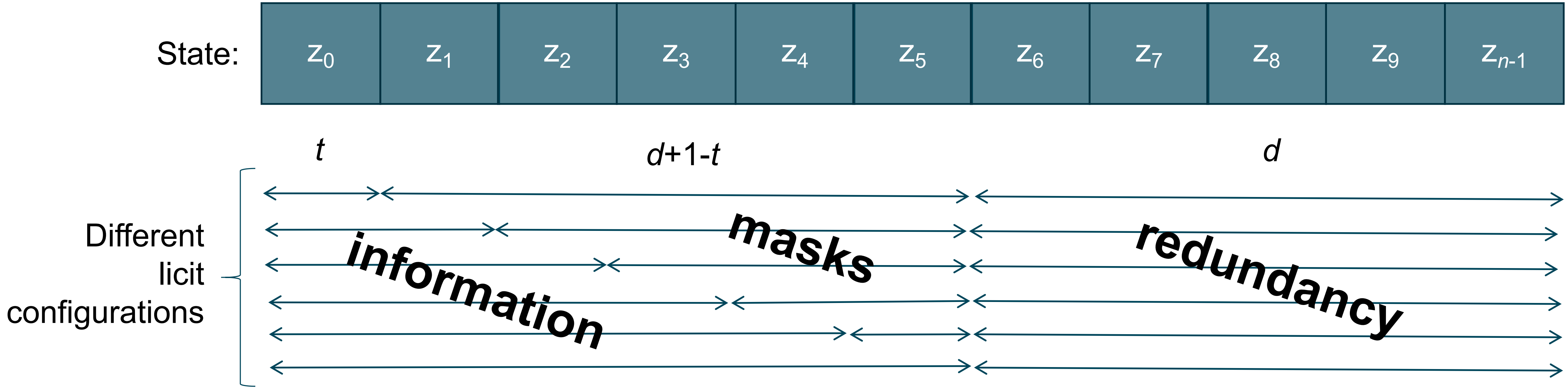
  - In: Probing Security through Input-Output Separation and Revisited Quasilinear Masking. (2021). IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(3), 599-640. https://doi.org/10.46586/tches.v2021.i3.599-640

Quasi-Linear Masking with Cost Amortization

Cost amortization: tradeoff between

- Side-channel order $d+1-t$
  - *versus*
- Amount of processed information simultaneously: t

Fault detection: checking the d MSBs are null



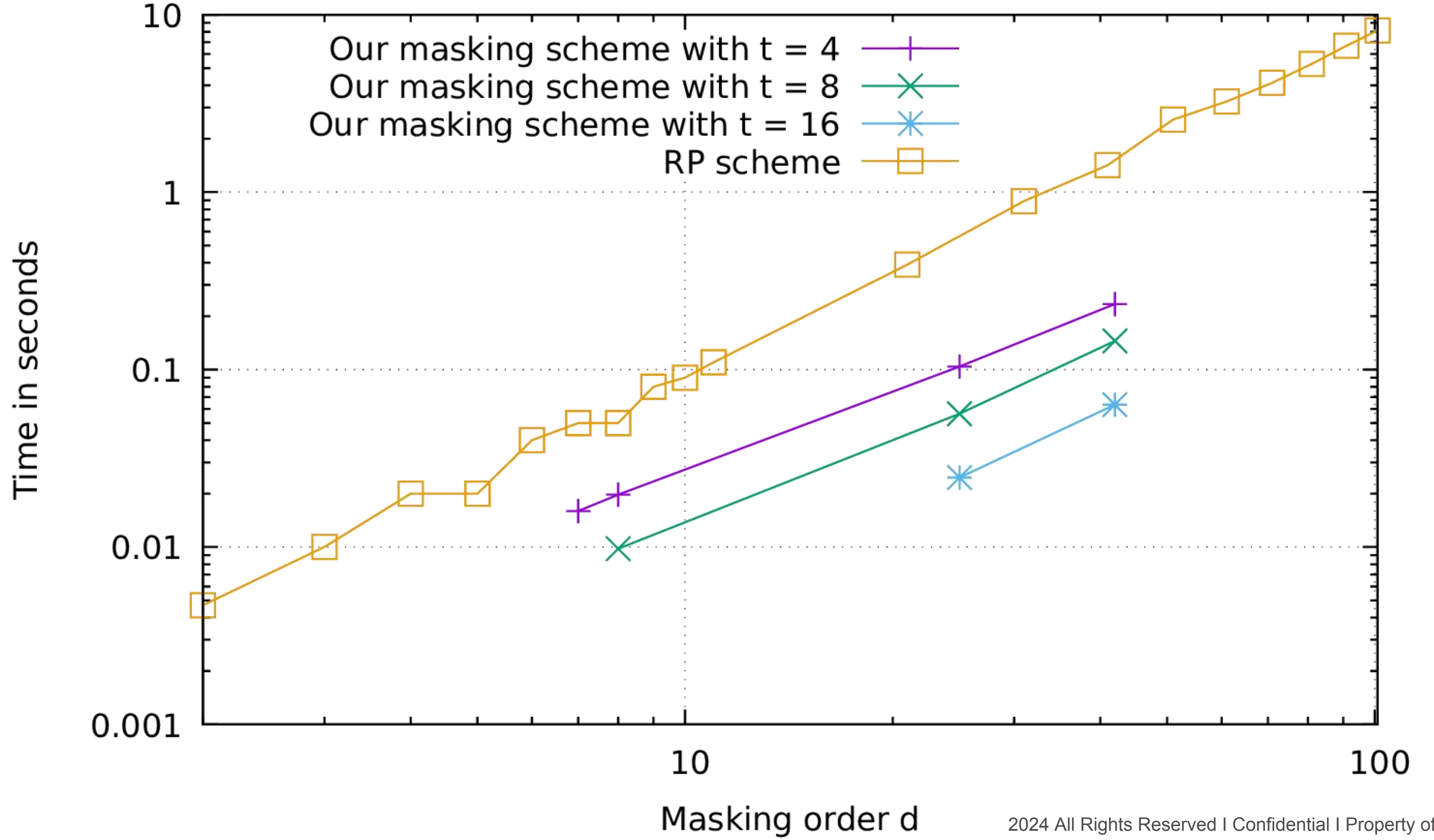
SECURE-IC THE SECURITY SCIENCE COMPANY

Side-Channel Security Order *versus* Fault Detection / Correction, in F_{256}

n	d	t	SCA order $(d+1-t)$	Nb. of detected faults	Nb. of corrected faults
5	2	1	2	2	0
		2	1		
15	7	1	7	7	3
		2	6		
		$\vdots$	$\vdots$		
		7	1		
17	8	1	8	8	3
		2	7		
		$\vdots$	$\vdots$		
		8	1		

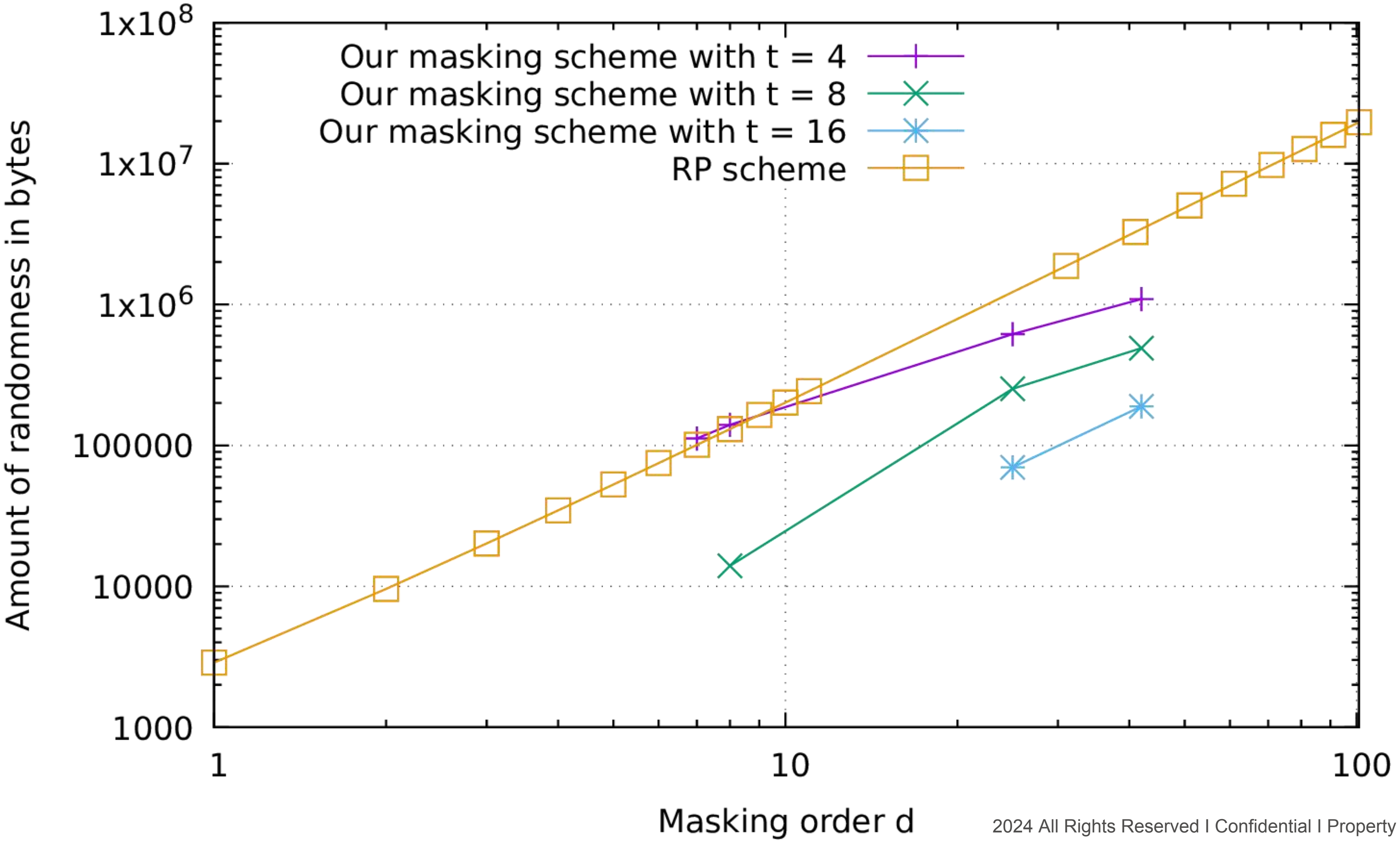
Performance on F_{256} with Cost Amortization

- Computation time for 50 times AES calculation, with pre-calculated multiplication.



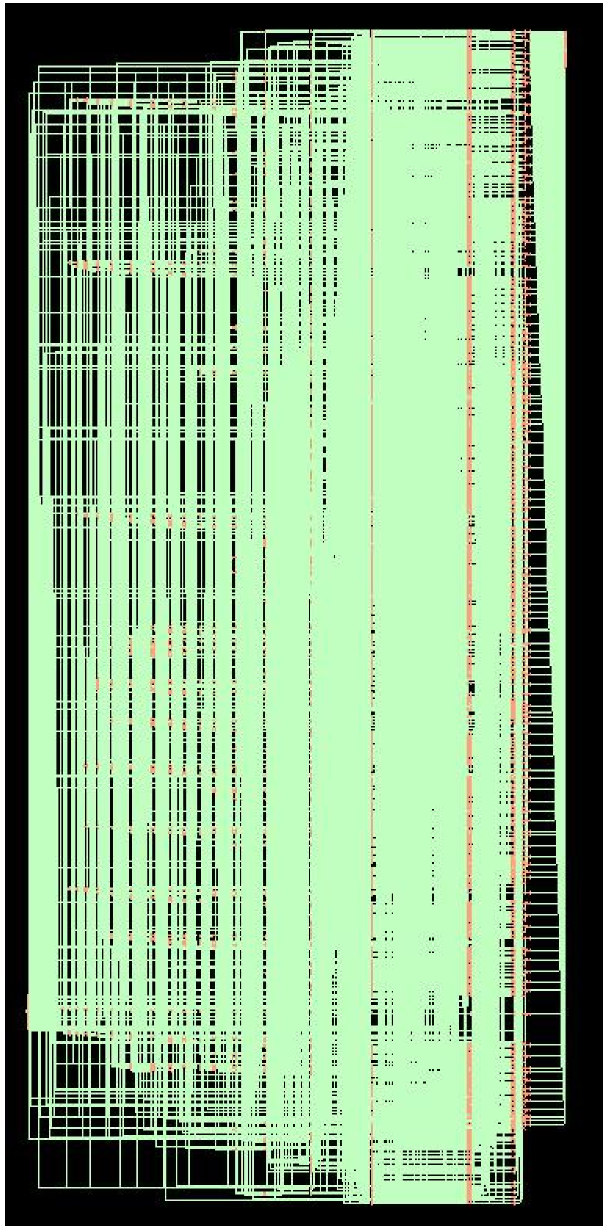
Performance on F_{256} with Cost Amortization

- The amount of randomness generated in terms of bytes.

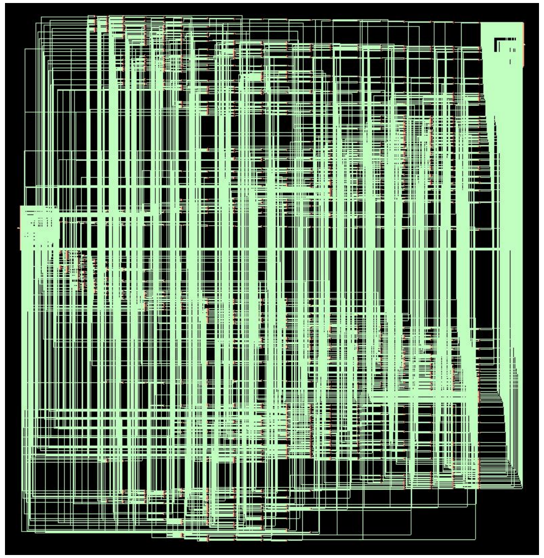


Area saving on F_{256} with Cost Amortization

- ASIC synthesis, using VHDL in 130 nm technology



From Vandermonde to FFT



Comparison of our masking scheme with the state of the art

Scheme name	Side-channel protection		Fault protection		Field
	Complexity	Cost am.	End-to-end	Detection	
ParTI [SMG16]	Quadratic $(\mathcal{O}(d^2))$	No	Yes	At checkpoints	$\mathbb{F}_2$
CAPA [RMB+18]	Quadratic $(\mathcal{O}(d^2))$	No	Yes	At checkpoints	$\mathbb{F}_2$
GJR [GJR18]	Quasi-linear $(\mathcal{O}(d \log d))$	No	No	N/A	$\mathbb{F}_p$
M&M [MAN+19]	Quadratic $(\mathcal{O}(d^2))$	No	Yes	Infective	$\mathbb{F}_2$
DOMREP [GPK+21]	Quadratic $(\mathcal{O}(d^2))$	No	Yes	At checkpoints	$\mathbb{F}_2$
GJR+ [GPRV21]	Quasi-linear $(\mathcal{O}(d \log d))$	No	No	N/A	$\mathbb{F}_q$
CINI MINIS [FRSG22]	Quadratic $(\mathcal{O}(d^2))$	No	Yes	At checkpoints	$\mathbb{F}_2$
RTIK [Pla22]	Polynomial $(\mathcal{O}(d^{\log_2 3}))$	No	No	N/A	$\mathbb{F}_2$
SotA / laOla [BEF+23]	Quadratic $(\mathcal{O}(d^2))$	No	Yes	At checkpoints	$\mathbb{F}_q$
Our work	Quasi-linear $(\mathcal{O}(d \log d))$	Yes	Yes	At checkpoints	$\mathbb{F}_q$

AGENDA

Conclusions and Perspectives

We achieve such masking protection:

- Minimizing the number of multiplications
- Cost amortization and fault detection capability
- Quasi-linear masking complexity
- Code-Based Masking (CBM) compliant

Code available online:

- https://github.com/daif-abde/FFT_masking

Perspectives:

- Application to Crystals Kyber (q = 3329):
  - The values of n are $\{2^i, 2 \leq i \leq 8\} \cup \{13 \cdot 2\, i, 0 \leq i \leq 7\}$.

Acknowledgments:

- BPI, project **X7PQC** (project call "Cryptographie post quantique", held by the National Quantum Strategy "*Develop the post-quantum cryptographical offering*" and *the National Cyber Strategy "Development of innovative and critical cyber technologies*").
- The Research Council of Norway

THANK YOU FOR YOUR ATTENTION

CONTACTS

EMEA	sales-EMEA@secure-IC.com
APAC	sales-APAC@secure-IC.com
CHINA	sales-CHINA@secure-IC.com
JAPAN	sales-JAPAN@secure-IC.com
AMERICAS	sales-US@secure-IC.com