riscure

Applying Horizontal Clustering Side-Channel Attacks on Embedded ECC Implementations

E. Nascimento and L. Chmielewski

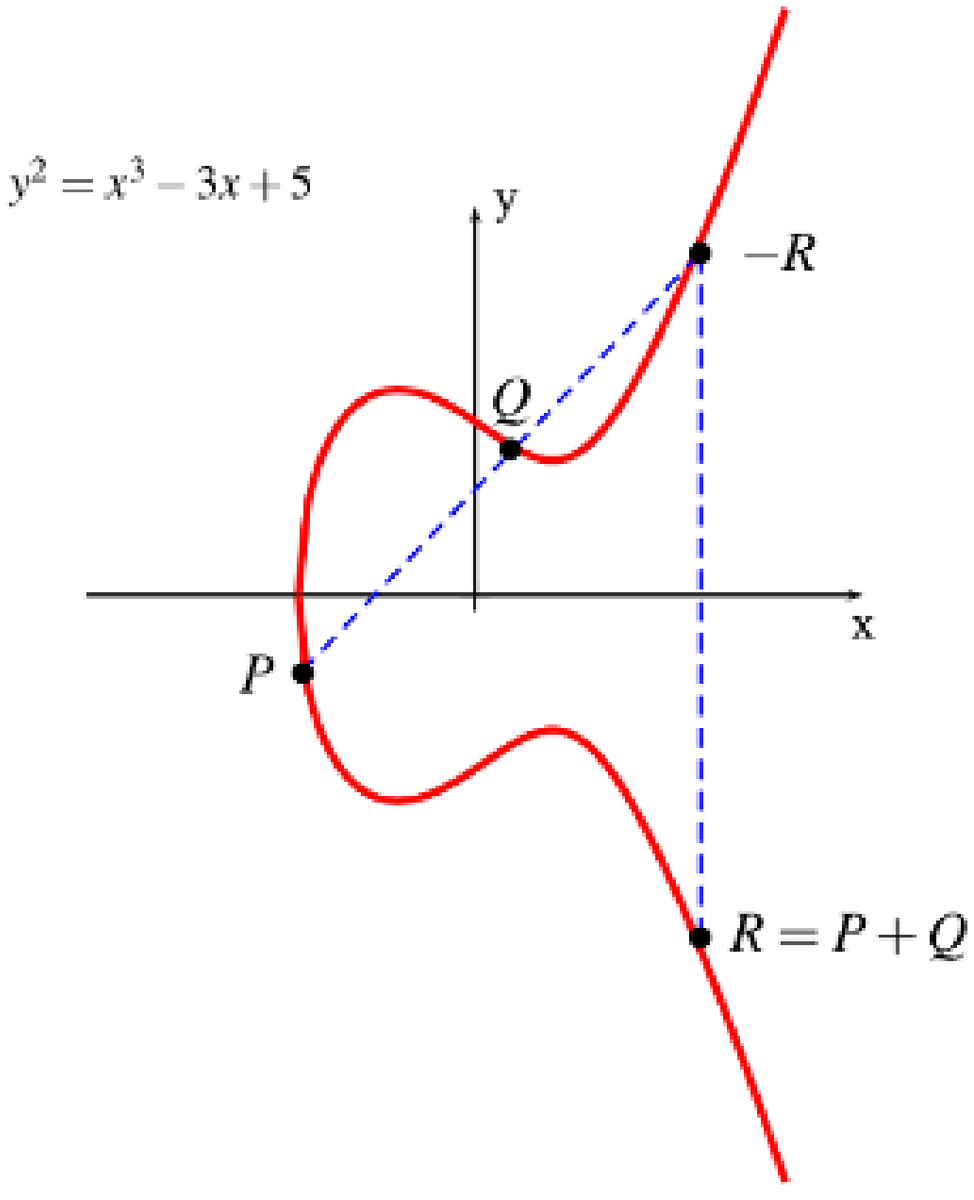
November 15th, CARDIS 2017

Outline

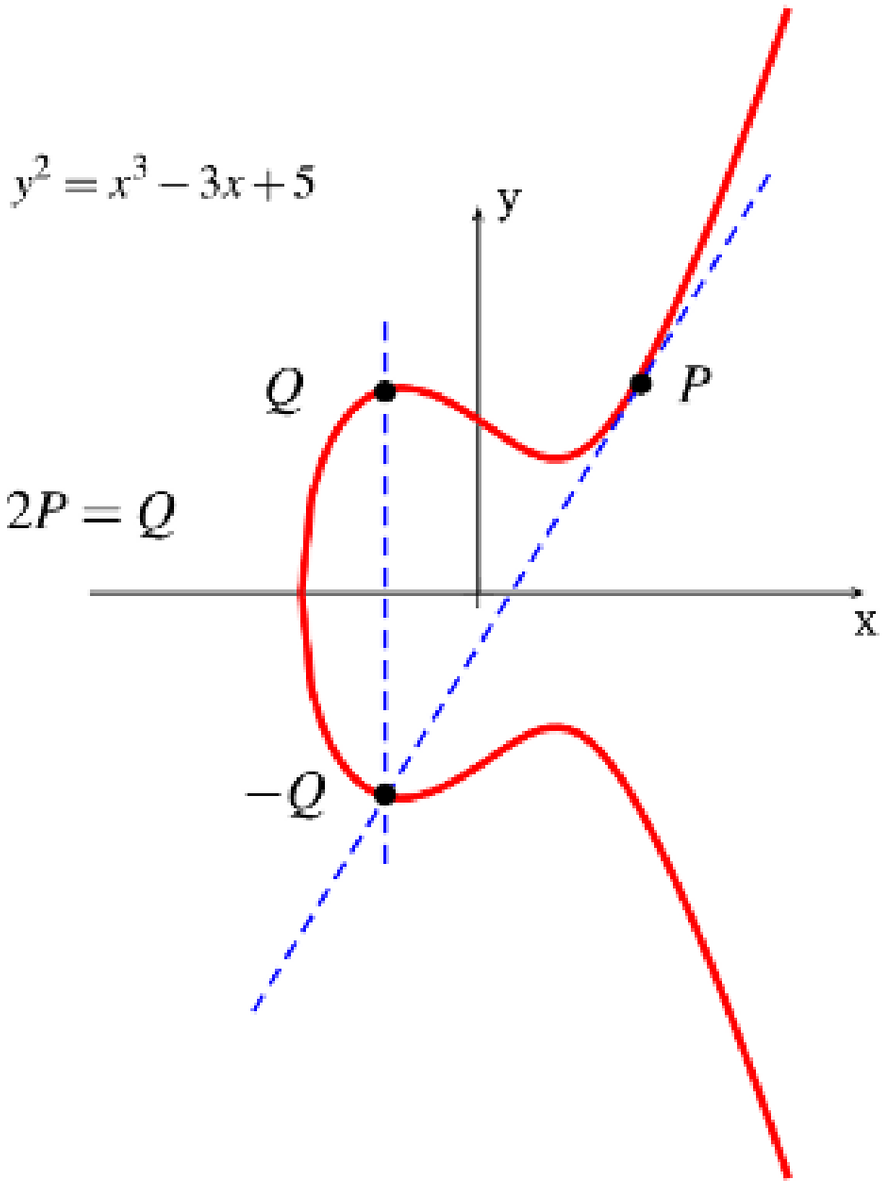
- Goal: practical attack on heavily protected implementation
- Introduction
  - ECC scalar multiplication, horizontal attacks, target, contributions...
- Attack Framework
- Attack Results
- Conclusions & Future Work

ECC

Addition



Doubling



Attacked Curve / Protocol

- Bernstein proposed Curve25519 and the associated X25519 Diffie-Hellman Key Exchange protocol in 2006.
- Curve25519 is an elliptic curve in the Montgomery form with equation

$$E(\mathbb{F}_p) : y^2 = x^3 + 48662x^2 + x$$

over the prime finite field F_p , $p = 2^{255} - 19$ (pseudo-Mersenne).

RSA → ECC

- (n, e) Public key → the curve description + public points
- (n, d) Private key → integer s

- Message: m → P = (x,y)
- Decryption: $m = c^d \bmod n$ → M = [s] P

- Having public data and m it is hard to get d / s from m / M.

- A lot of implementation details omitted:
  - Montgomery Reduction,
  - Affine, Projective, … coordinates (compress / decompress)

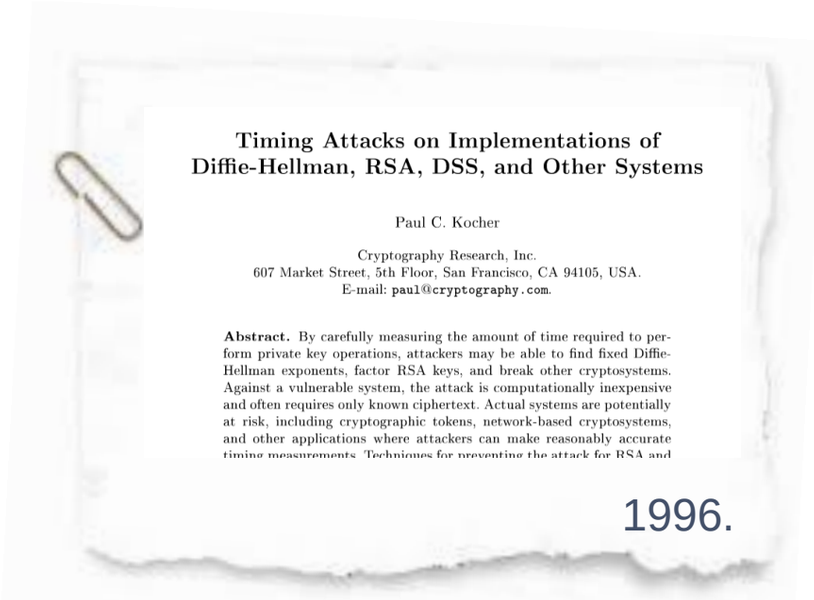
Simple Power Analysis on ECC

$s = \sum_{i=0}^{n-1} s_i \, 2^i$ (scalar in ***binary*** base)

```
sc_mult(P){
    A = 0
    for ( i = n-1 ; i>0 ; i++ )
        A = 2A
        if (d_i = =1)
            A = A+P
        end if
    end for
    Return A = [s] P
}
```

D

A



Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems

Paul C. Kocher

Cryptography Research, Inc.
607 Market Street, 5th Floor, San Francisco, CA 94105, USA.
E-mail: paul@cryptography.com.

Abstract. By carefully measuring the amount of time required to perform private key operations, attackers may be able to find fixed Diffie-Hellman exponents, factor RSA keys, and break other cryptosystems. Against a vulnerable system, the attack is computationally inexpensive and often requires only known ciphertext. Actual systems are potentially at risk, including cryptographic tokens, network-based cryptosystems, and other applications where attackers can make reasonably accurate timing measurements. Techniques for preventing the attack for RSA and

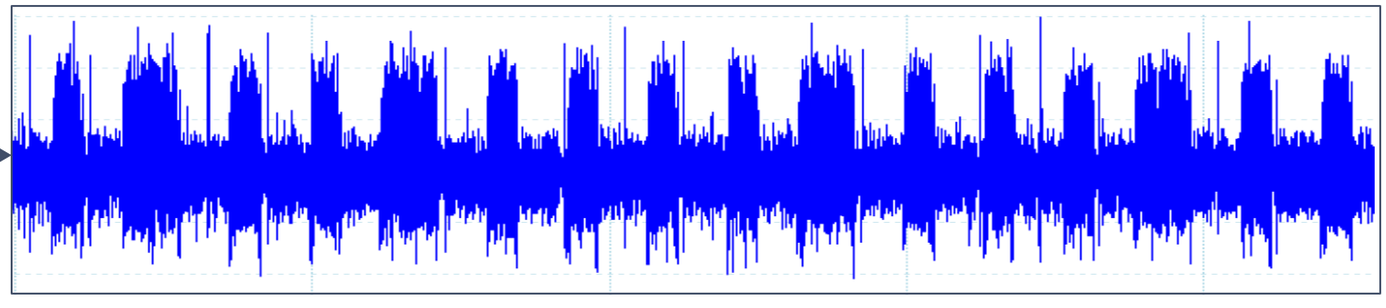
1996.

"By carefully measuring the *amount of time* required to perform private key operations, attackers may be able to find […] RSA keys."

Current Probe

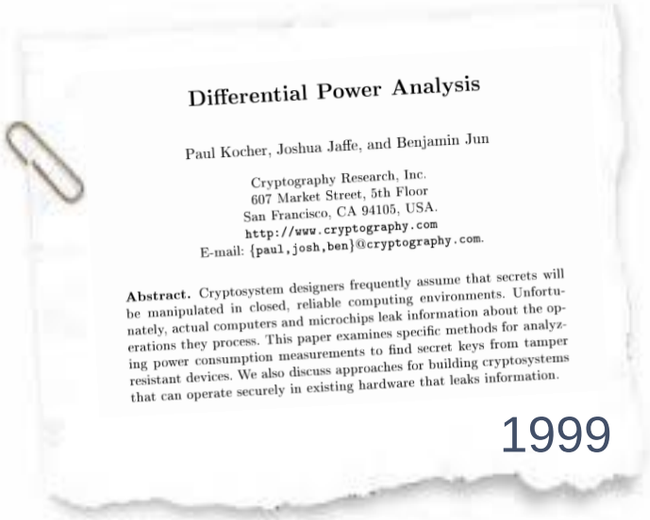
1 0 1 0 0 0 1 0 0 1 0

… D A D D A D D D D A D D D A A A …



Differential Power Analysis and Correlation Power Analysis on ECC

riscure



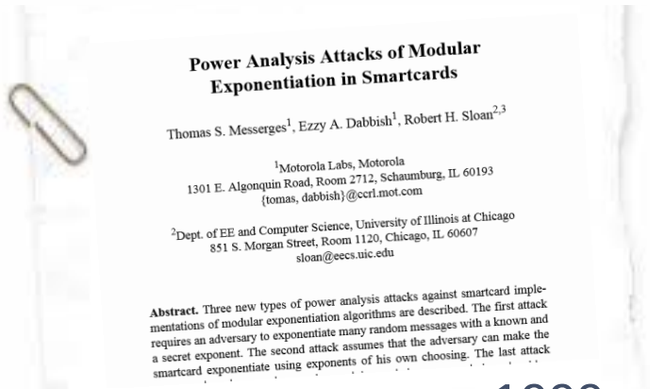
Differential Power Analysis

Paul Kocher, Joshua Jaffe, and Benjamin Jun

Cryptography Research, Inc.
607 Market Street, 5th Floor
San Francisco, CA 94105, USA.
http://www.cryptography.com
E-mail: {paul,josh,ben}@cryptography.com

Abstract. Cryptosystem designers frequently assume that secrets will be manipulated in closed, reliable computing environments. Unfortunately, actual computers and microchips leak information about the operations they process. This paper examines specific methods for analyzing power consumption measurements to find secret keys from tamper resistant devices. We also discuss approaches for building cryptosystems that can operate securely in existing hardware that leaks information.

1999



Power Analysis Attacks of Modular Exponentiation in Smartcards

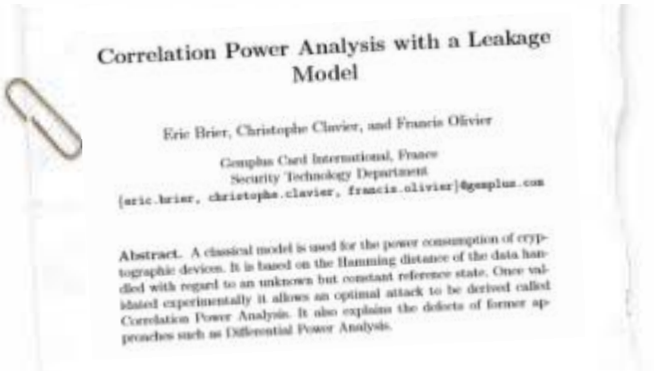
Thomas S. Messerges[1], Ezzy A. Dabbish[1], Robert H. Sloan[2,3]

[1]Motorola Labs, Motorola
1301 E. Algonquin Road, Room 2712, Schaumburg, IL 60193
{tomas, dabbish}@ccrl.mot.com

[2]Dept. of EE and Computer Science, University of Illinois at Chicago
851 S. Morgan Street, Room 1120, Chicago, IL 60607
sloan@eecs.uic.edu

Abstract. Three new types of power analysis attacks against smartcard implementations of modular exponentiation algorithms are described. The first attack requires an adversary to exponentiate many random messages with a known and a secret exponent. The second attack assumes that the adversary can make the smartcard exponentiate using exponents of his own choosing. The last attack

1999



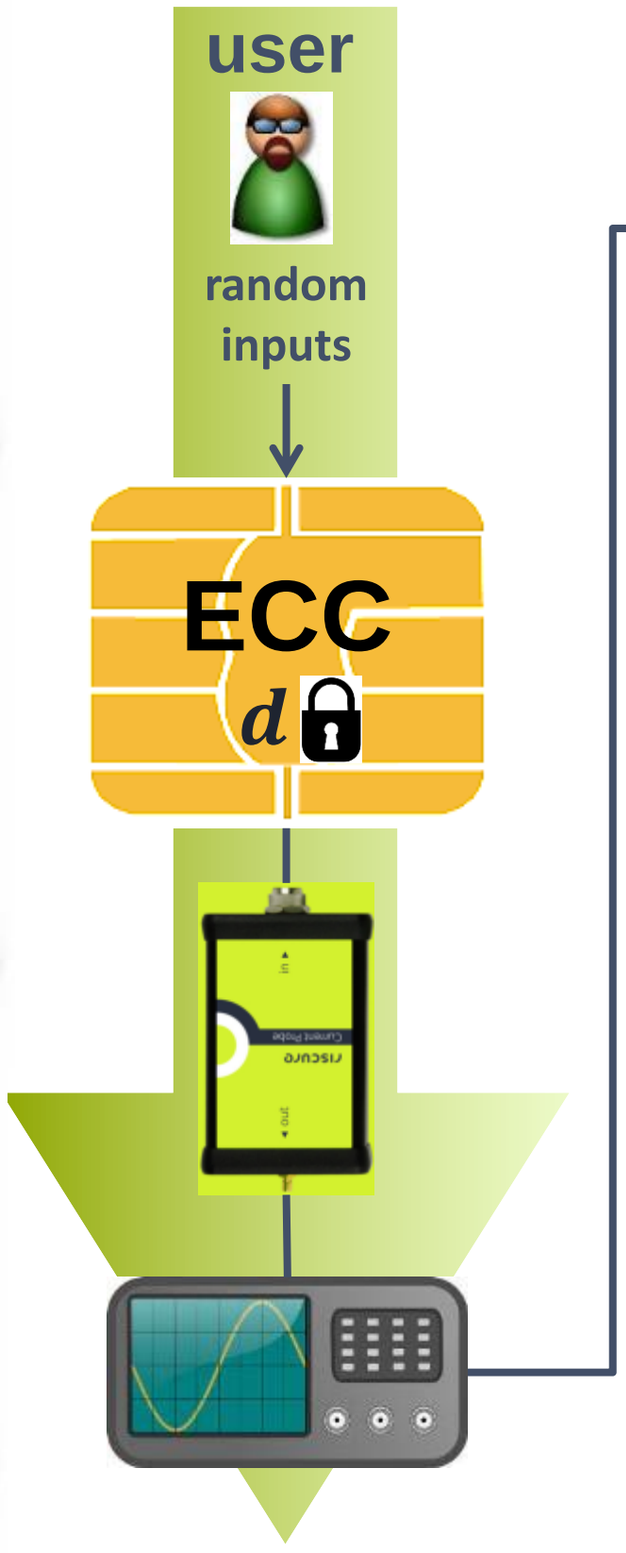
Correlation Power Analysis with a Leakage Model

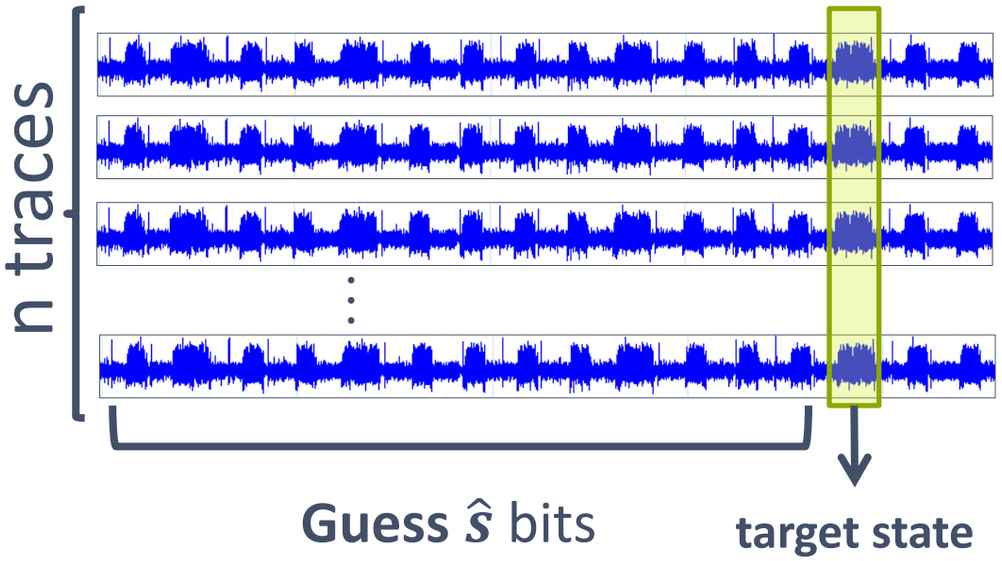
Eric Brier, Christophe Clavier, and Francis Olivier

Gemplus Card International, France
Security Technology Department
{eric.brier, christophe.clavier, francis.olivier}@gemplus.com

Abstract. A classical model is used for the power consumption of cryptographic devices. It is based on the Hamming distance of the data handled with regard to an unknown but constant reference state. Once validated experimentally it allows an optimal attack to be derived called Correlation Power Analysis. It also explains the defects of former approaches such as Differential Power Analysis.

2004





f_i = **Selection Function**(random inputs, $\hat{s}$, target state)

- Hamming weight of a register (**e.g., 16 bits**)
- Hamming distance between current and previous state
- Identity model (value of a register)

$$f_i = \begin{cases} 0 & if\ HW \leq 8 \\ 1 & if\ HW > 8 \end{cases} \qquad f_i = HW(state)$$

$$\text{DPA} = \frac{1}{\sum f_i}\sum_{i=1}^{n} f_i t_i - \frac{1}{\sum(1-f_i)}\sum_{i=1}^{n}(1-f_i)t_i$$

$$\text{CPA} = \rho(trace, f_i)$$

Why do coordinate and scalar randomization protect ECC?

$$M = [s]P = [s](X, Y) = [s](x, y, 1)$$

1. $M = [s](x.z, y.z, z)$ ⟶ coordinate blinding

2. $s_r = s + r.|E|$ ⟶ scalar blinding

3. $M_r = [s_r](x.z, y.z, z)$ ⟶ blinded scalar mult.

4. ⟶ no unblinding

The sequence of operations (D, A) is related to the scalar bits. However:

If ***s*** is random:

- the sequence of scalar bits changes for every ECC execution

If ***P*** is randomized:

- Intermediate data is random (masked) -> hardly predicted!

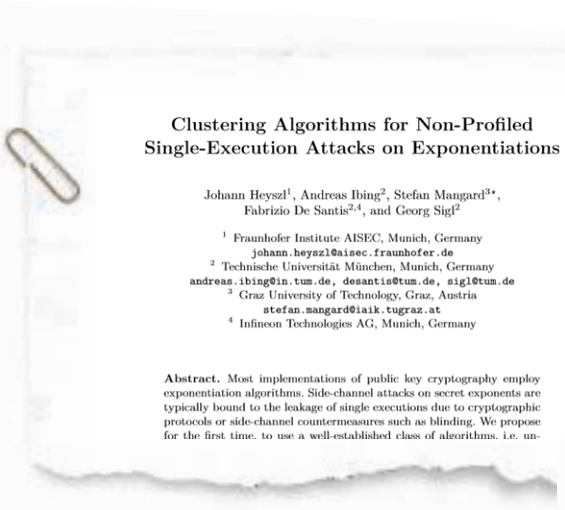
DPA is based on the prediction of intermediate data. SPA is protected too is the operations in scalar multiplication are regular and ***s*** is random.

Coordinate blinding + Scalar Randomization: WHAT NOW?

riscure

- **Horizontal correlation analysis** is not feasible
- The target might have a CPU. What if it leaks?
  - Conditional branches? Address-bit?

"**Clustering algorithms** can recover **CPU** and **address-bit** leakages"



Clustering Algorithms for Non-Profiled Single-Execution Attacks on Exponentiations

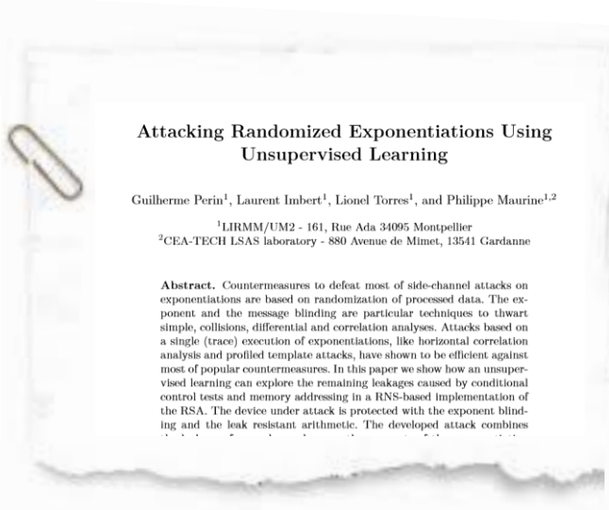
Johann Heyszl[1], Andreas Ibing[2], Stefan Mangard[3⋆], Fabrizio De Santis[2,4], and Georg Sigl[2]

[1] Fraunhofer Institute AISEC, Munich, Germany
johann.heyszl@aisec.fraunhofer.de
[2] Technische Universität München, Munich, Germany
andreas.ibing@in.tum.de, desantis@tum.de, sigl@tum.de
[3] Graz University of Technology, Graz, Austria
stefan.mangard@iaik.tugraz.at
[4] Infineon Technologies AG, Munich, Germany

Abstract. Most implementations of public key cryptography employ exponentiation algorithms. Side-channel attacks on secret exponents are typically bound to the leakage of single executions due to cryptographic protocols or side-channel countermeasures such as blinding. We propose for the first time, to use a well-established class of algorithms, i.e. un-

2013

K-means



Attacking Randomized Exponentiations Using Unsupervised Learning

Guilherme Perin[1], Laurent Imbert[1], Lionel Torres[1], and Philippe Maurine[1,2]

[1]LIRMM/UM2 - 161, Rue Ada 34095 Montpellier
[2]CEA-TECH LSAS laboratory - 880 Avenue de Mimet, 13541 Gardanne

Abstract. Countermeasures to defeat most of side-channel attacks on exponentiations are based on randomization of processed data. The exponent and the message blinding are particular techniques to thwart simple, collisions, differential and correlation analyses. Attacks based on a single (trace) execution of exponentiations, like horizontal correlation analysis and profiled template attacks, have shown to be efficient against most of popular countermeasures. In this paper we show how an unsupervised learning can explore the remaining leakages caused by conditional control tests and memory addressing in a RNS-based implementation of the RSA. The device under attack is protected with the exponent blinding and the leak resistant arithmetic. The developed attack combines

2014

K-means

Fuzzy K-means

Improving Non-Profiled Attacks on Exponentiations Based on Clustering and Extracting Leakage from Multi-Channel High-Resolution EM Measurements

Robert Specht[1], Johann Heyszl[1], Martin Kleinsteuber[2], and Georg Sigl[3]

[1] Fraunhofer Institute AISEC, Munich, Germany
robert.specht@aisec.fraunhofer.de, johann.heyszl@aisec.fraunhofer.de
[2] Technische Universität München, Munich, Germany
kleinsteuber@tum.de, sigl@tum.de

Abstract. The success probability of side-channel attacks depends on the used measurement techniques as well as the algorithmic processing to exploit available leakage. This is particularly critical in case of asymmetric cryptography, where attackers are only allowed single side-channel

2015

Expectation-Maximization + PCA

A Semi-Parametric Approach for Side-Channel Attacks on Protected RSA Implementations

Guilherme Perin and Lukasz Chmielewski

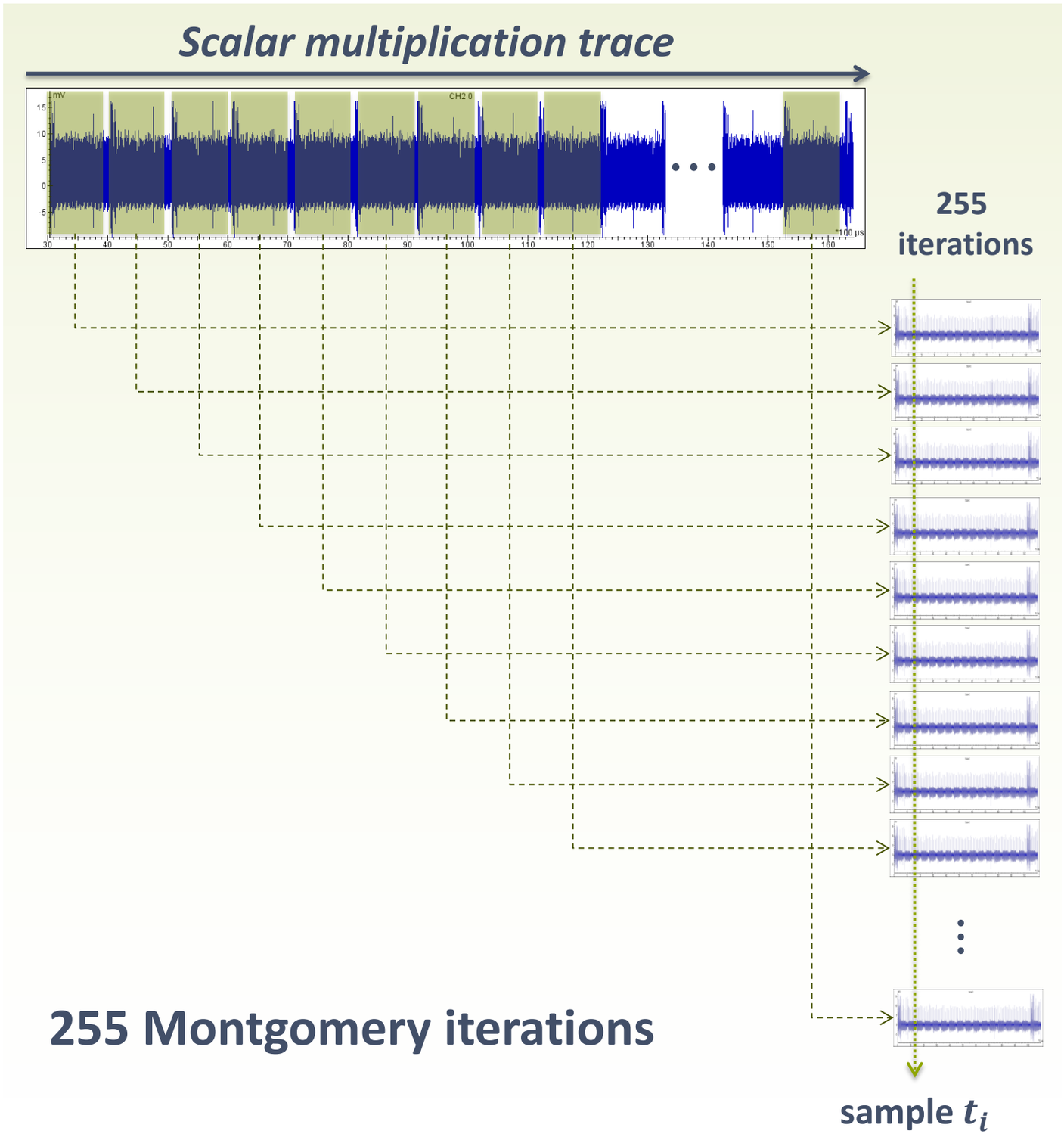
Riscure BV, Delftechpark 49, Delft, The Netherlands,
Perin@riscure.com, Chmielewski@riscure.com

Abstract. Side-channel attacks on RSA aim at recovering the secret exponent by processing multiple power or electromagnetic traces. The exponent blinding is the main countermeasure which avoids the application of classical forms of side-channel attacks, like SPA, DPA, CPA and template attacks. Horizontal attacks overcome RSA countermeasures by attacking single traces. However, the processing of a single trace is limited

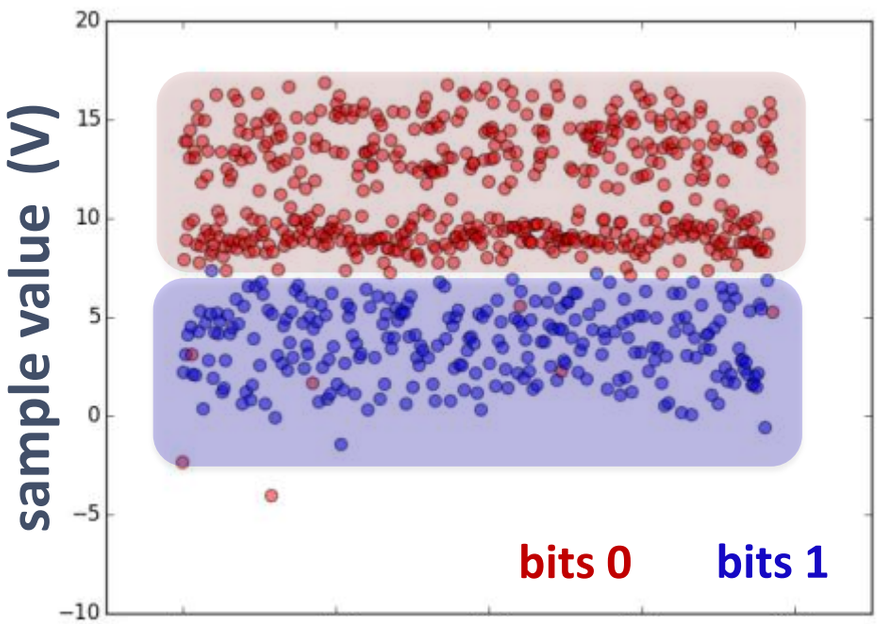
2015

K-means , F. K-Means, E-M + POIs select.

Example: clustering algorithm (k-means)



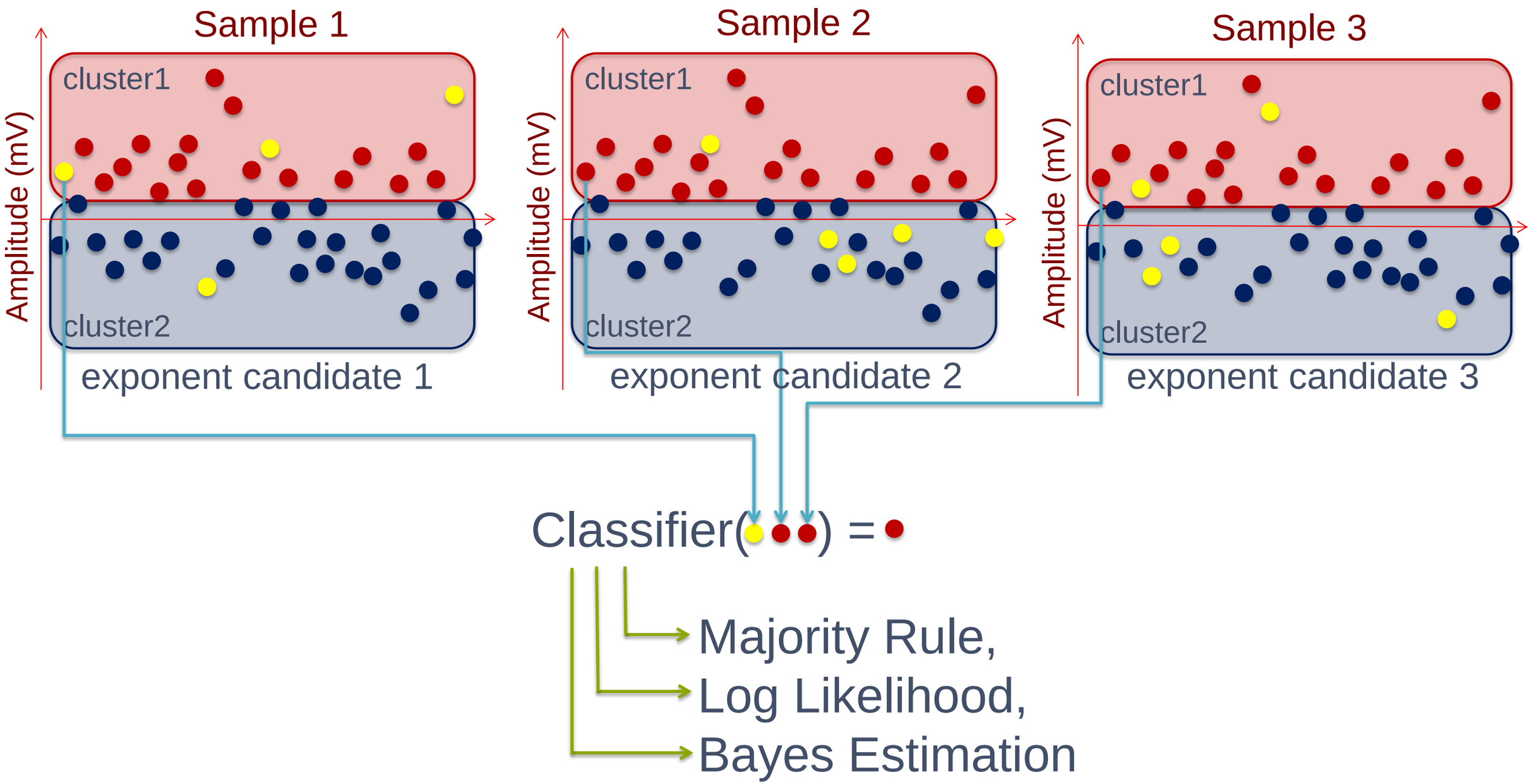
Apply **k-means** to the set of 255 samples:



confidence probability

255 Samples

Cluster Analysis



- What about multi-dimensional clustering?

Target 1

Target 1

Target 2

riscure

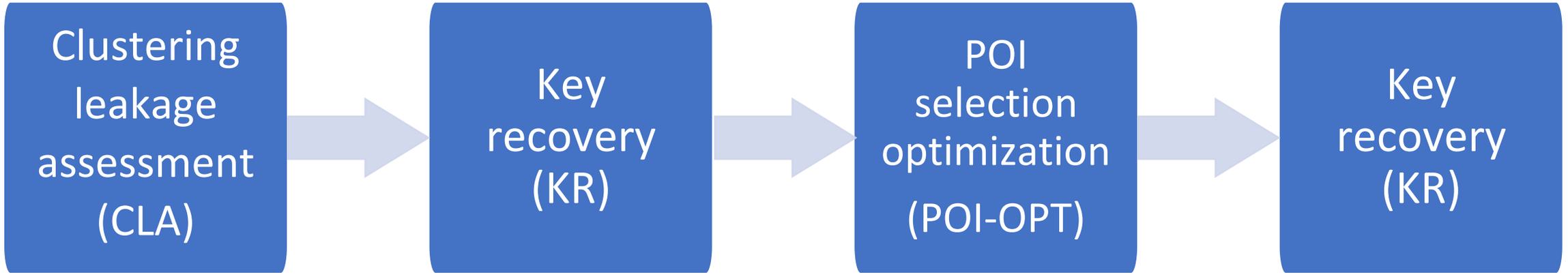
Algorithm 1 Montgomery ladder with arithmetic cswap and randomized projective coordinates.

```
// ... initialization omitted ..
bprev ← 0
for i = 254 . . . 0 do
    RE_RANDOMIZE_COORDS(work_state)
    b ← bit i of scalar
    s ← b ⊕ bprev
    bprev ← b
    CSWAP(work_state, s)
    LADDERSTEP(work_state)
end for
// ... return ommited ..
```

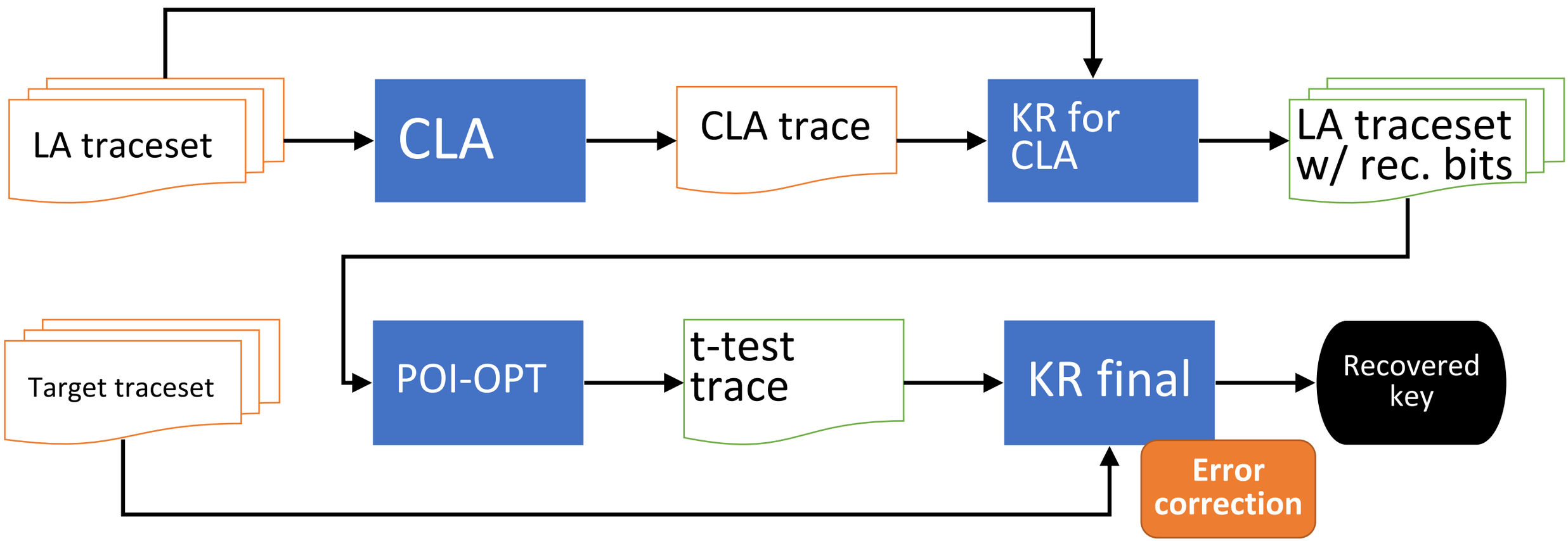
Contributions

- Single trace attack against ECDH:
  - countermeasures: regularity, projective coordinate, (re-)randomization, scalar randomization, among others.
- Applications to the Montgomery Ladder for Curve25519 from the μNaCl, based on cswap operation through:
  - arithmetic of field elements (cswap-arith),
  - swapping the pointers to such elements (cswap-pointer).
- Non-profiled attack framework (so no template attack)
- Single dimensional (POIs) and multi-dimensional clustering (attack)
- 97.64% for cswap-arith and 99.60% cswap-pointer.
  - Some brute force considerations…

HCA framework (I)

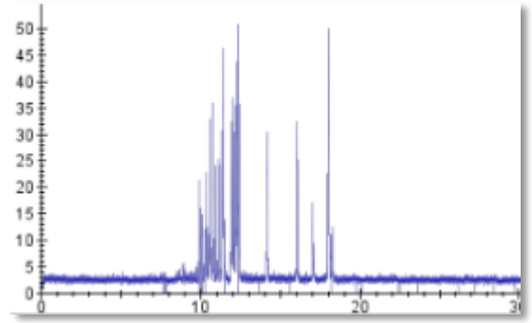


HCA framework (II)



Cluster Leakage Assessment (CLA) (I)

- Leakage assessment (LA): used to figure out if, and where, there is leakage; and how strong it is.
- The time indices with the strongest leakage are the so-called points of interest (**POIs**)



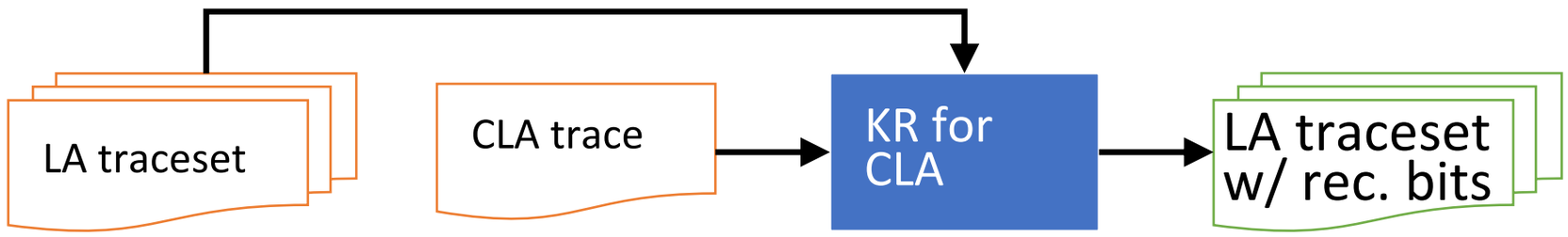
- HCA single-trace attack context is very restricted:
  - Adv. has no knowledge of (and can't change) the secret key or any ephemeral secret data used in any traces.

Cluster Leakage Assessment (CLA) (II)

- Based on **cluster leakage assessment (CLA)** [Perin'15]:
  - clustering-based non-supervised method we used to overcome such challenges.
- Works on implementations protected with (combinations of) classic countermeasures, i.e.:
  - regularity
  - scalar randomization
  - point randomization
- **Distinguishers**:

SOSD	Sum of squared differences
SOST	Sum-of-squared t-values
MIA	Mutual information analysis

Key Recovery (HCA-KR) (I)



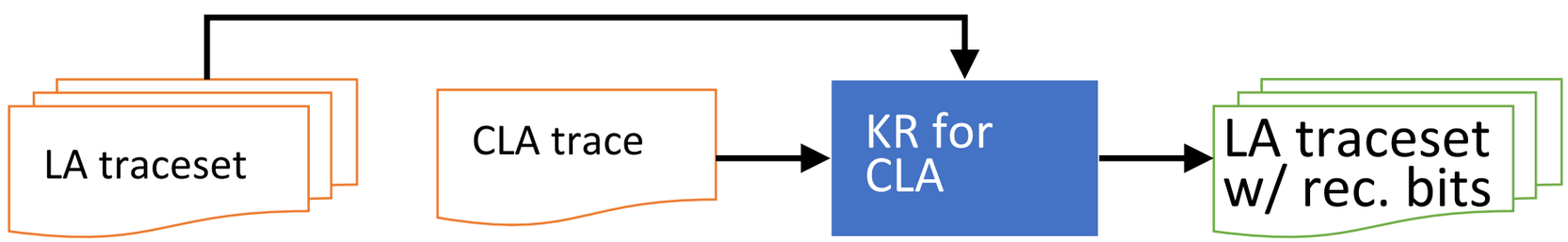
Two approaches for clustering:

- **Single-dimensional**
  - Runs on the set of samples at a given single time index (POI)
  - Output is two clusters, one for each possible scalar bit value
  - Followed by decision on the final scalar bit candidate value, based on: **Majority Rule** (MJ) or **Log-Likelihood** (LL)

- **Multi-dimensional (aka multi-attribute)**
  - Runs on all samples, at all points of a set of time indices (POIs)
    - Does not require combination step
  - Capable of exploiting higher-order leakage
  - However, it is more affected by noise

Key Recovery (HCA-KR) (II)

KR for CLA



- Run KR on the **LA traceset** using POIs in the **CLA trace**
- Output is a list of recovered key candidate bits for such segment traces

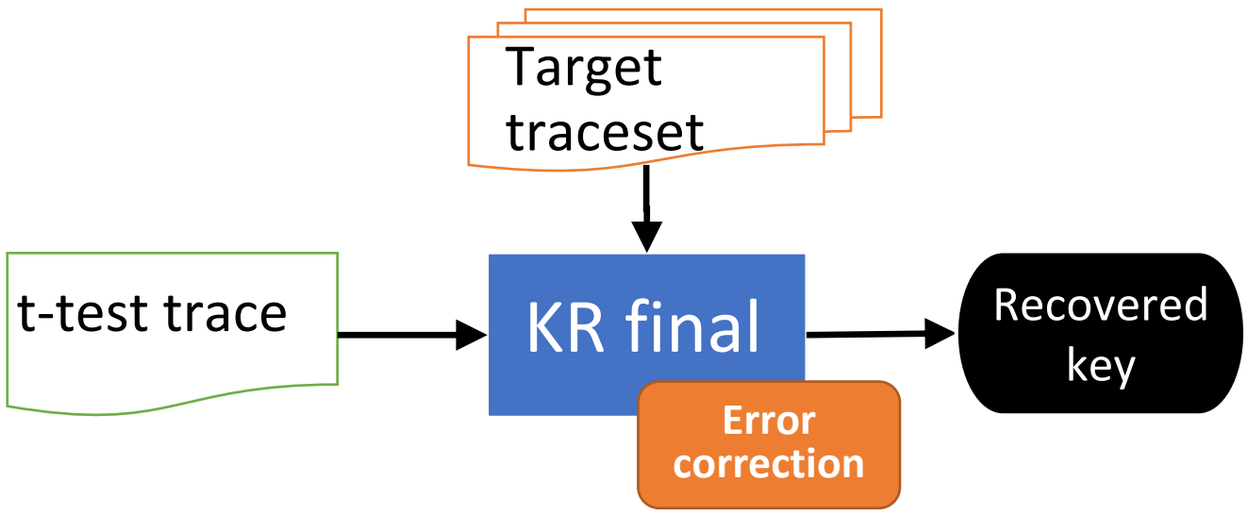
Key Recovery (HCA-KR) (II)

POI-OPT

- Refines the POIs found on CLA step
- Apply t-test with two groups:
  - **Group 1**: traces w/ candidate scalar bit equal to **0**
  - **Group 2**: traces w/ candidate scalar bit equal to **1**
- Points with largest t-statistics are the refined POIs.

Key Recovery (HCA-KR) (III)

KR final



For each trace in target traceset:

1. Run KR using POIs in t-test trace, obtain a candidate scalar
2. Run the probabilistic key error correction algorithm on it
3. If correct scalar found, return it;
4. Else, loop.

Attack Results - Initial Evaluation Experiment (I)

Acquisition

- 300 full ECSM traces for cswap-arith and cswap-pointer
- After preprocessing:
  - two tracesets of 76,500 ECSM iterations each
  - cswap-arith: each trace with 8,000 samples (8 bits per sample)
  - cswap-pointer: each trace with 1,000 samples

Maximum success rate

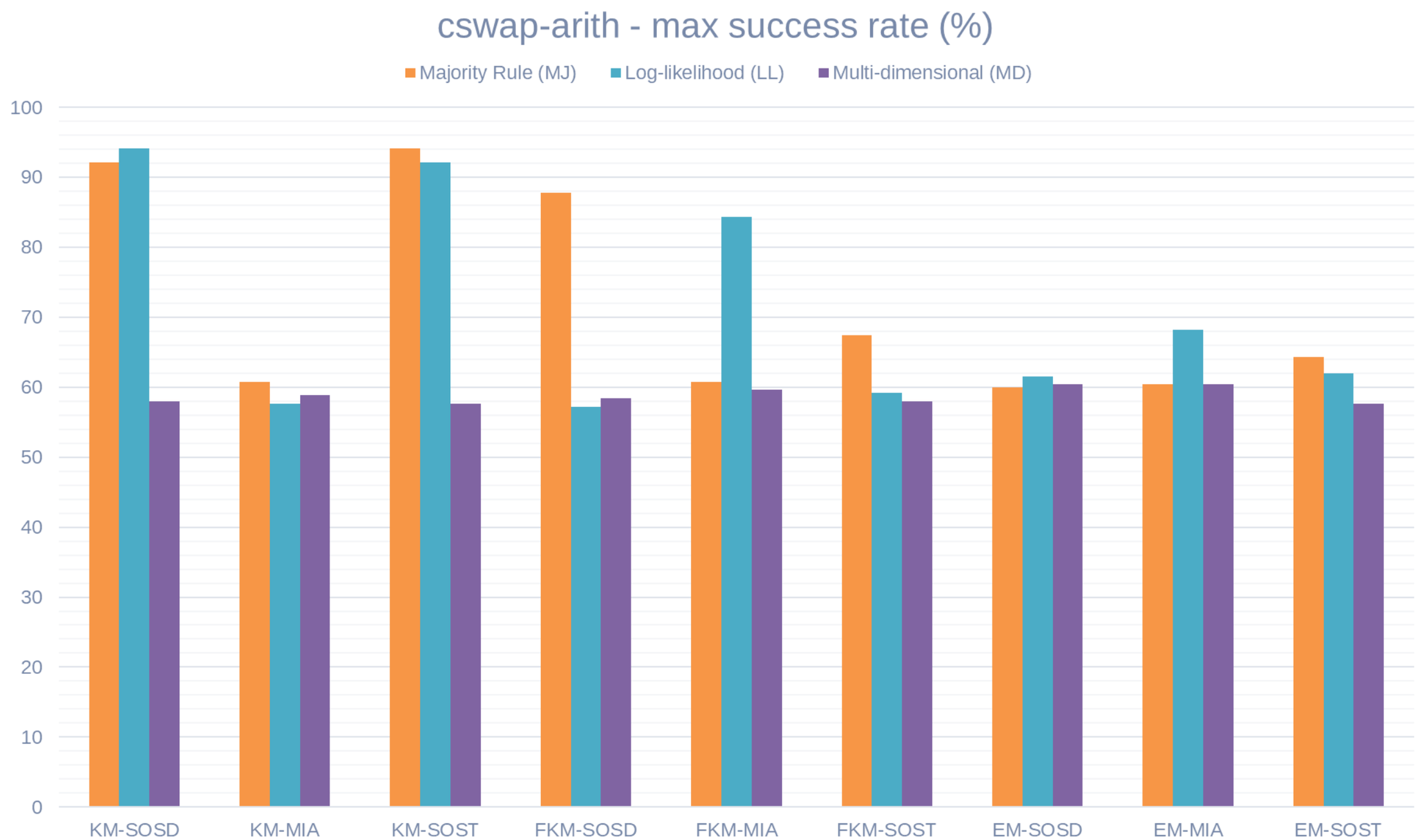
- A successful attack on a single trace of a set of target traces is sufficient.
- If $SR_1, \dots, SR_{n_t}$ are the success rates (e.g., number of correctly recovered bits) for each of n_t target traces, then we report $\max\{SR_1, \dots, SR_{n_t}\}$ as the overall attack/experiment success rate.

First experiment, parameters with fixed values

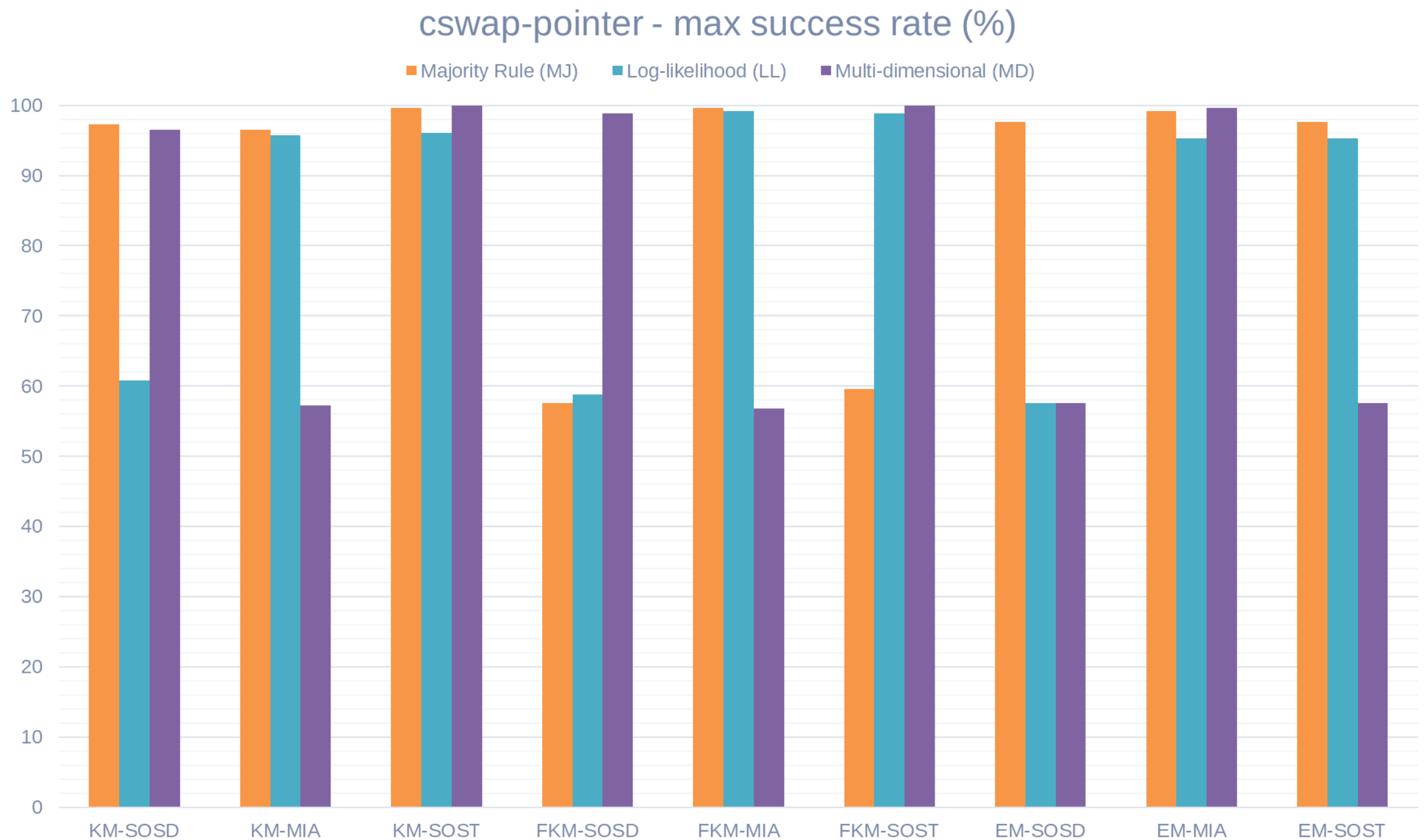
- POI-OPT enabled, Tukey test for OD, replace by median for OH, and:

	CLA	KR for CLA	KR final
# traces	100	100	100
# POIs	-	20	20

Attack Results - Initial Evaluation Experiment – cswap-arith (III)



Attack Results - Initial Evaluation Experiment – cswap-pointer (IV)



Attack Results – Comparison of KKA, CLA and POI-OPT traces

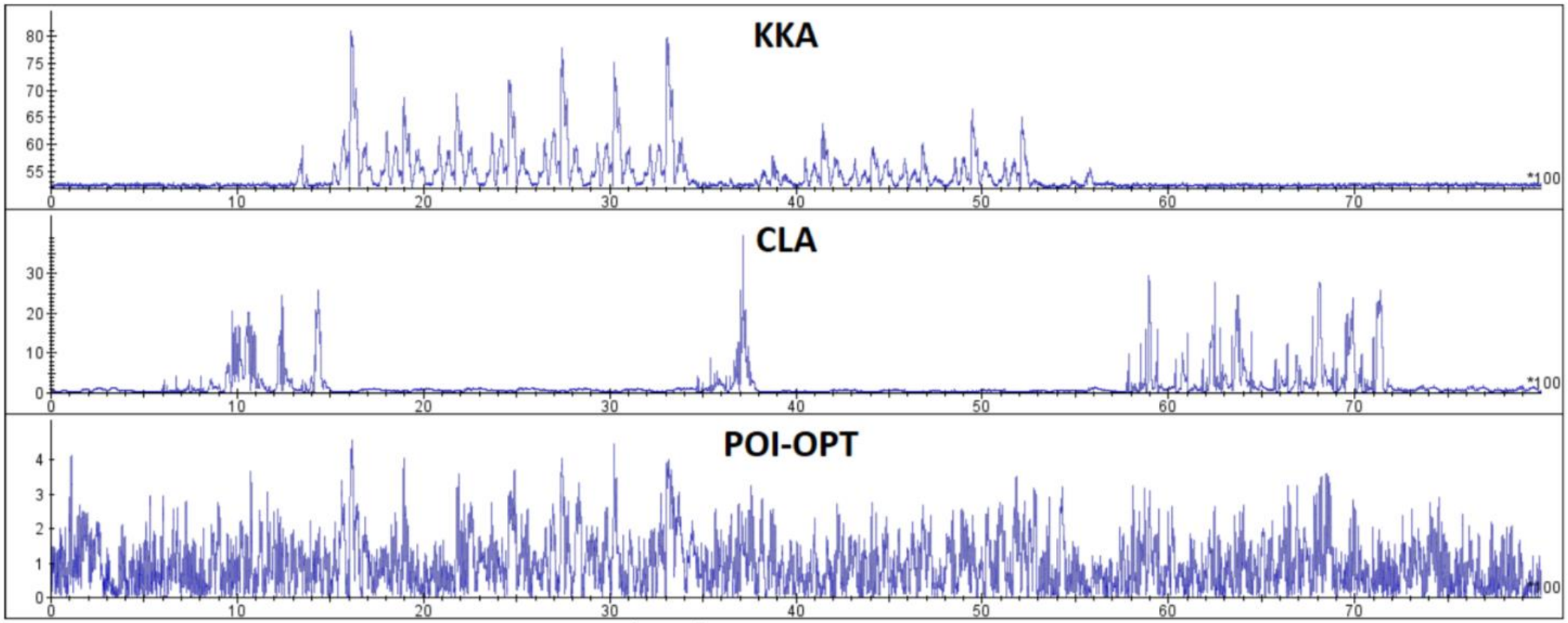


Fig. 3: Known-key analysis trace (top) and leakage assessment traces right after CLA (middle) and POI-OPT (bottom) steps, for cswap-arith.

Attack Results – Final Experiment Results

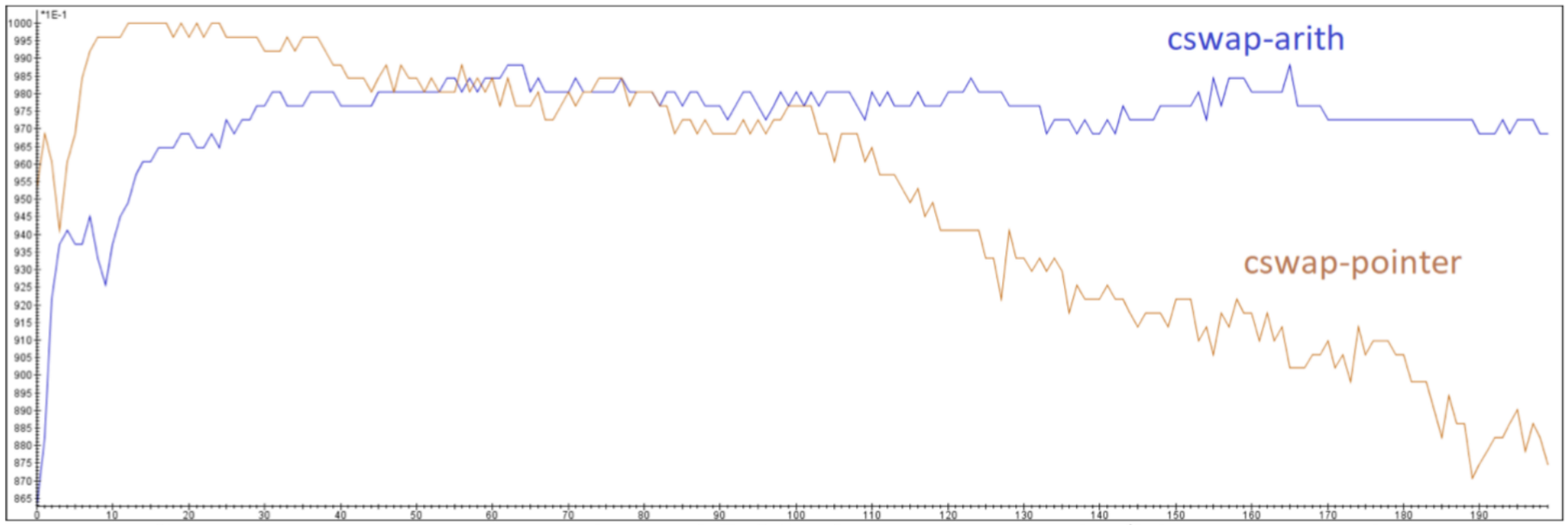


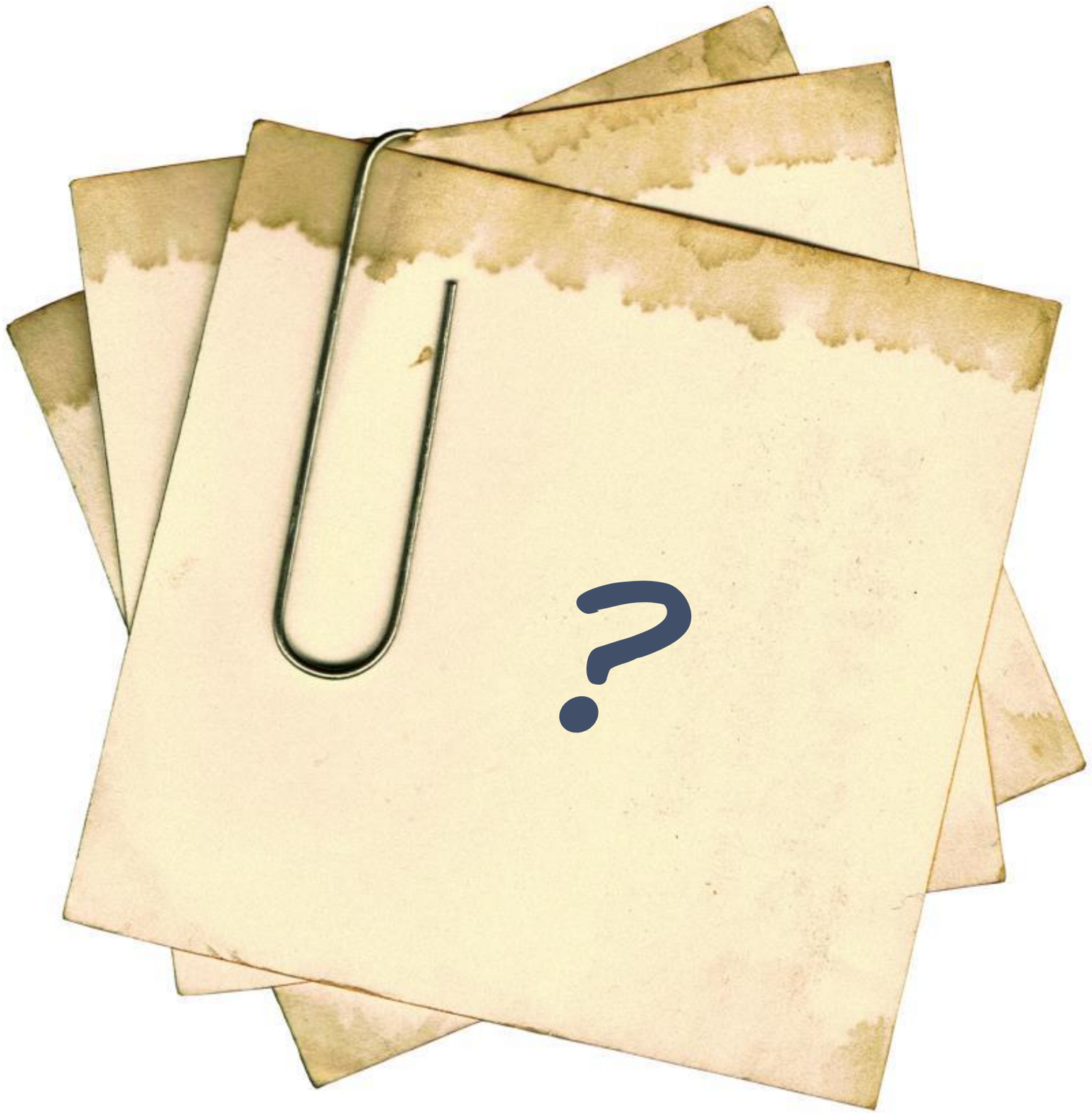
Fig. 4: Success rate versus number of POIs for cswap-arith (blue curve, almost constant) and cswap-pointer (brown curve, decreasing).

- cswap-arith: $\mathrm{SR} = \mathbf{97.64}\%$ for $nPOIs = 100$
- cswap-pointer: $\mathrm{SR} = \mathbf{99.60}\%$ for $nPOIs = 38$
  - curve shape indicates a strong sensitivity to noise
- Only 251 bits out of the 255 bits are unknown; other bits are fixed in the X25519 spec. So the number of bits with possibly incorrect values are:
  - 6 for cswap-arith
  - 2 for cswap-pointer

Conclusions & Future Work

- Practical Attack on heavily protected implementation
- CLA (POIs selection) barely works and needs improvement for more noisy channels
- More work processing of the traces (e.g. PCA)
- Evaluating against countermeasures
- Brute-force

riscure



riscure

Challenge your security

Contact: Chmielewski@riscure.com

Riscure B.V.
Frontier Building, Delftechpark 49
2628 XJ Delft
The Netherlands
Phone: +31 15 251 40 90

www.riscure.com

Riscure North America
71 Stevenson Street, Suite 400
San Francisco, CA 94105
USA
Phone: +1 650 646 99 79

inforequest@riscure.com