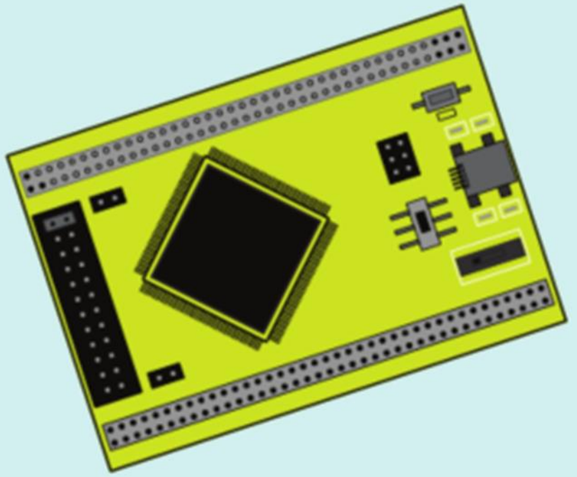
Practical Side-Channel Attacks on ECC



Workshop on Elliptic Curve Cryptography,
Autumn School on Isogenies,
Taipei, 29th of October 2024

Łukasz Chmielewski

CROCS
Centre for Research on Cryptography and Security

MUNI
FI

Outline

- Disclaimer: subjective
  - Mainly academic take but I tried to take the commercial aspect into account
- A (very) brief introduction to Side-Channel Analysis against ECC
  - Passive vs. active SCA
  - Very brief introduction to ECC (which attacks happen in practice)
  - What is the context of Side-Channel Attacks? Certifications
- Classic non-profiled attacks like timing analysis, SPA and DPA
  - SCA Countermeasures
  - For historical reasons there will be RSA mentioned
- (Selected) Attack History
  - From a classical SCA point of view.
  - Why single trace?
  - Briefly cover existing horizontal techniques
  - Mention attacks on Isogeny-based schemes.
- Recent Developments + Excercises

(VERY BRIEF) INTRO TO SIDE-CHANNEL ANALYSIS AND FAULT INJECTION

Why Is Hardware Security Important?

Card / Money Theft

Identity Theft

Premium Content Theft

Phone / Money Theft

Impersonation

(Relatively) Recent Practical Attacks on ECC

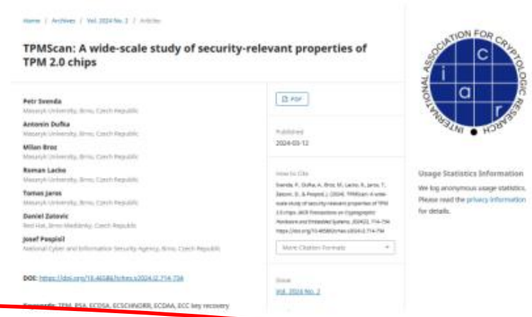
TPM-FAIL, November 13, 2019

LadderLeak, May 28, 2020

SCA Titan: January 7, 2021

Minerva, October 3, 2019

TPMScan, March 12, 2024



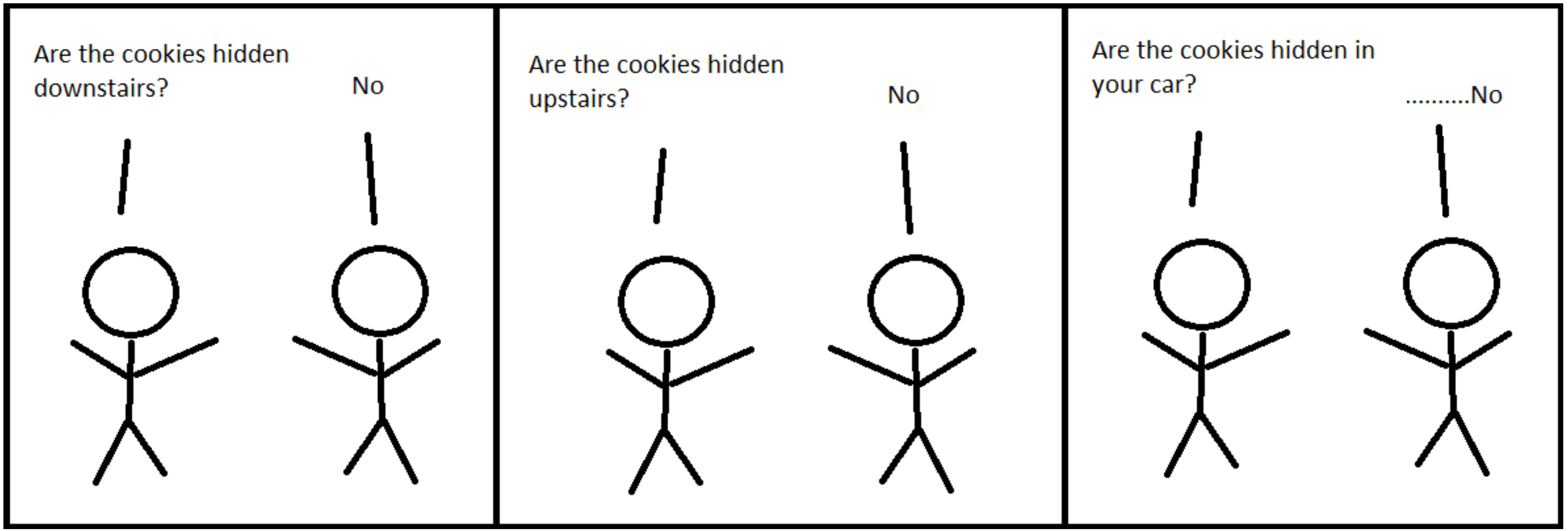
EUCLEAK, Recently

Introduction:

Side Channel Analysis

https://www.independent.co.uk/news/world/europe/melting-snow-being-used-by-police-to-find-cannabis-farms-in-the-netherlands-10036057.html

Cookies Example



https://www.simplethread.com/great-scott-timing-attack-demo/

Side Channels

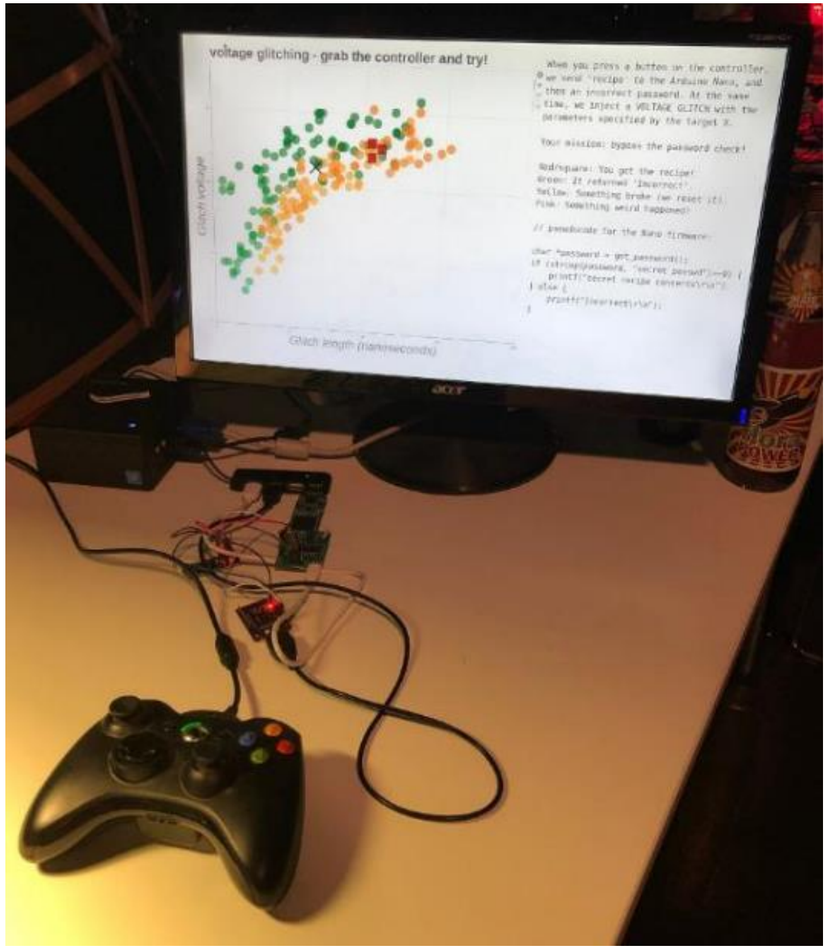
- Time 
- Power 
- Electro Magnetic Emanations 
- Light 
- Sound 
- Temperature 
- …

Passive vs. Active Side Channels

Passive: analyze device behavior

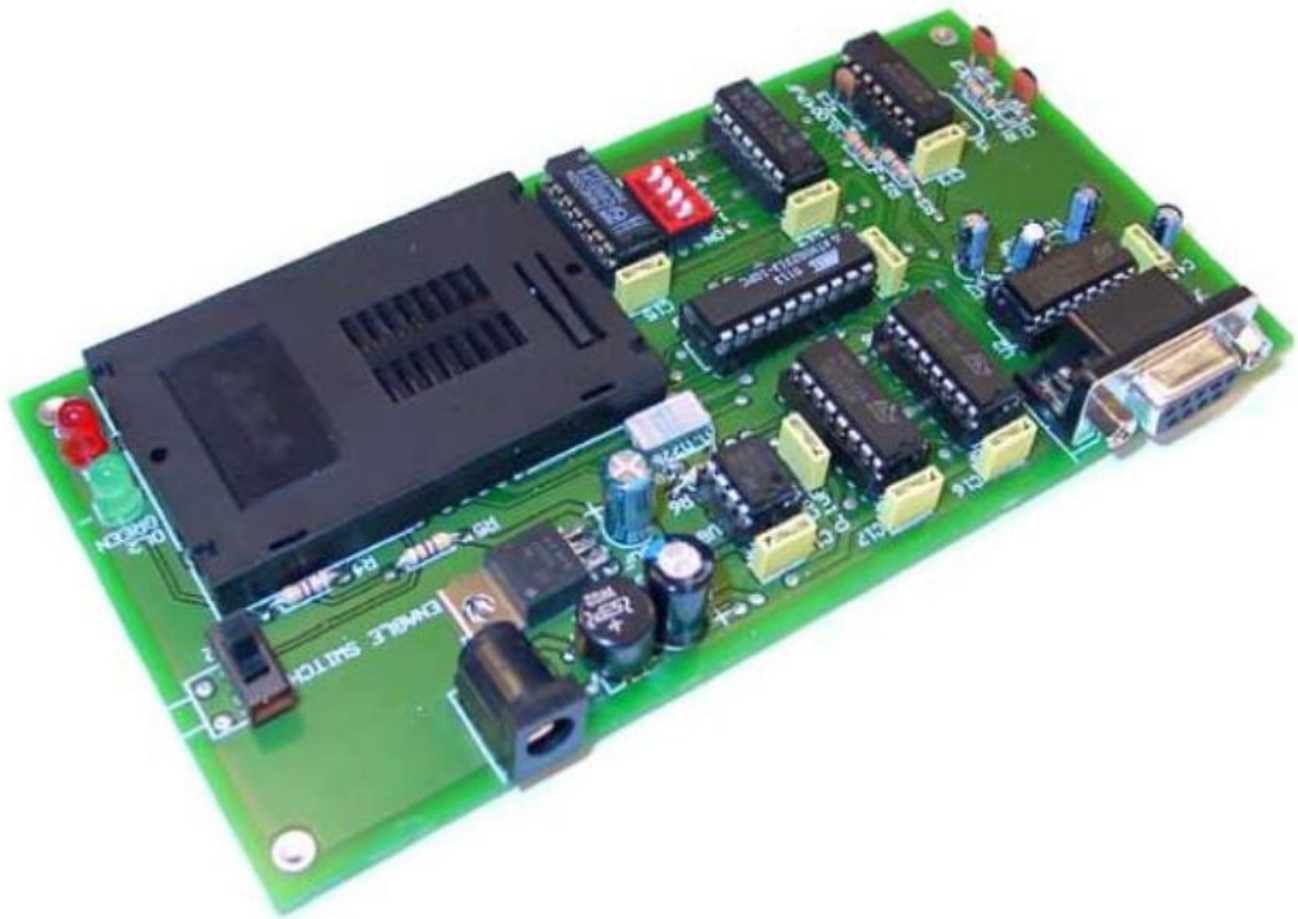
Active: change device behavior

Some Practical Side-Channel Setups



“Commercial” Example: the “unlooper” device

```
1  void entry() {
2      void* start  = 0x80000000;
3      void* length = 0x00400000;
4
5      serial_puts("Start Secure Boot...\n");
6
7      loadOSFromHardDrive(start);
8
9      if (! authenticateOS(start,length) )
10         do {} while(1);
11
12     serial_puts("Run OS\n");
13
14     boot_next_stage(start);
15     //starts executing at the address start
16 }
```

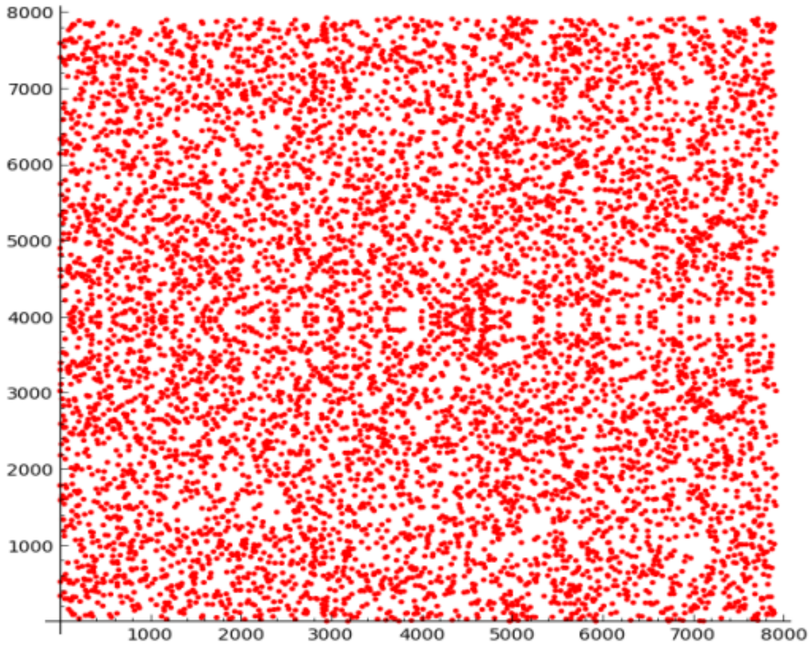


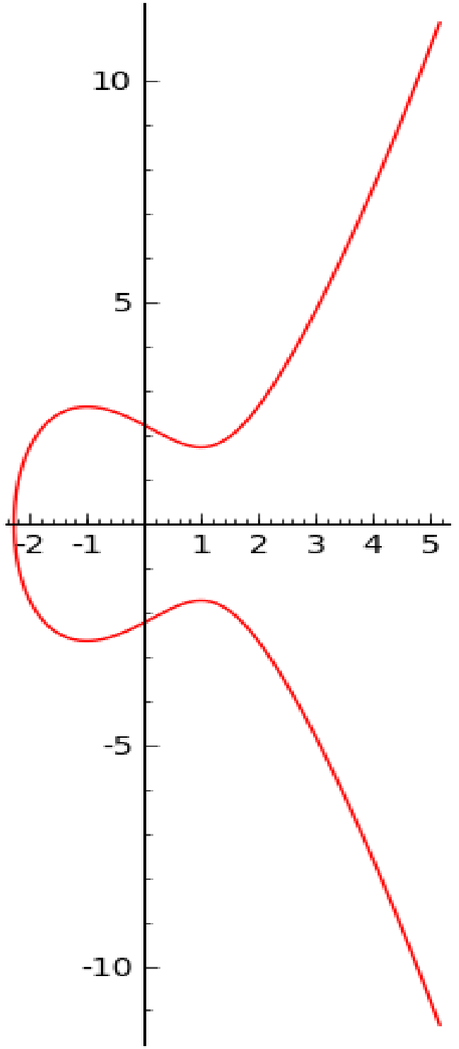
(VERY BRIEF) INTRO TO ECC & ATTACK CONSIDERATIONS

Elliptic curves

- Example
  - $y^2 = x^3 - 3x^2 + 5$ plus some double and add operartions
- How would it look: *mod* 7919?

(x, y)





Scalar multiplication by *k*

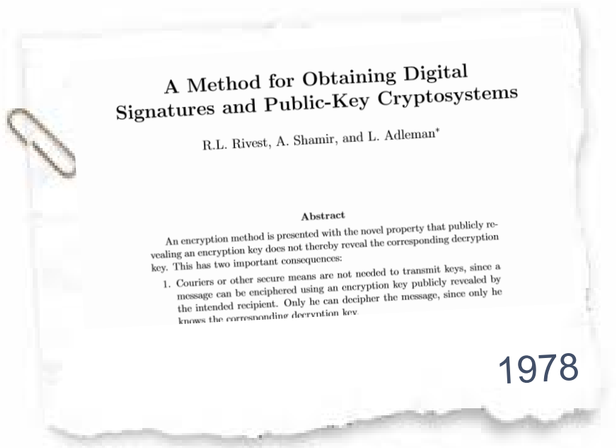
- Group = points on the curve + special point ∞
- Add a point to itself k times: Q = [k]P
- Analogous to modular exponentiation
- square-and-multiply = double-and-add
- Many (equivalent) algorithms
- LTR, RTL, Window, Comb, Ladders, …

```
x=∞
for j=|k|-1 to 0 {
    x=DBL(x)
  if kj=1
    x=ADD(x,P)
return x0
```

```
(0xb1011) = 11
x=∞
j=3
  x=DBL(∞)
  x=ADD(∞,P)
j=2
  x=DBL(P)
j=1
  x=DBL(2P)
  x=ADD(4P,P)
j=0
  x=DBL(5P)
  x=ADD(10P,P)=11P
```

RSA ≈ ECC

- RSA is based on modular exponentiation
- exponentiation ≈ scalar multiplication
  - $m = c^d \bmod n \approx Q = [k]P$
- multiplication ≈ points addition
- squaring ≈ point doubling



A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

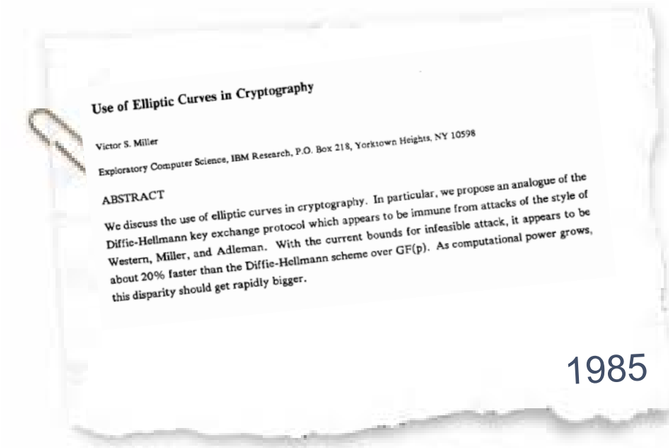
R.L. Rivest, A. Shamir, and L. Adleman*

Abstract

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences:

1. Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.

1978



Use of Elliptic Curves in Cryptography

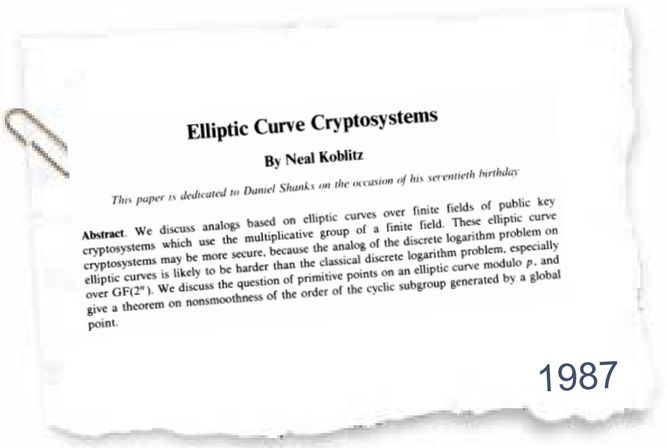
Victor S. Miller

Exploratory Computer Science, IBM Research, P.O. Box 218, Yorktown Heights, NY 10598

ABSTRACT

We discuss the use of elliptic curves in cryptography. In particular, we propose an analogue of the Diffie-Hellmann key exchange protocol which appears to be immune from attacks of the style of Western, Miller, and Adleman. With the current bounds for infeasible attack, it appears to be about 20% faster than the Diffie-Hellmann scheme over GF(p). As computational power grows, this disparity should get rapidly bigger.

1985



Elliptic Curve Cryptosystems

By Neal Koblitz

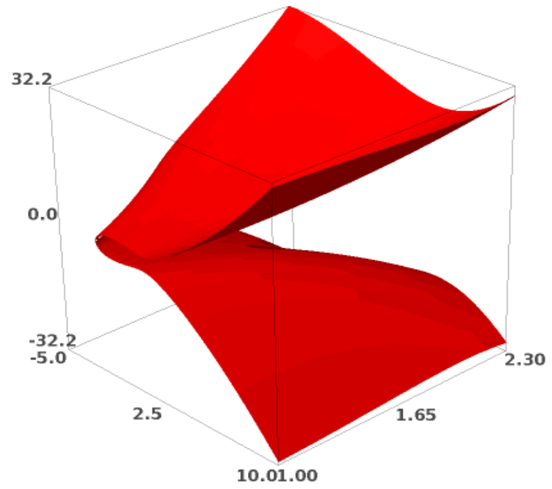
This paper is dedicated to Daniel Shanks on the occasion of his seventieth birthday

Abstract. We discuss analogs based on elliptic curves over finite fields of public key cryptosystems which use the multiplicative group of a finite field. These elliptic curve cryptosystems may be more secure, because the analog of the discrete logarithm problem on elliptic curves is likely to be harder than the classical discrete logarithm problem, especially over GF(2^n). We discuss the question of primitive points on an elliptic curve modulo p, and give a theorem on nonsmoothness of the order of the cyclic subgroup generated by a global point.

1987

Implementation Considerations

- Some curves are more common than others:
  - NIST curves (p256), Curve25519, or secp256k1
- Many security properties
  - Curves classification wrt. security: http://safecurves.cr.yp.to/
- Are there powerful fault attacks against ECC?
  - Invalid point attacks
  - Attacks against deterministic ECDSA
- The classical (x,y) representation is rarely used
  - For transport points compression is often used (note that x defines y)
  - For efficiency reasons it is better to represent a point in more dimensions (e.g., (x,y,z))



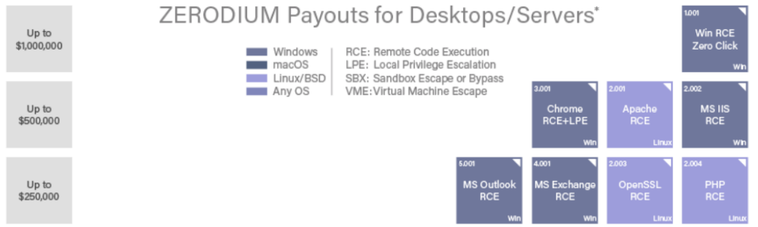
What are the applications of ECC?

- Cryptocurrencies 
- Not used for payment
- (Certified) Secure Element
  - Common Criteria
- Isogenies Based Cryptography ☺

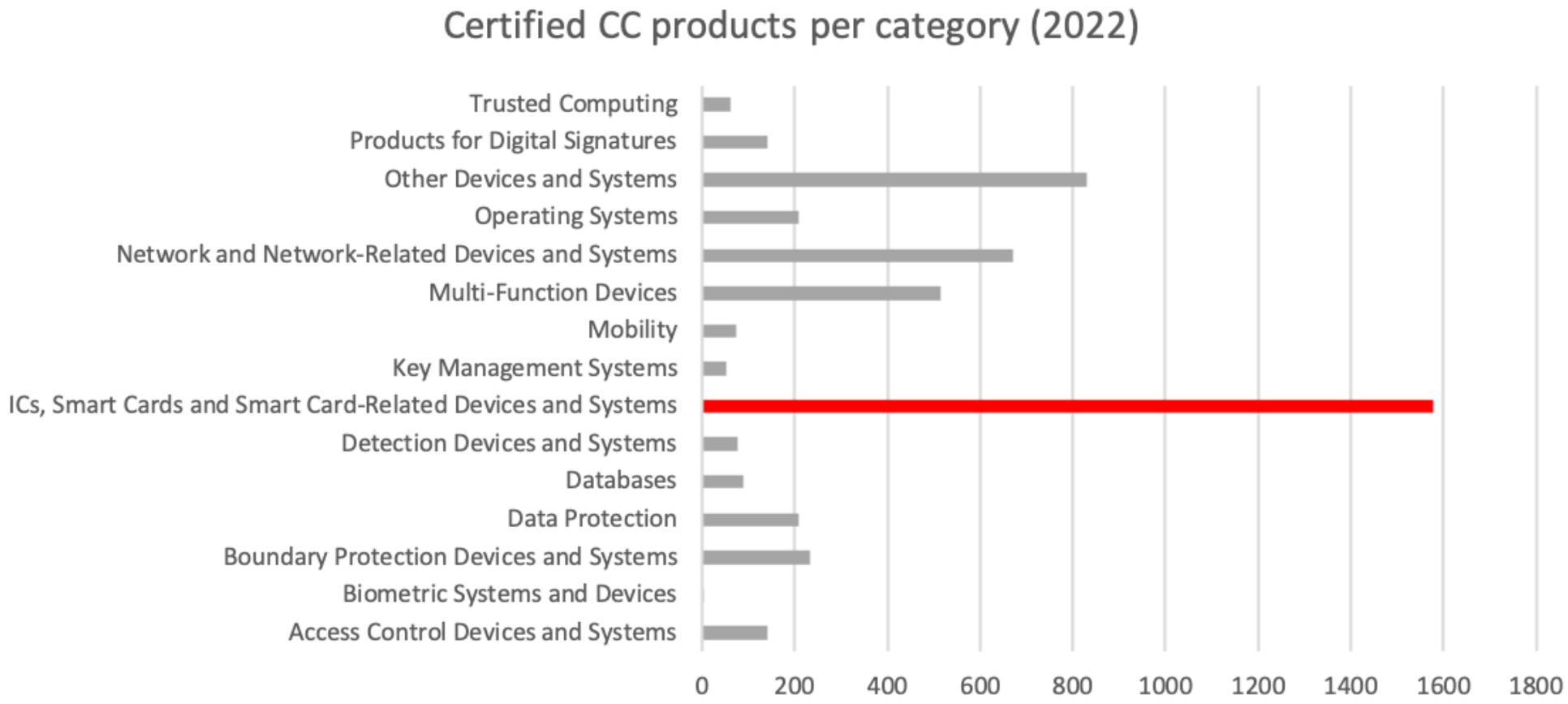
- Signatures: ECDSA and deterministic variants
- Key Exchange: ECDH

What is the context of performing a successful side-channel attack?

- Can the attack be patched?
  - How? Products Replaced?
- How does typical work look?
  - As an academic researcher / manufacturer: develop and attack & test
  - As an analyst: evaluations and Common Criteria
  - As an independent researcher (or small company): work for customers to improve their security; end goal – often CC
- You found the attack on an ECC product - what then?
  - Sell the bug as zero-day? Something like ZERODIUM
  - Responsible disclosure? What About CC?
  - Bug Bounties? What About CC?
  - How often does it happen? Big impact, see https://crocs.fi.muni.cz/public/papers/rsa_ccs17



Common Criteria (Products)



SELECTED ATTACKS HISTORY

Timing (Warm-up): What is Wrong?

```
Input: 4-digit PIN code
Output: PIN verified or rejected
Process CheckPIN (pin[4])
int pin_ok=0;
if (pin[0]==5)
   if (pin[1]==9)
      if (pin[2]==0)
         if (pin[3]==2)
            pin_ok=1;
         end
      end
   end
end
return pin_ok;
EndProcess
```

Timing 1: What is Wrong?

Algorithm Left-to-right double-and-add algorithm

1: **Input:** P, $k = (k_{n-1}, k_{n-2}, \ldots, k_0)_2$
2: **Output:** $Q = [k]P$
3: $R_0 \leftarrow O$
4: **for** $i \leftarrow n-1$ down to 0 **do**
5: $\quad R_0 \leftarrow 2R_0$
6: $\quad$ **if** $k_i = 1$ **then**
7: $\quad\quad R_0 \leftarrow R_0 + P$
8: **return** R_0

Timing 2: What is Wrong?

Algorithm Right-to-left double-and-add algorithm

1: **Input:** P, $k = (k_{n-1}, k_{n-2}, \ldots, k_0)_2$
2: **Output:** $Q = [k]P$
3: $R_0 \leftarrow P$
4: $Q \leftarrow O$
5: **for** $i \leftarrow 0$ **to** $n - 1$ **do**
6: **if** $k_i = 1$ **then**
7: $Q \leftarrow Q + R_0$
8: $R_0 \leftarrow 2R_0$
9: **return** Q

Timing 3: What is Wrong?

Algorithm Left-to-right double-and-always-add algorithm

1: **Input:** P, $k = (k_{n-1}, k_{n-2}, \ldots, k_0)_2$
2: **Output:** $Q = [k]P$
3: $R_0 \leftarrow O$, $R_1 \leftarrow O$
4: **for** $i \leftarrow n-1$ **down to** 0 **do**
5: $\quad R_0 \leftarrow 2R_0$
6: $\quad R_{1-k_i} \leftarrow R_{1-k_i} + P$
7: **return** R_0

Timing 4: What is Wrong?

Algorithm Left-to-right double-and-add-always algorithm [Coron99]

1: **Input:** P, $k = (k_{n-1}, k_{n-2}, \ldots, k_0)_2$
2: **Output:** $Q = [k]P$
3: $R_0 \leftarrow P$
4: **for** $i \leftarrow n-2$ **down to** 0 **do**
5: $\quad R_0 \leftarrow 2R_0$
6: $\quad R_1 \leftarrow R_0 + P$
7: $\quad R_0 \leftarrow R_{k_i}$
8: **return** R_0

Timing 5: What is Wrong?

Algorithm The Montgomery Ladder

1: **Input:** $\boldsymbol{P}$, $k = (k_{n-1}, k_{n-2}, \ldots, k_0)_2$
2: **Output:** $\boldsymbol{Q} = [k]\boldsymbol{P}$
3: $\boldsymbol{R}_0 \leftarrow \mathcal{O}$
4: $\boldsymbol{R}_1 \leftarrow \boldsymbol{P}$
5: **for** $i \leftarrow n-1$ **down to** 0 **do**
6: $\quad b \leftarrow 1 - k_i$
7: $\quad \boldsymbol{R}_b \leftarrow \boldsymbol{R}_0 + \boldsymbol{R}_1$
8: $\quad \boldsymbol{R}_{k_i} \leftarrow 2 \cdot \boldsymbol{R}_{k_i}$
9: **return** $\boldsymbol{R}_0$

Timing 6: What is Wrong?

Algorithm The Montgomery ladder for x-coordinate-based X25519 scalar multiplication on $E : y^2 = x^3 + 486662x^2 + x$

Input: $k \in \{0, \ldots, 2^{255} - 1\}$ and the x-coordinate x_P of a point P
Output: $x_{[k]P}$, the x-coordinate of $[k]P$
$X_1 \leftarrow 1, Z_1 \leftarrow 0, X_2 \leftarrow x_P, Z_2 \leftarrow 1$
$p \leftarrow 0$
for $i \leftarrow 254$ **down to** 0 **do**
    $c \leftarrow k_i \oplus p, p \leftarrow k_i$
    $(X_1, X_2) \leftarrow \text{cswap}(X_1, X_2, c), (Z_1, Z_2) \leftarrow \text{cswap}(Z_1, Z_2, c)$
    $(X_1, Z_1, X_2, Z_2) \leftarrow \text{ladderstep}(x_P, X_1, Z_1, X_2, Z_2)$
$(X_1, X_2) \leftarrow \text{cswap}(X_1, X_2, p), (Z_1, Z_2) \leftarrow \text{cswap}(Z_1, Z_2, p)$
return (X_1, Z_1)

Timing 6: solution cswap

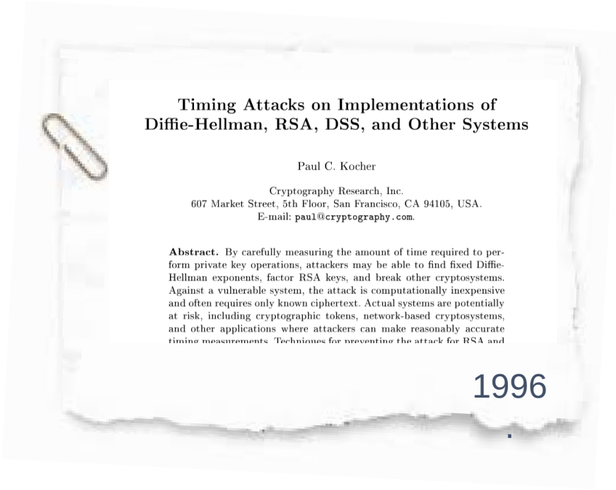
```
void fe25519_cswap(fe25519* in1, fe25519* in2, int condition) {
    int32_t mask = condition;
    mask = -mask;

    for (uint32_t ctr = 0; ctr < 8; ctr++) {
        uint32_t val1 = in1->as_uint32_t[ctr];
        uint32_t val2 = in2->as_uint32_t[ctr];
        uint32_t temp = val1;

        val1 ^= mask & (val2 ^ val1);
        val2 ^= mask & (val2 ^ temp);

        in1->as_uint32_t[ctr] = val1;
        in2->as_uint32_t[ctr] = val2;
    }
}
```

Simple Power Analysis (SPA) on ECC (scalarmult)

```
ScalarMult(P) {
  A = ∞
  for ( i = n-1; i ≥ 0; i--)
        A = DOUBLE(A)
        if (k_i = =1)
              A = ADD(A,P)
        end if
  end for
  Return A = [k]P
}
```

D

A



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

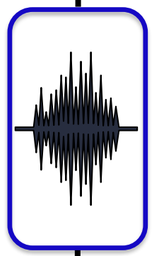
Paul C. Kocher

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

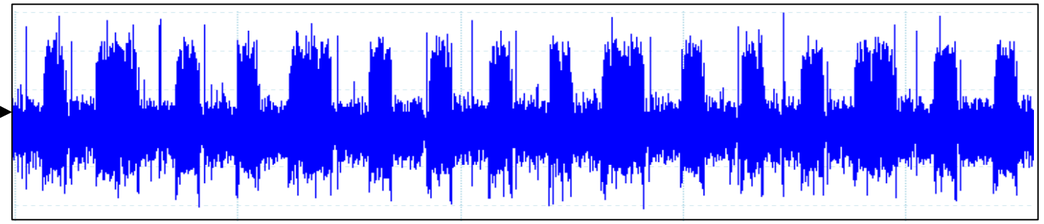
There is an exercise about that



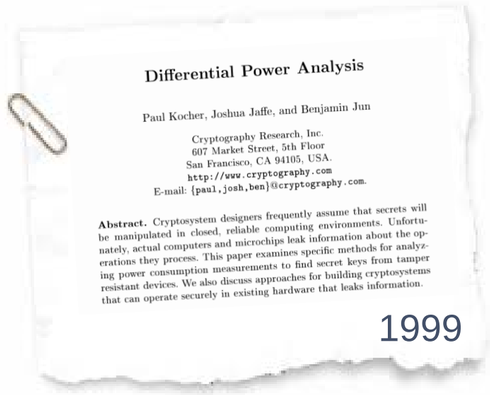
Probe

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 D D ...



Differential (Correlation) Power Analysis Power Analysis on PKC



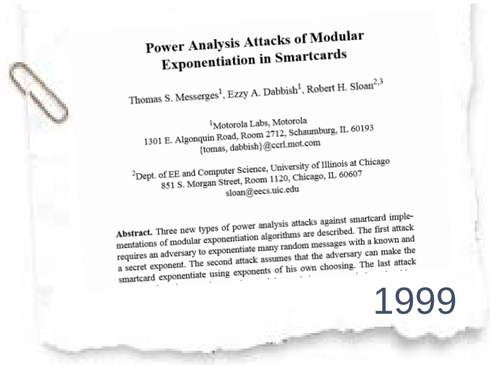
Differential Power Analysis

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,joah,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

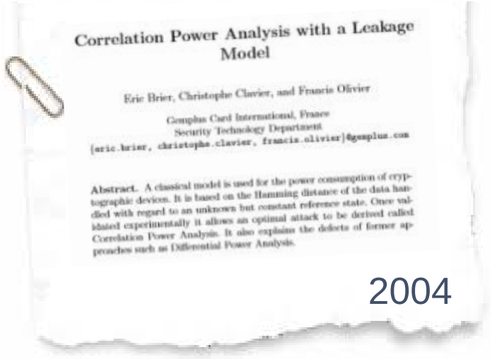
Thomas S. Messerges[1], Ezzy A. Dabbish[1], Robert H. Sloan[2,3]

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



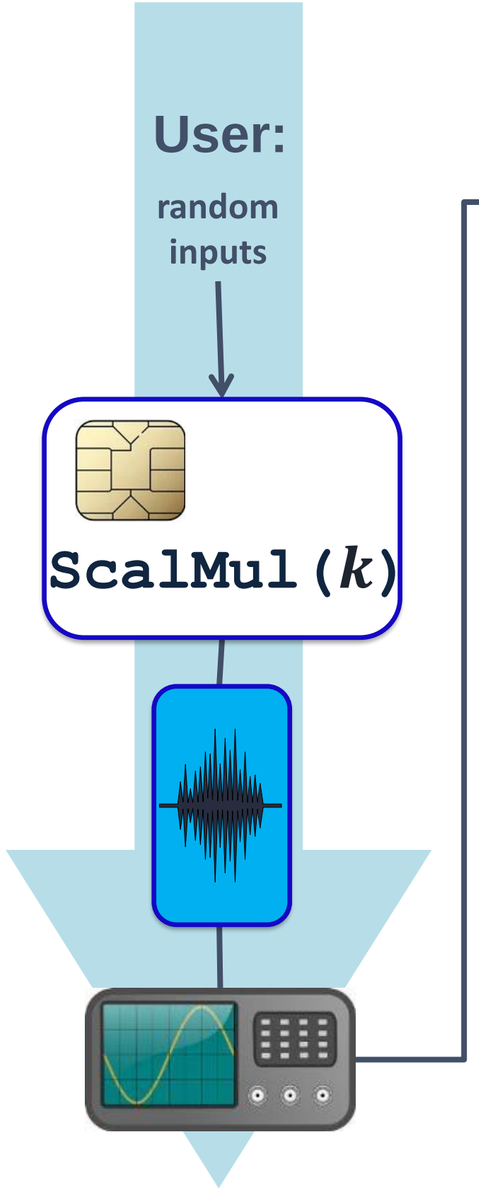
Correlation Power Analysis with a Leakage Model

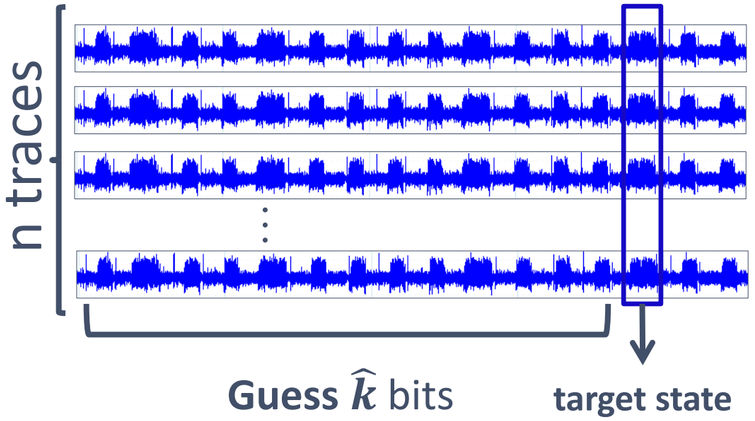
Eric Brier, Christophe Clavier, and Francis Olivier

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, $\hat{k}$, target state)

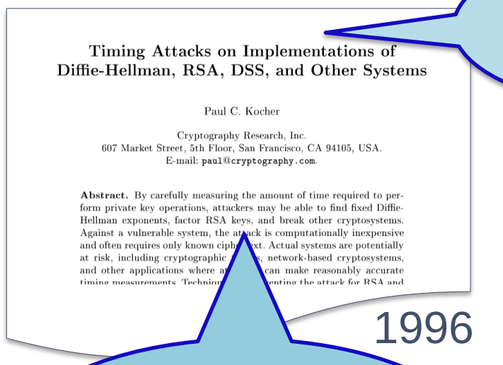
- HW of a register
- HD between current and previous register state
- ID model (value of a register)

$$f_i = \begin{cases} 0 & if\ HW \leq 16 \\ 1 & if\ HW > 16 \end{cases} \qquad f_i = HW(reg\_state)$$

DPA = Difference of Means

CPA = Pearson correlation

PKC Countermeasures


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

Paul C. Kocher

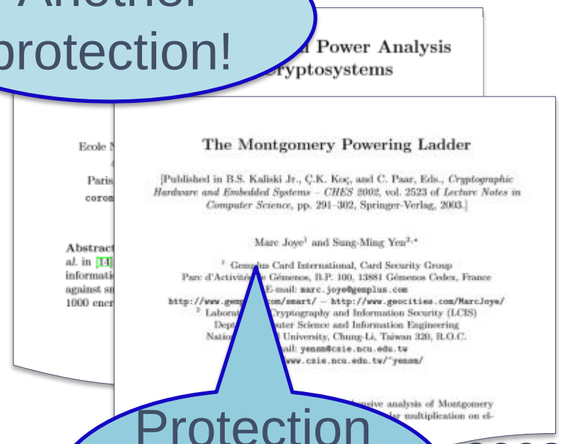
Cryptography Research, Inc.
607 Market Street, 5th Floor, San Francisco, CA 94105, USA.
E-mail: paul@cryptography.com

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 extending the attack for RSA and

1996

Attack + the protection:

Another protection!


Power Analysis
Cryptosystems

The Montgomery Powering Ladder

[Published in B.S. Kaliski Jr., Ç.K. Koç, and C. Paar, Eds., *Cryptographic Hardware and Embedded Systems – CHES 2002*, vol. 2523 of *Lecture Notes in Computer Science*, pp. 291–302, Springer-Verlag, 2003.]

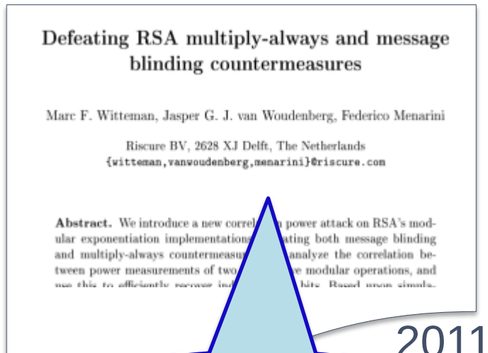
Marc Joye[1] and Sung-Ming Yen[2,*]

[1] Gemplus Card International, Card Security Group
Parc d'Activités de Gémenos, B.P. 100, 13881 Gémenos Cedex, France
E-mail: marc.joye@gemplus.com
http://www.gemplus.com/smart/ – http://www.geocities.com/MarcJoye/
[2] Laboratory for Cryptography and Information Security (LCIS)
Department of Computer Science and Information Engineering
National Central University, Chung-Li, Taiwan 320, R.O.C.
E-mail: yensm@csie.ncu.edu.tw
http://www.csie.ncu.edu.tw/~yensm/

2002

Protection against SPA:

Cross-Correlation


Defeating RSA multiply-always and message blinding countermeasures

Marc F. Witteman, Jasper G. J. van Woudenberg, Federico Menarini

Riscure BV, 2628 XJ Delft, The Netherlands
{witteman,vanwoudenberg,menarini}@riscure.com

Abstract. We introduce a new correl... power attack on RSA's modular exponentiation implementation... both message blinding and multiply-always countermeasu... analyze the correlation between power measurements of two... modular operations, and

2011

Attack against these protections

Randomize the message

+

Make the sequence of **squares** and **multiplications** unrelated to the exponent bits (**SMSMSMSMSM**…)

+

Randomize the exponent (private key)

Why do message and exponent blinding protect RSA against SCA?

$$c = m^d mod\ N$$

1. $m_r = m.r^{-e} mod\ N$ → message blinding
2. $d_r = d + r.\varphi(n)$ ⟶ exponent blinding
3. $c_r = {m_r}^{d_r}\ mod\ n$ ⟶ blinded exponentiation
4. $c = c_r.r\ mod\ n$ ⟶ message "unblinding"

The sequence of operations (S, M) is related to the exponent bits.

However:

- If ***d*** is random: the sequence of exponent bits changes for every RSA execution
- If ***m*** is random: Intermediate data is random (masked) -> hardly predicted!

DPA is based on the prediction of intermediate data.

Thesis: *Any side-channel attack requiring **multiple traces** are repelled by message **and** exponent blinding countermeasures.*

Why do coordinate and scalar blinding protect ECC against SCA?

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

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.

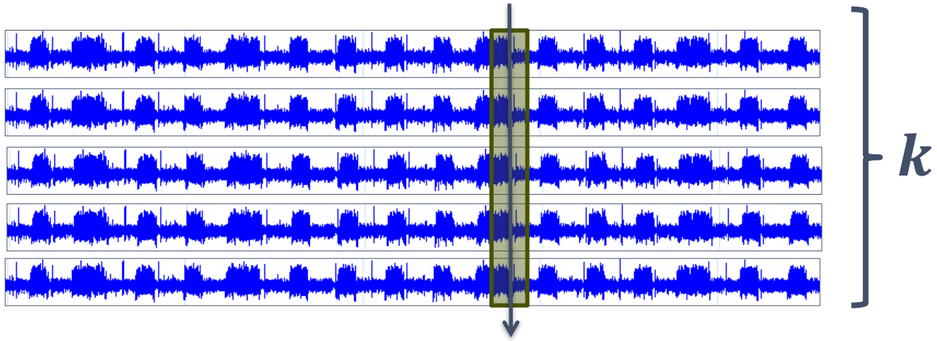
4. ⟶ no unblinding

The same situation as for RSA. Point blinding is also possible but not presented above.

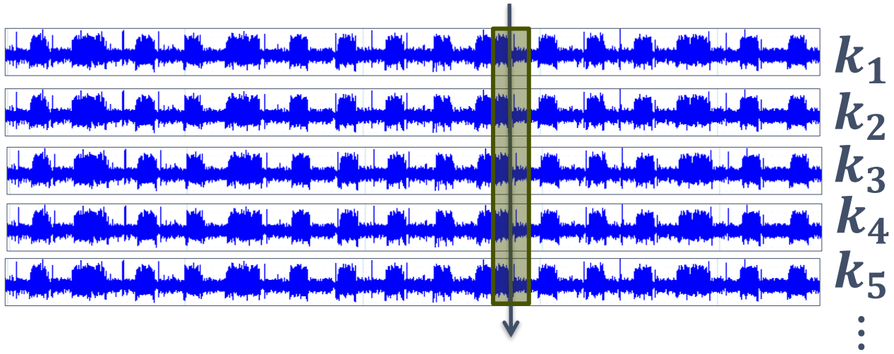
Note: there are of course differences in some detailed countermeasures.

Why horizontal?

Classical or ***vertical*** side-channel attacks (DPA, CPA) assume that the **secret is fixed** for every measured power trace.



Horizontal side-channel attacks assume that the **secret is different** for every measured power trace.

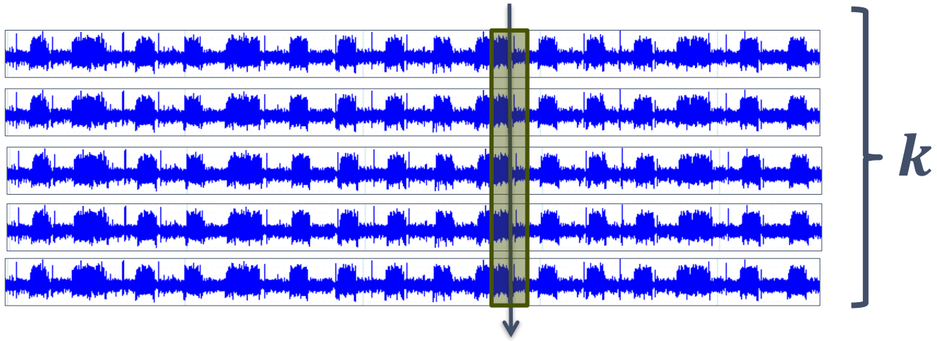


$$k_1 \neq k_2 \neq k_3 \neq k_4 \neq k_5 \neq \cdots$$

Target state has a random and hardly predictable data!

Why horizontal?

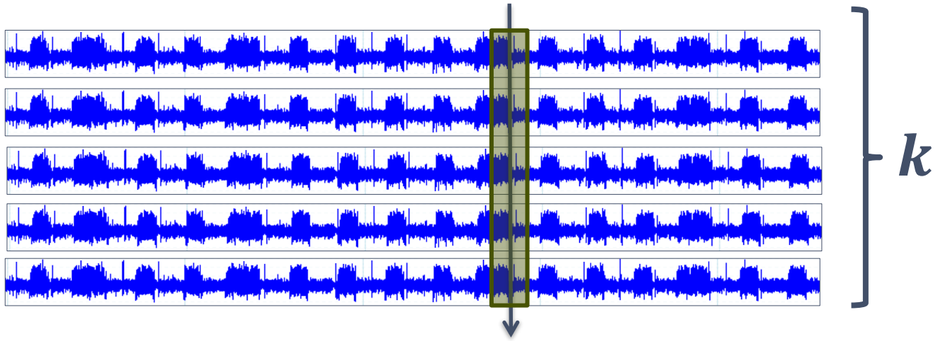
Classical or ***vertical*** side-channel attacks (DPA, CPA) assume that the **secret is fixed** for every measured power trace.



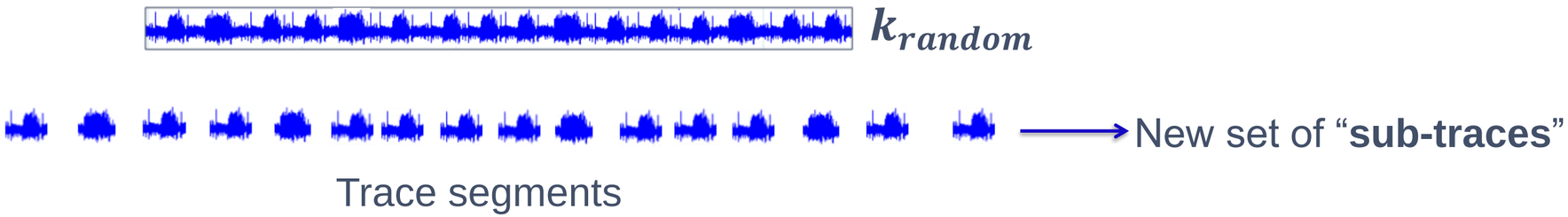
Horizontal side-channel attacks assume that the **secret is different** for every measured power trace.

Why horizontal?

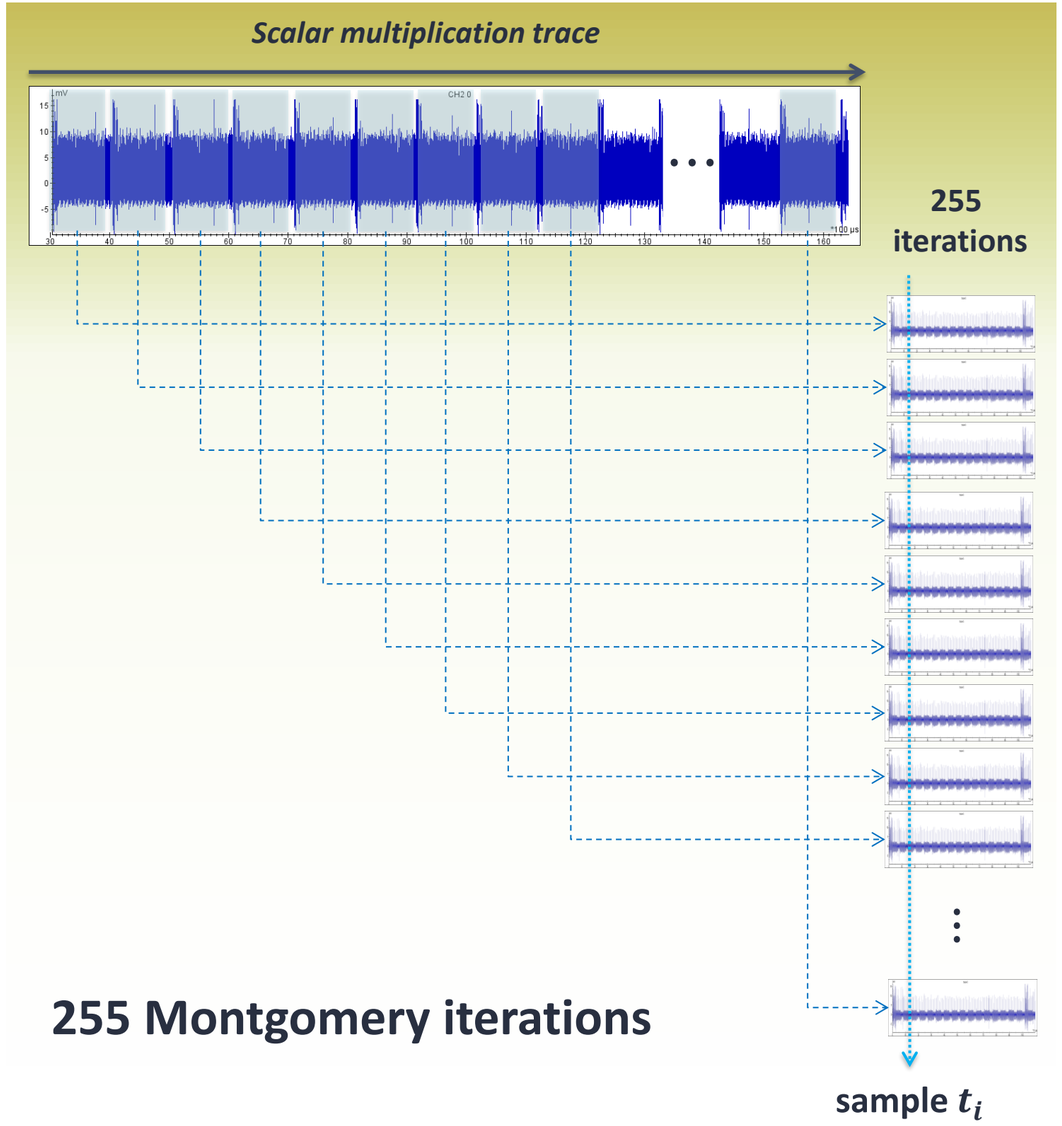
Classical or ***vertical*** side-channel attacks (DPA, CPA) assume that the **secret is fixed** for every measured power trace.



Horizontal side-channel attacks assume that the **secret is different** for every measured power trace.

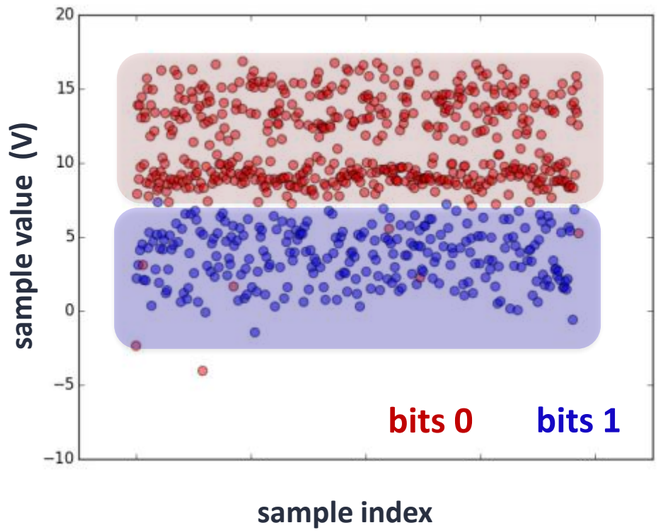


Horizontal Attacks on PKC, example: clustering



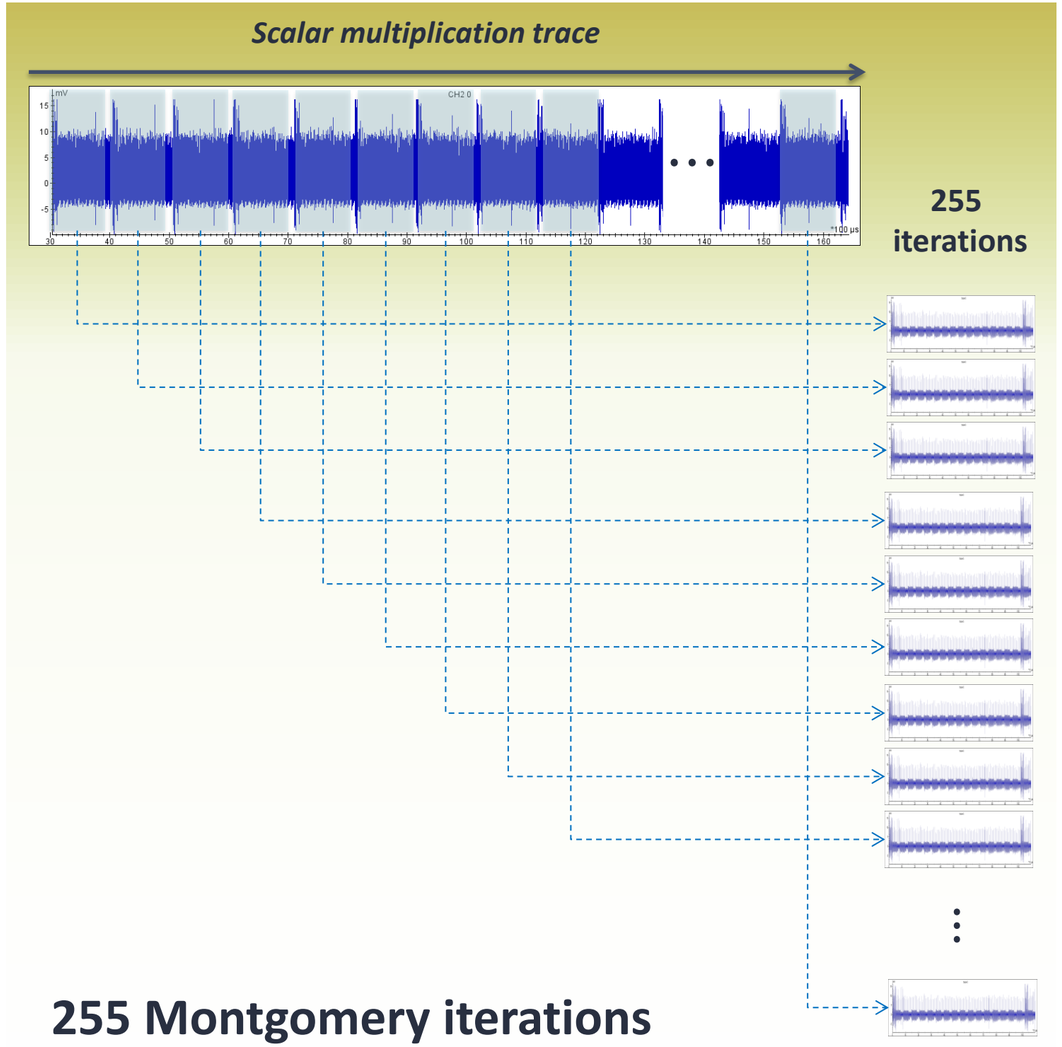
255 Montgomery iterations

Apply clustering (e.g. **k-means**) to the set of 255 samples:



255 Samples

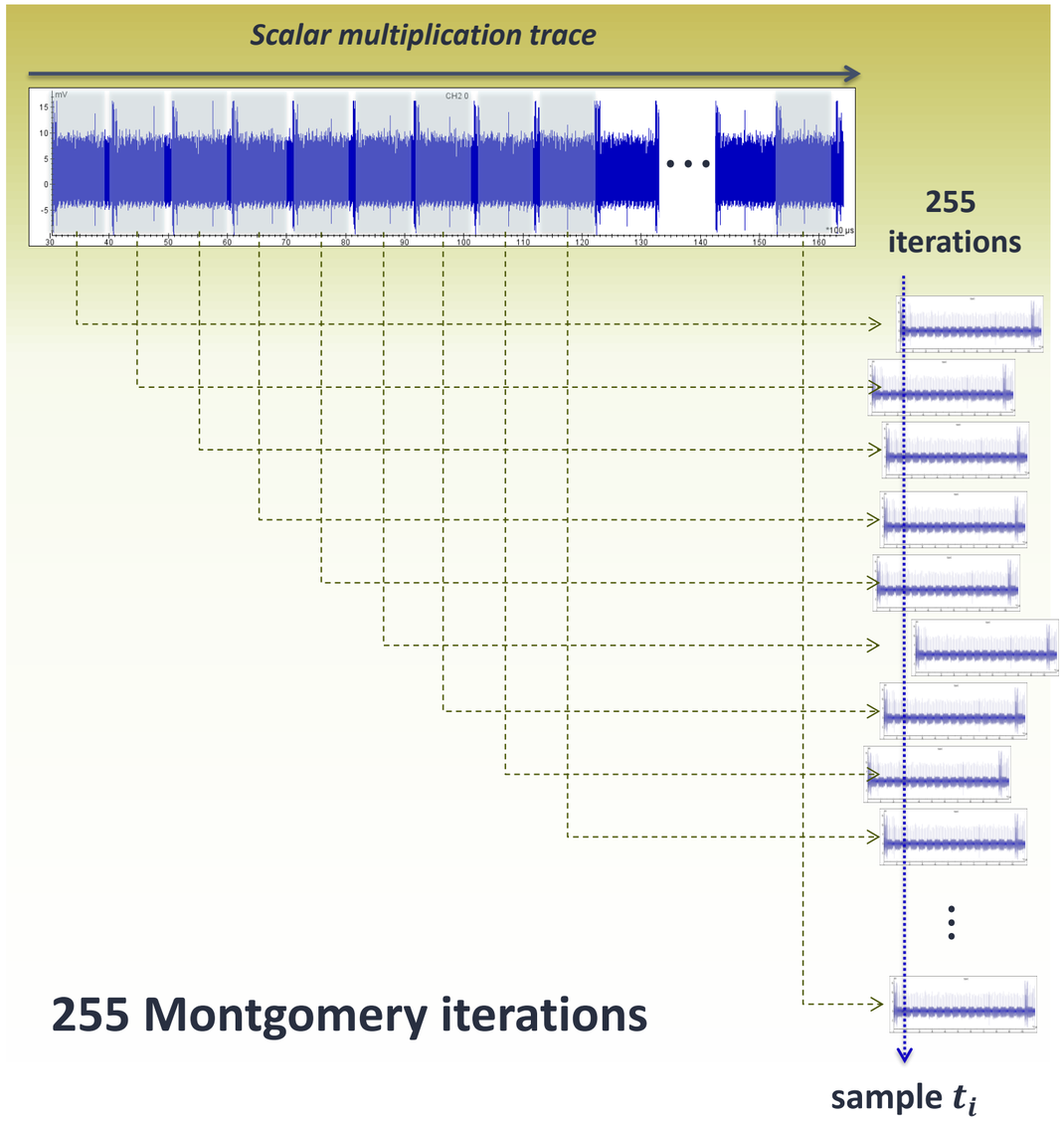
Horizontal Attack on PKC, profiled approaches



Template Attacks

Deep Learning

Problems

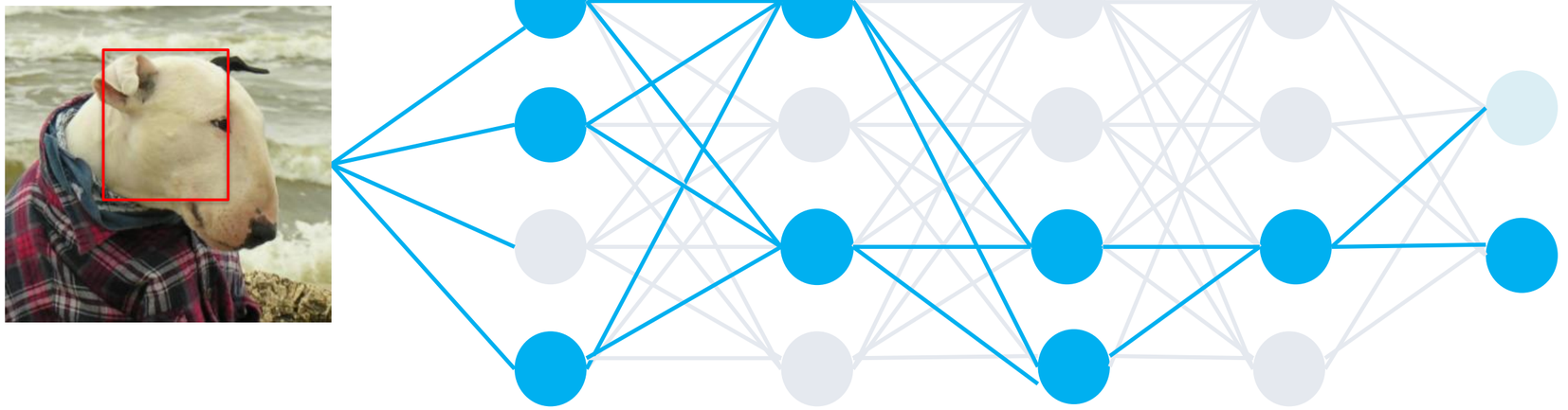


- Noise makes it hard to align
- How to find point of interest?
- In unsupervised setting!!!

Deep Learning

Black
Bullterrier = 98%

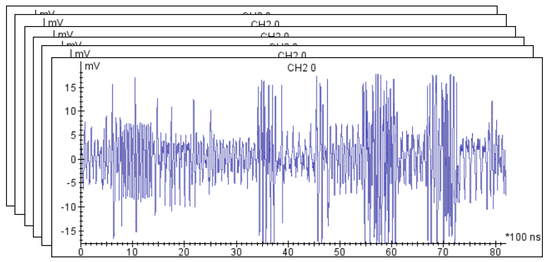
White
Bullterrier = 2%



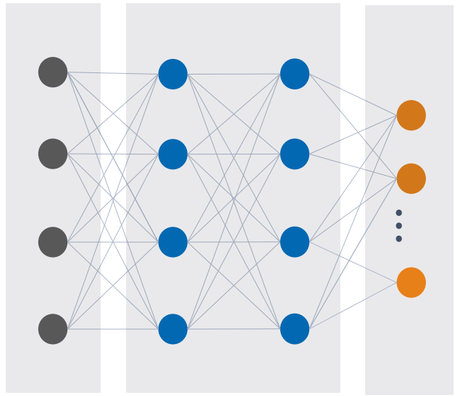
Black
Bullterrier = 5%

White
Bullterrier = 95%

Deep Learning in SCA



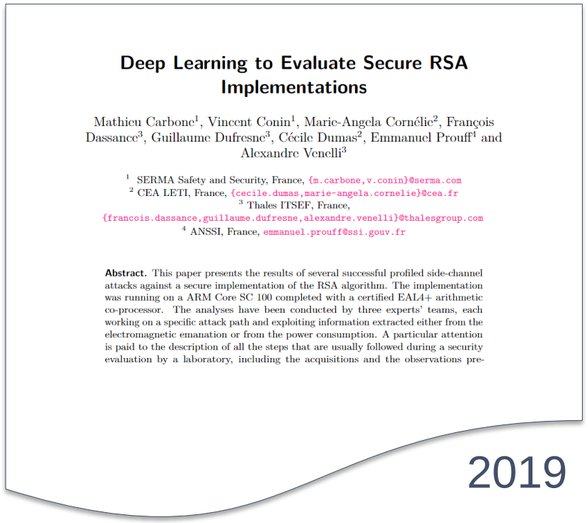
Back-propagation



Class 0

Class 1

Desired Output → Error ← Actual Output



Deep Learning to Evaluate Secure RSA Implementations

Mathieu Carbone[1], Vincent Conin[1], Marie-Angela Cornélie[2], François Dassance[3], Guillaume Dufresne[3], Cécile Dumas[2], Emmanuel Prouff[4] and Alexandre Venelli[3]

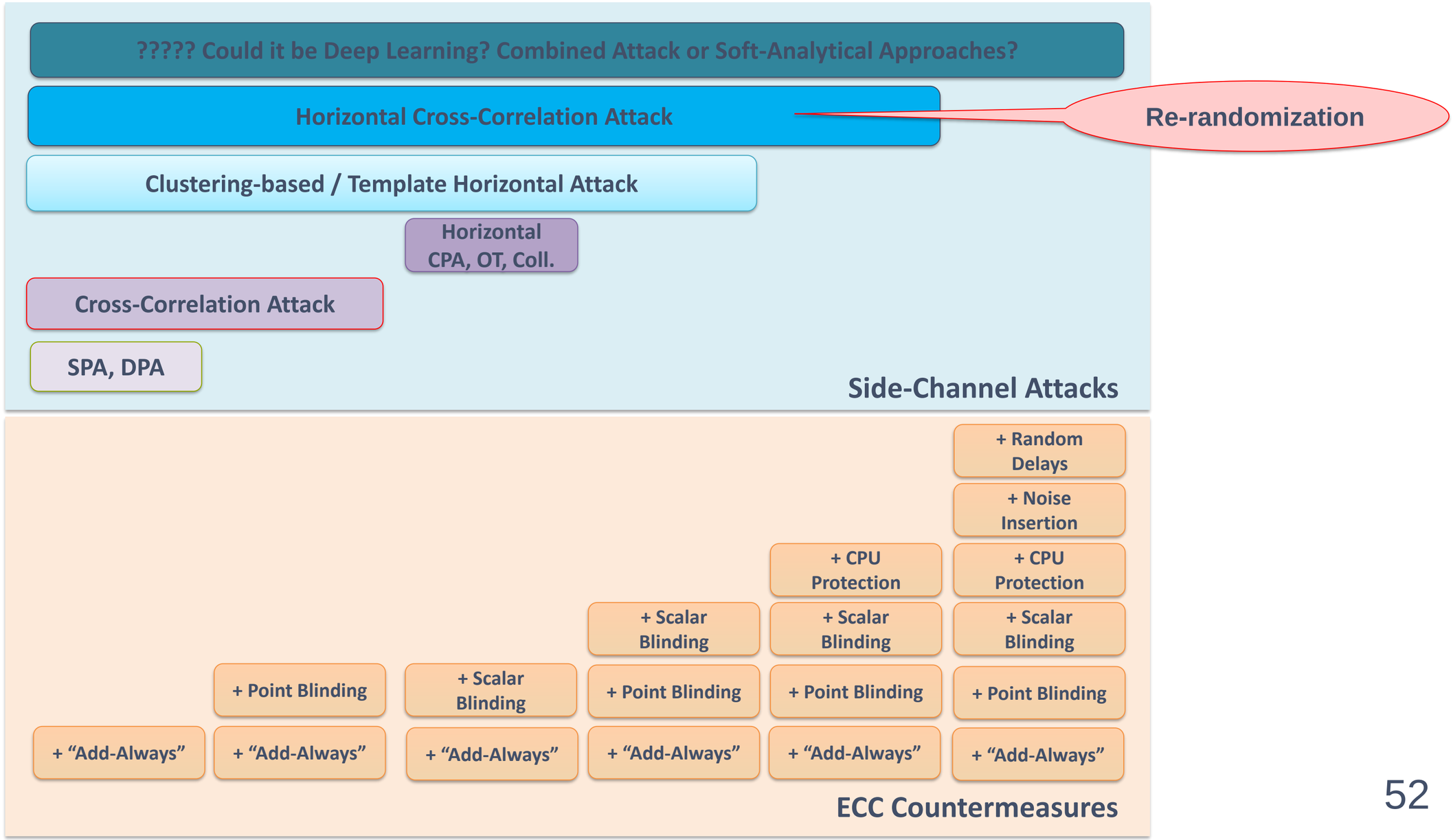
[1] SERMA Safety and Security, France, {m.carbone,v.conin}@serma.com
[2] CEA LETI, France, {cecile.dumas,marie-angela.cornelie}@cea.fr
[3] Thales ITSEF, France, {francois.dassance,guillaume.dufresne,alexandre.venelli}@thalesgroup.com
[4] ANSSI, France, emmanuel.prouff@ssi.gouv.fr

Abstract. This paper presents the results of several successful profiled side-channel attacks against a secure implementation of the RSA algorithm. The implementation was running on a ARM Core SC 100 completed with a certified EAL4+ arithmetic co-processor. The analyses have been conducted by three experts' teams, each working on a specific attack path and exploiting information extracted either from the electromagnetic emanation or from the power consumption. A particular attention is paid to the description of all the steps that are usually followed during a security evaluation by a laboratory, including the acquisitions and the observations pre-

2019

For ECC usually only 2 classes

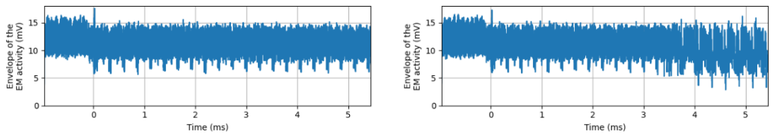
Attacks vs. Countermeasure



Application to Isogenies

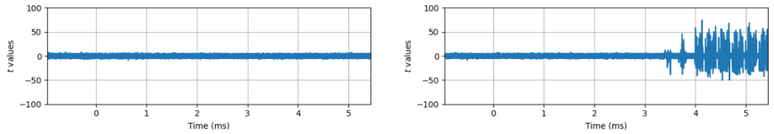
Attacks on Isogeny-based Schemes

- Zero-value attack – analogical to ECC case.



(a) Trace for the wrong hypothesis.

(b) Trace for the correct hypothesis.



(c) t-test for the wrong hypothesis.

(d) t-test for the correct hypothesis.

Figure 2: Experimental results for an attack on one bit.

SIKE Channels

Zero-Value Side-Channel Attacks on SIKE

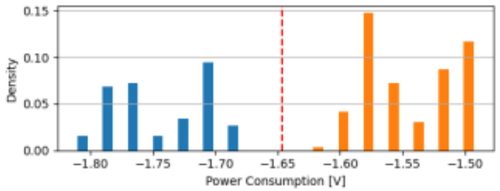
Luca De Feo[1], Nadia El Mrabet[2], Aymeric Genêt[34], Novak Kaluderović[3], Natacha Linard de Guertechin[5], Simon Pontié[6] and Élise Tasso[6]

[1] IBM Research Europe, Zürich, Switzerland, ches22@defeo.lu
[2] Mines Saint-Étienne, CEA-Tech, Centre CMP, Gardanne, France, nadia.el-mrabet@emse.fr
[3] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, aymeric.genet@epfl.ch,novak.kaluderovic@epfl.ch
[4] Nagra Kudelski Group, Cheseaux-sur-Lausanne, Switzerland, aymeric.genet@nagra.com
[5] CYSEC SA, Lausanne, Switzerland, natacha.linard@cysec.com
[6] CEA Tech, Centre CMP, Équipe Commune CEA Tech - Mines Saint-Étienne, F-13541 Gardanne, France; Université Grenoble Alpes, CEA-Leti, F-38000 Grenoble, France, elise.tasso2@cea.fr,simon.pontie@cea.fr

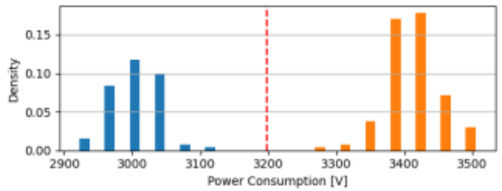
Abstract. We present new side-channel attacks on SIKE, the isogeny-based candidate in the NIST PQC competition. Previous works had shown that SIKE is vulnerable to differential power analysis, and pointed to coordinate randomization as an effective countermeasure. We show that coordinate randomization alone is not sufficient, because SIKE is vulnerable to a class of attacks similar to refined power analysis in elliptic curve cryptography, named *zero-value attacks*. We describe and confirm in the

2022

- Clustering against swap points function (like in cswap) in three point ladder



(a) $t = 1943$



(b) $f = 0.92$ [MHz]

Fig. 5: Example of a power sample distributions ($\ell = 0$). The threshold (in red) was found by Algorithm 3.

Single-trace clustering power analysis of the point-swapping procedure in the three point ladder of Cortex-M4 SIKE

Aymeric Genêt[1,2] and Novak Kaluđerović[1]

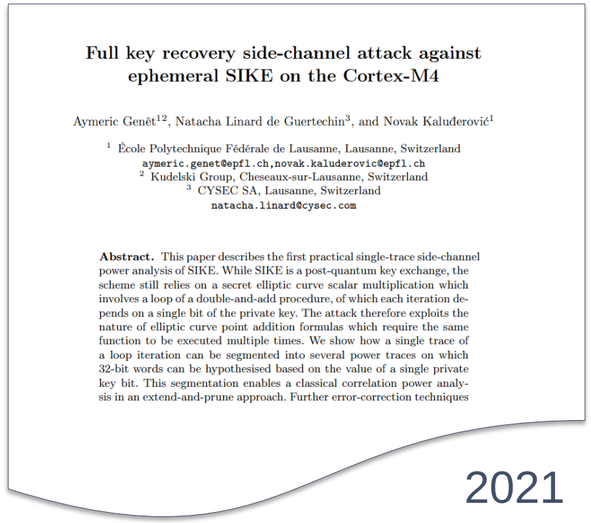
[1] École Polytechnique Fédérale de Lausanne, Ecublens, Switzerland
{aymeric.genet, novak.kaluderovic}@epfl.ch
[2] Kudelski Group, Cheseaux-sur-Lausanne, Switzerland
aymeric.genet@nagra.com

Abstract. In this paper, the recommended implementation of the post-quantum key exchange SIKE for Cortex-M4 is attacked through power analysis with a single trace by clustering with the k-means algorithm the power samples of all the invocations of the elliptic curve point swapping function in the constant-time coordinate-randomized three point ladder. Because each sample depends on whether two consecutive bits

2022

Attacks on Isogeny-based Schemes cont’d

- Correlation Power Analysis on SIKE



Full key recovery side-channel attack against ephemeral SIKE on the Cortex-M4

Aymeric Genêt[1,2], Natacha Linard de Guertechin[3], and Novak Kaluđerović[1]

[1] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
aymeric.genet@epfl.ch,novak.kaluderovic@epfl.ch
[2] Kudelski Group, Cheseaux-sur-Lausanne, Switzerland
[3] CYSEC SA, Lausanne, Switzerland
natacha.linard@cysec.com

Abstract. This paper describes the first practical single-trace side-channel power analysis of SIKE. While SIKE is a post-quantum key exchange, the scheme still relies on a secret elliptic curve scalar multiplication which involves a loop of a double-and-add procedure, of which each iteration depends on a single bit of the private key. The attack therefore exploits the nature of elliptic curve point addition formulas which require the same function to be executed multiple times. We show how a single trace of a loop iteration can be segmented into several power traces on which 32-bit words can be hypothesised based on the value of a single private key bit. This segmentation enables a classical correlation power analysis in an extend-and-prune approach. Further error-correction techniques

2021

- Are there attacks not on SIKE?
  Fault Injection Attack, not specific scalar multiplication.

Disorientation faults in CSIDH

Gustavo Banegas[1], Juliane Krämer[2], Tanja Lange[3,4], Michael Meyer[2], Lorenz Panny[4], Krijn Reijnders[5], Jana Sotáková[6], and Monika Trimoska[5]

[1] Inria and Laboratoire d’Informatique de l’Ecole polytechnique, Institut Polytechnique de Paris, Palaiseau, France
gustavo@cryptme.in
[2] University of Regensburg, Germany
juliane.kraemer@ur.de, michael@random-oracles.org
[3] Eindhoven University of Technology, Eindhoven, the Netherlands
tanja@hyperelliptic.org
[4] Academia Sinica, Taipei, Taiwan
lorenz@yx7.cc
[5] Radboud University, Nijmegen, The Netherlands
krijn@cs.ru.nl, monika.trimoska@ru.nl
[6] University of Amsterdam and QuSoft, Amsterdam, The Netherlands
j.s.sotakova@uva.nl

Abstract. We investigate a new class of fault-injection attacks against the CSIDH family of cryptographic group actions. Our *disorientation*

2023

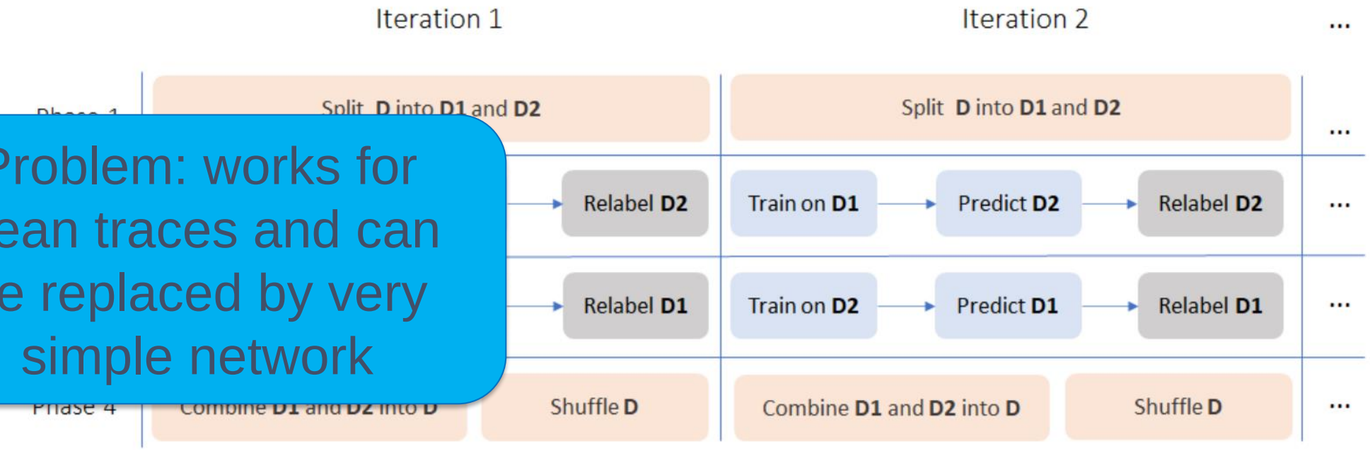
PROMISING RECENT APPROACHES

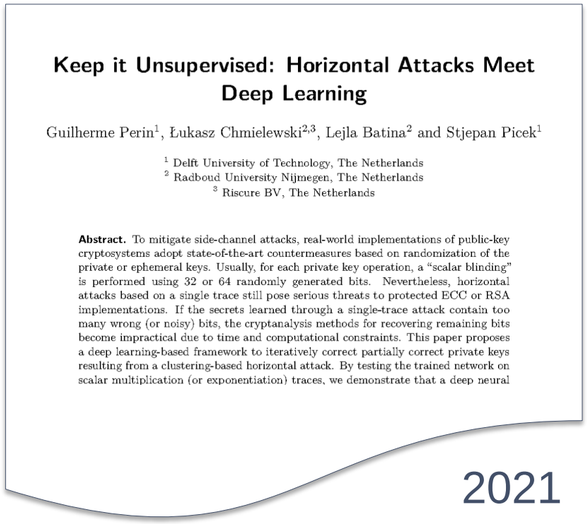
Unsupervised DL Framework

- Solution:
  1. Use simple Horizontal Clustering to recover noisy labels.
  2. Use DL on noisy labels to impr the accuracy of labeling.

Problem: works for clean traces and can be replaced by very simple network

Iterative Framework:





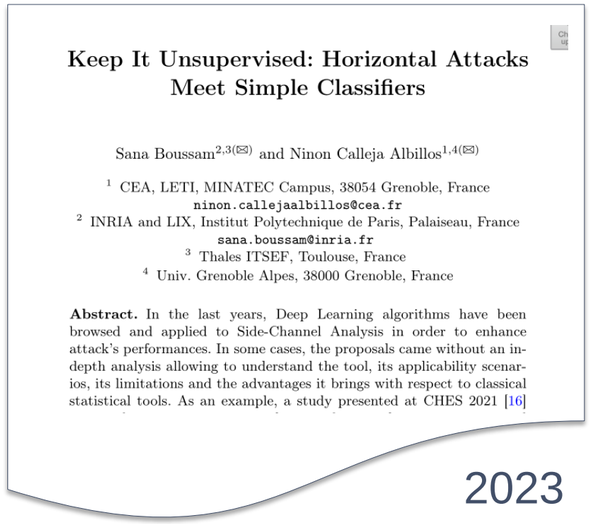
Keep it Unsupervised: Horizontal Attacks Meet Deep Learning

Guilherme Perin[1], Łukasz Chmielewski[2,3], Lejla Batina[2] and Stjepan Picek[1]

[1] Delft University of Technology, The Netherlands
[2] Radboud University Nijmegen, The Netherlands
[3] Riscure BV, The Netherlands

Abstract. To mitigate side-channel attacks, real-world implementations of public-key cryptosystems adopt state-of-the-art countermeasures based on randomization of the private or ephemeral keys. Usually, for each private key operation, a "scalar blinding" is performed using 32 or 64 randomly generated bits. Nevertheless, horizontal attacks based on a single trace still pose serious threats to protected ECC or RSA implementations. If the secrets learned through a single-trace attack contain too many wrong (or noisy) bits, the cryptanalysis methods for recovering remaining bits become impractical due to time and computational constraints. This paper proposes a deep learning-based framework to iteratively correct partially correct private keys resulting from a clustering-based horizontal attack. By testing the trained network on scalar multiplication (or exponentiation) traces, we demonstrate that a deep neural

2021



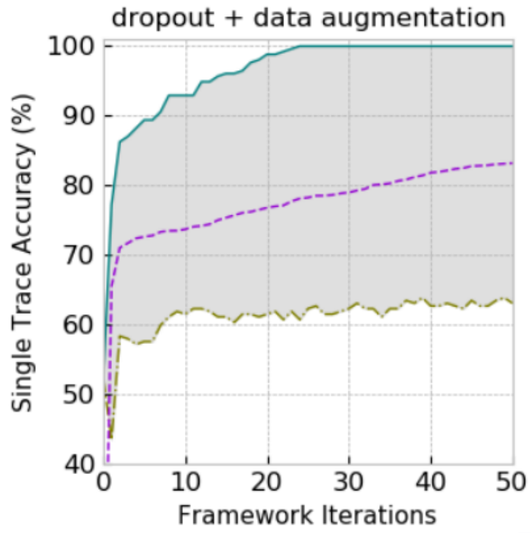
Keep It Unsupervised: Horizontal Attacks Meet Simple Classifiers

Sana Boussam[2,3(✉)] and Ninon Calleja Albillos[1,4(✉)]

[1] CEA, LETI, MINATEC Campus, 38054 Grenoble, France
ninon.callejaalbillos@cea.fr
[2] INRIA and LIX, Institut Polytechnique de Paris, Palaiseau, France
sana.boussam@inria.fr
[3] Thales ITSEF, Toulouse, France
[4] Univ. Grenoble Alpes, 38000 Grenoble, France

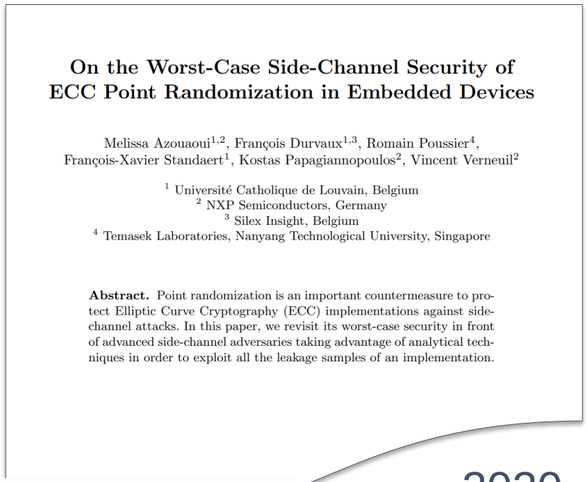
Abstract. In the last years, Deep Learning algorithms have been browsed and applied to Side-Channel Analysis in order to enhance attack's performances. In some cases, the proposals came without an in-depth analysis allowing to understand the tool, its applicability scenarios, its limitations and the advantages it brings with respect to classical statistical tools. As an example, a study presented at CHES 2021 [16]

2023



Soft analytical side-channel attack (SASCA) on ECC

- Similar to the Big Mac attack
- Template Attacks on many intermediates from a single trace
- Side-channel information throughout the factor graph using the Belief Propagation (BP) the distribution of the secret key, given the distributions of intermediates.
- Leaky targets (e.g., 8-bit or 16-bit microcontrollers), but there are also results on 32-bit (but not for ECC)



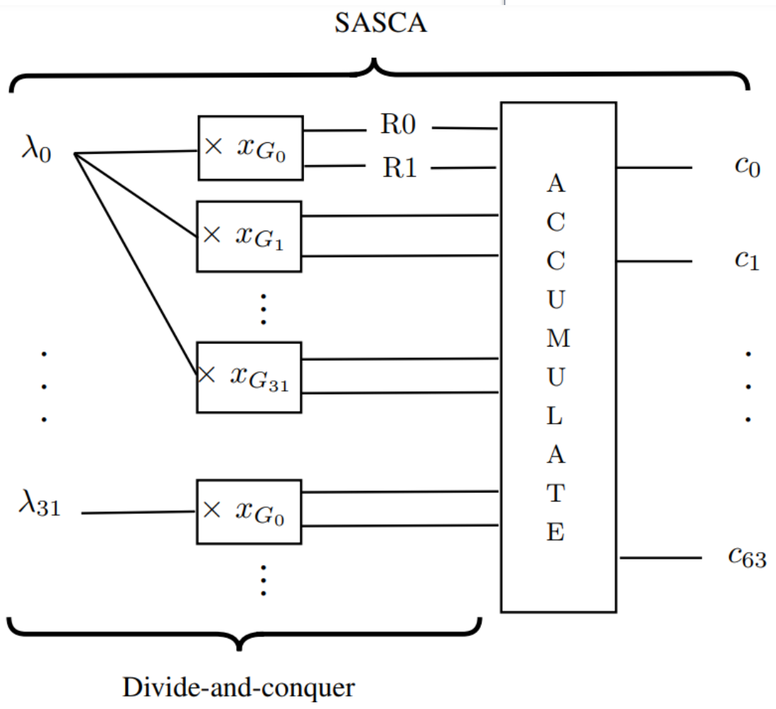
On the Worst-Case Side-Channel Security of ECC Point Randomization in Embedded Devices

Melissa Azouaoui[1,2], François Durvaux[1,3], Romain Poussier[4], François-Xavier Standaert[1], Kostas Papagiannopoulos[2], Vincent Verneuil[2]

[1] Université Catholique de Louvain, Belgium
[2] NXP Semiconductors, Germany
[3] Silex Insight, Belgium
[4] Temasek Laboratories, Nanyang Technological University, Singapore

Abstract. Point randomization is an important countermeasure to protect Elliptic Curve Cryptography (ECC) implementations against side-channel attacks. In this paper, we revisit its worst-case security in front of advanced side-channel adversaries taking advantage of analytical techniques in order to exploit all the leakage samples of an implementation.

2020



So what about Titan and Eucleak?

6

- Black-box attacks on protected implementations
- Manual Reverse Engineering (RE) based on side-channel traces
- Open profile/similar devices to the attacked ones
- Searching for issues
- Single-trace AI-assisted attacks on single traces

- Read: https://ninjalab.io/eucleak/

So what about Eucleak?

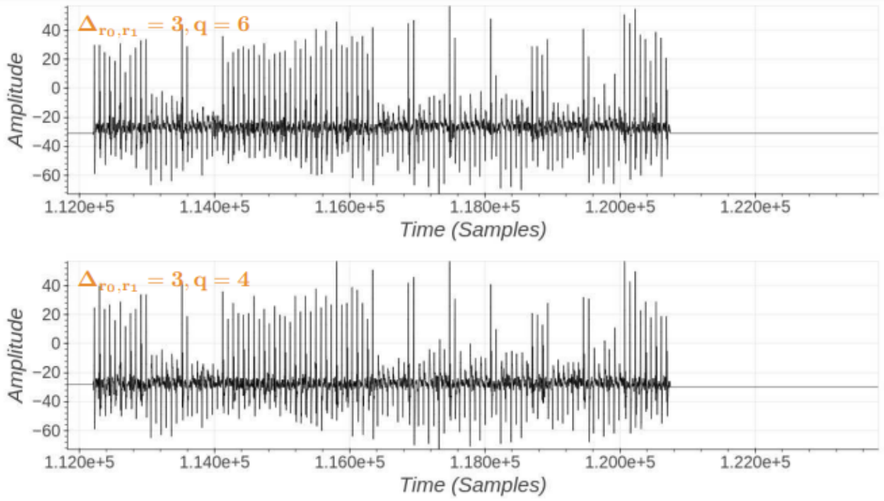


Figure 4.9: `Feitian A22 JavaCard` – ECDSA Signature Verification – EEA Computation – Resynchronized Iteration – $\Delta_{r_0,r_1} = 3$, quotient $= 6$ (Top) - $\Delta_{r_0,r_1} = 3$, quotient $= 4$ (Bottom)

4.2.2 Failed Attempts

Only considering the first 100 `EEA` traces from the ECDSA signature verification acquisition campaign (*i.e.* there is a total of 13893 `EEA` iterations), we manually regroup matching subtraces (for odd resynchronized iterations) for some small parameters choices $(\Delta_{r_0,r_1}, q) \in \{(1,1),(1,2),(1,3),(2,2),(2,3),(2,4),(2,5),(2,6),(2,7)\}$.

This is a long and error prone process:

- for a parameter choice (Δ_{r_0,r_1}, q), we extract all subtraces that match the parameters (omitting the *compact* iterations, and only selecting the odd iterations);
- from this set of subtraces, manually divide them into matching groups where all subtraces look alike (this step requires a comfortable visualization tool). For our parameter choices, the number of groups can be 1 (*e.g.* $(\Delta_{r_0,r_1}, q) = (2,7)$), 2 (*e.g.* $(\Delta_{r_0,r_1}, q) = (1,1)$) or 3 (*e.g.* $(\Delta_{r_0,r_1}, q) = (2,2)$, see Figure 4.8).

For the *attack* phase, we first regrouped the subtraces that looked similar (this was done using an smoothing process in order remove small jitters, it is presented in the attack on `YubiKey 5Ci`, Section 5.4). Then, by hand (there is a total of 156 iterations in the target EEA execution), we looked for wrongly regrouped subtraces. Finally, for each group (again by hand), we tried to match the group with a reference set (by visual comparison with the reference subtraces obtained in the profiling step). This by-hand step took hours of work and would necessitate to be automatized if this attack were to be taken outside a laboratory. For our purpose (demonstrate that the attack is possible) it was the (painful but) shortest path.

Are there way to automatically reverse-engineer ECC implementation from side-channel traces?

CROCS

pyecsca

[pietska]

Reverse-engineering black-box elliptic curve cryptography via side-channel analysis

Jan Jancar, Vojtech Suchanek, Petr Svenda,
Vladimir Sedlacek, Łukasz Chmielewski

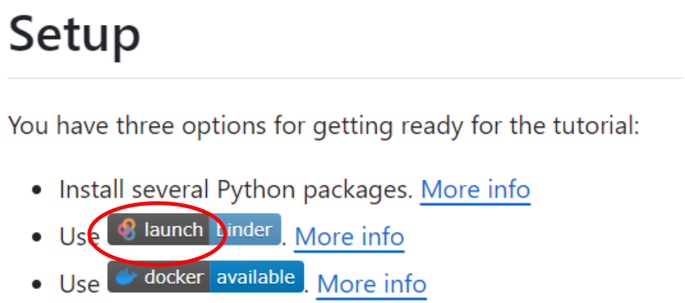
See you on Thursday!!!

Conclusions & Further reading

- Subjective list of important attacks and countermeasures
- Never-ending circle of countermeasures and attacks
  - One group inspires the other
- The is no golden solution for now
  - In the end, success depends on the designer and the person executing the attack
- Are there protected ECC libraries?
  - Closed-source ones - many
  - Open source: https://github.com/sca-secure-library-sca25519/sca25519
  - Recent list of attacks (2023): https://tches.iacr.org/index.php/TCHES/article/view/9962

Exercises

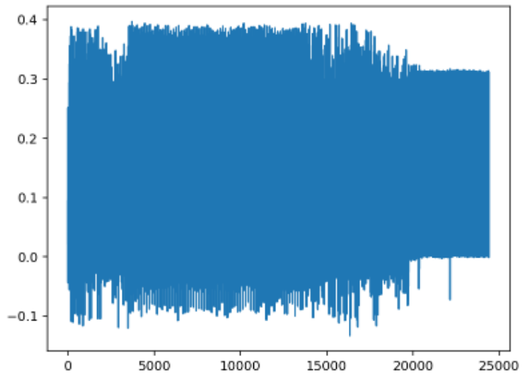
- Go here: https://github.com/J08nY/pyecsca-tutorial-ches2024/
  - Check the setup section.
    You can use binder.



- SPA on simulation and some simple analysis of Curve25519 traces.
- After it runs, you can perform:
  - start.ipynb and
  - implementations.ipynb Exercise A
  - Be aware of memory issues in binder (restart kernel when necessary)
- If you want to do more then you need to do the full installation – please contact me in case of issues!

Exercises Extra

- If you want to see how to capture some ECC traces, see:
  - https://github.com/sca-secure-library-sca25519/sca25519/tree/main/cw
- If you want to see a short demo on that – ask me. I have some simple board with me, and I am happy to show an acquisition ☺



- Extra exercises on “Practical Side-Channel Attacks on Real-World ECDSA Implementations”:
  - https://github.com/J08nY/cwnano-micro-ecc