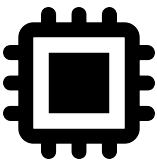
Thomas Roche
NinjaLab

Journées Codage et Cryptographie
Pornichet, FR – April 1st, 2025

Secure Elements

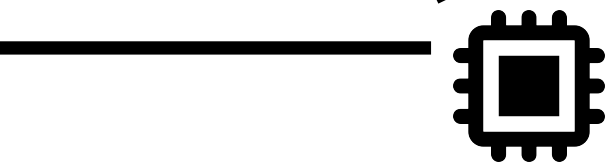


Secure Elements

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

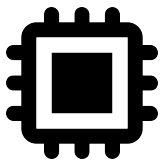
Remote Attacker



Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker



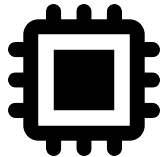
Simple HW
Simple SW
Simple I/O
Formal Methods

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

φ Attacker

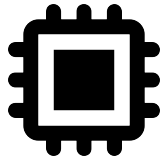


Simple HW
Simple SW
Simple I/O
Formal Methods

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker



φ Attacker

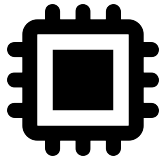
Simple HW
Simple SW
Simple I/O
Formal Methods

Side-Channel
Fault Injection
Invasive

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker



φ Attacker

Simple HW
Simple SW
Simple I/O
Formal Methods

Side-Channel
Fault Injection
Invasive

HW CMs
SW/Crypto CMs

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

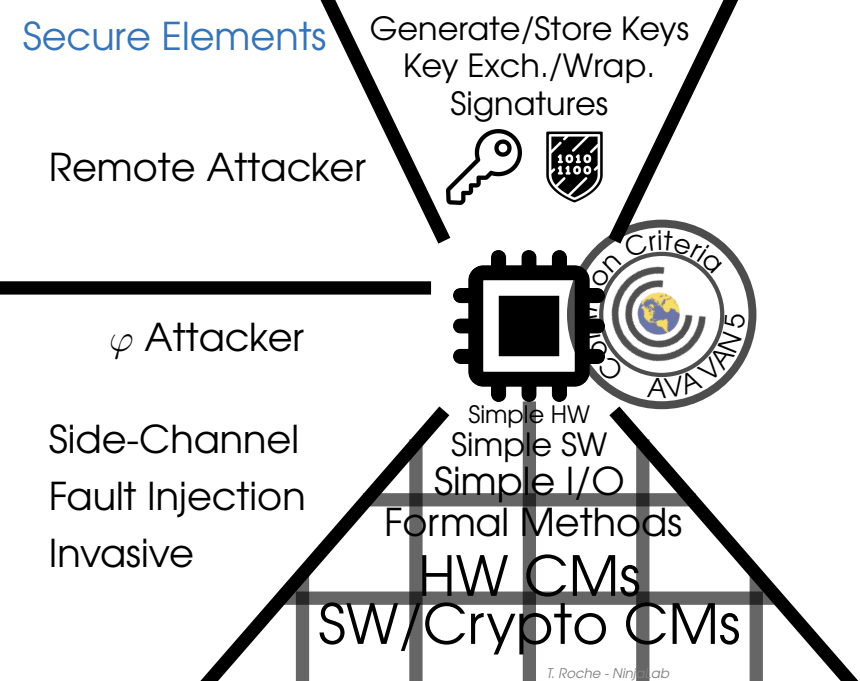
Remote Attacker

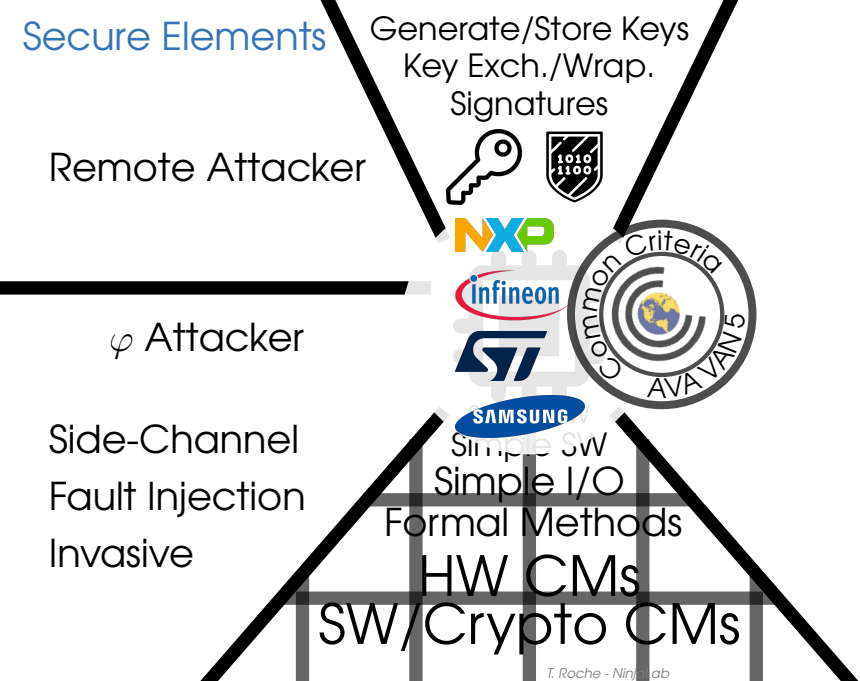
φ Attacker

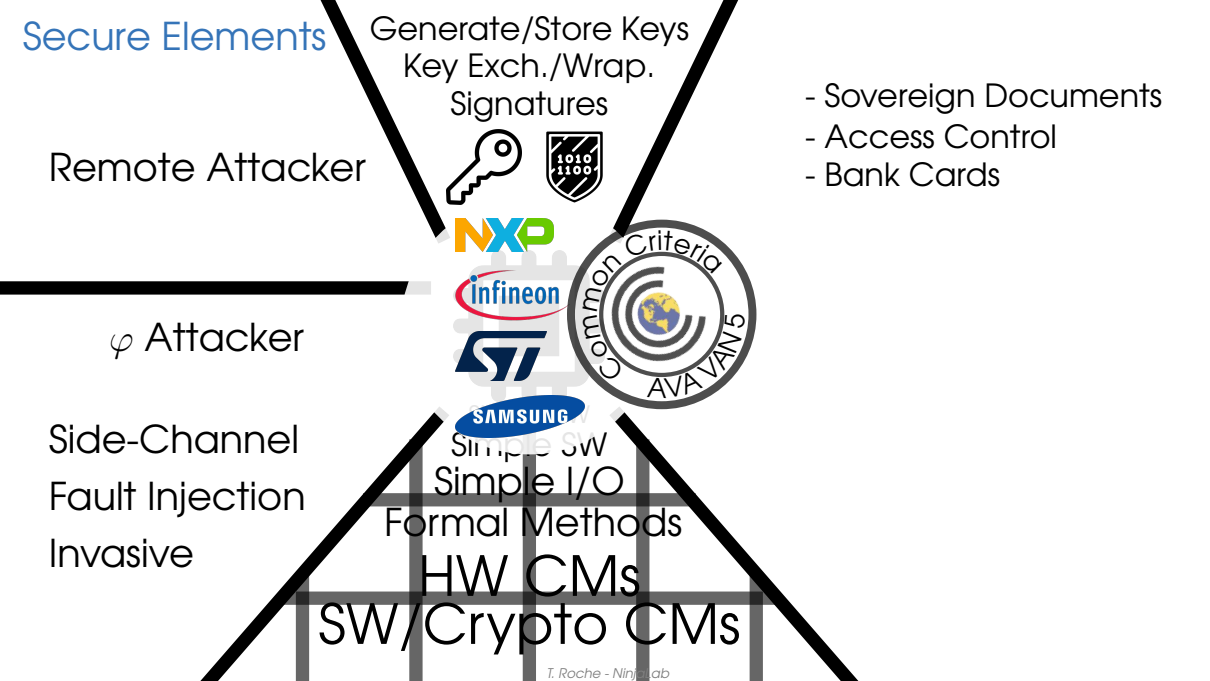
Simple HW
Simple SW
Simple I/O
Formal Methods

Side-Channel
Fault Injection
Invasive

HW CMs
SW/Crypto CMs


Secure Elements
Generate/Store Keys
Key Exch./Wrap.
Signatures
Remote Attacker
1010
1100
Common Criteria
EAL5
AVA VAN5
φ Attacker
Simple HW
Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs
Side-Channel
Fault Injection
Invasive
T. Roche - Ninjalab


Secure Elements
Generate/Store Keys
Key Exch./Wrap.
Signatures
Remote Attacker
NXP
infineon
SAMSUNG
Common Criteria
AVA VAN 5
φ Attacker
Side-Channel
Fault Injection
Invasive
Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs
T. Roche - NinjaLab


Secure Elements
Generate/Store Keys
Key Exch./Wrap.
Signatures
Remote Attacker
- Sovereign Documents
- Access Control
- Bank Cards
NXP
infineon
SAMSUNG
Common Criteria
AVA VAN 5
φ Attacker
Side-Channel
Fault Injection
Invasive
Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs
T. Roche - Ninjalab

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

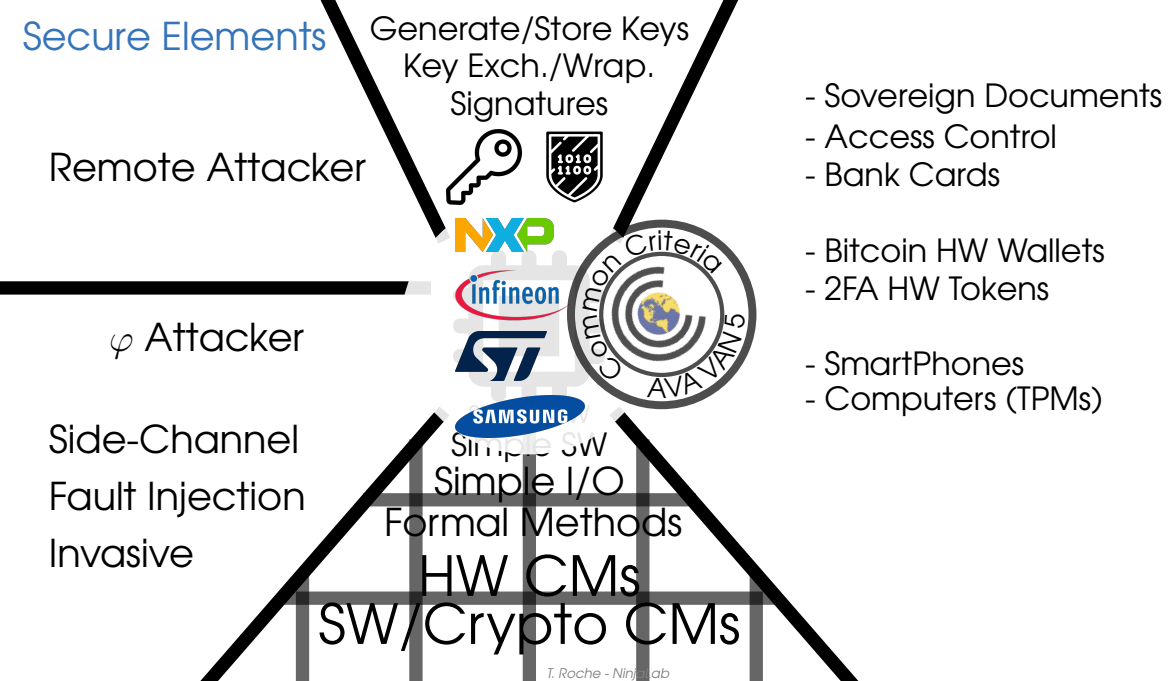
φ Attacker

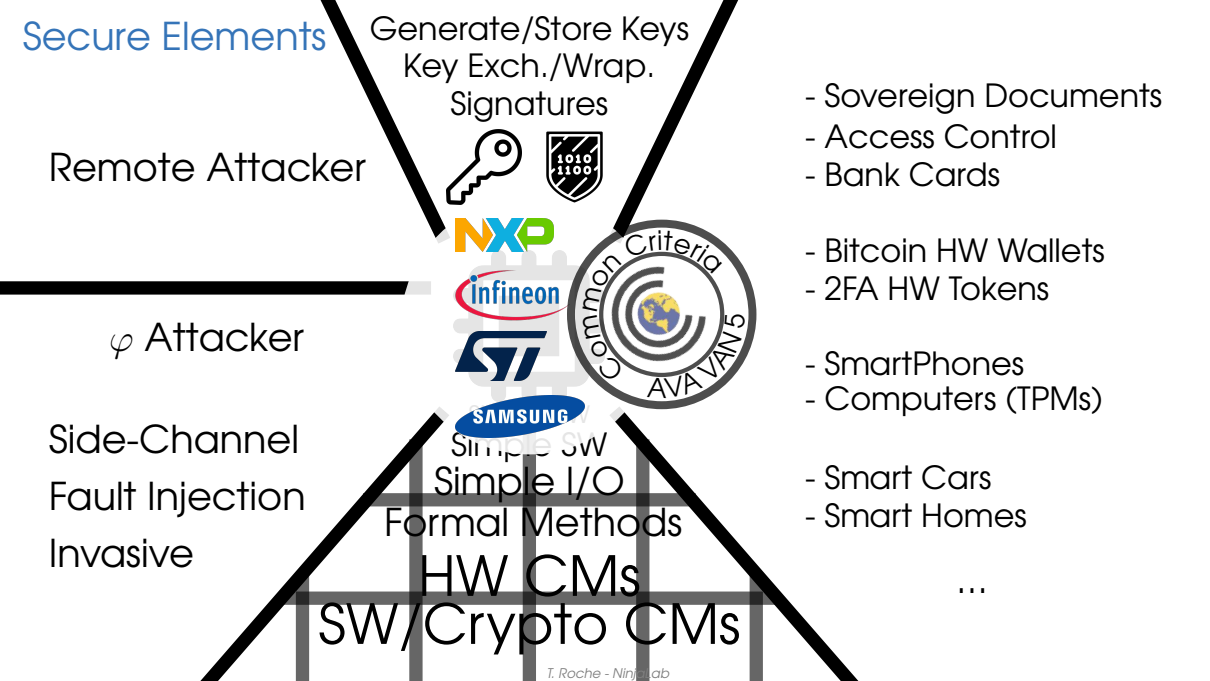
Side-Channel
Fault Injection
Invasive

Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs

- Sovereign Documents
- Access Control
- Bank Cards

- Bitcoin HW Wallets
- 2FA HW Tokens


Secure Elements
Generate/Store Keys
Key Exch./Wrap.
Signatures
Remote Attacker
NXP
infineon
SAMSUNG
Common Criteria
AVA VAN5
- Sovereign Documents
- Access Control
- Bank Cards
- Bitcoin HW Wallets
- 2FA HW Tokens
- SmartPhones
- Computers (TPMs)
φ Attacker
Side-Channel
Fault Injection
Invasive
Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs
T. Roche - Ninjalab


Secure Elements
Generate/Store Keys
Key Exch./Wrap.
Signatures
Remote Attacker
NXP
infineon
SAMSUNG
Common Criteria
AVA VAN 5
φ Attacker
Side-Channel
Fault Injection
Invasive
Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs
- Sovereign Documents
- Access Control
- Bank Cards
- Bitcoin HW Wallets
- 2FA HW Tokens
- SmartPhones
- Computers (TPMs)
- Smart Cars
- Smart Homes
...
T. Roche - Ninjalab

A SIDE JOURNEY TO TITAN

Remote Attacker

Generate/Store Keys
Key Exch./Wrap.
Signatures

NXP

infineon

SAMSUNG

- Sovereign Documents
- Access Control
- Bank Cards

- Bitcoin HW Wallets
- 2FA HW Tokens

- SmartPhones
- Computers (TPMs)

- Smart Cars
- Smart Homes

...

φ Attacker

Side-Channel
Fault Injection
Invasive

Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs

EUCLEAK

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

- Sovereign Documents
- Access Control
- Bank Cards

- Bitcoin HW Wallets
- 2FA HW Tokens

- SmartPhones
- Computers (TPMs)

- Smart Cars
- Smart Homes

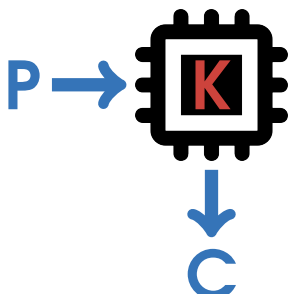
...

φ Attacker

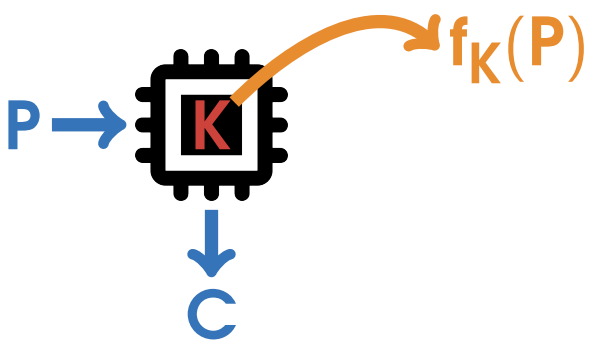
Side-Channel
Fault Injection
Invasive

Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs

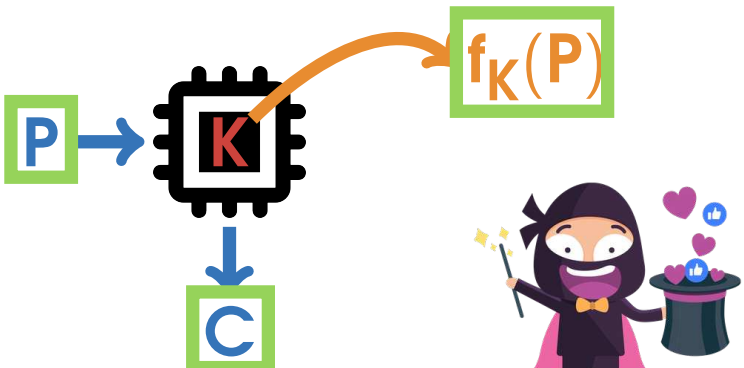
Side-Channel Analysis



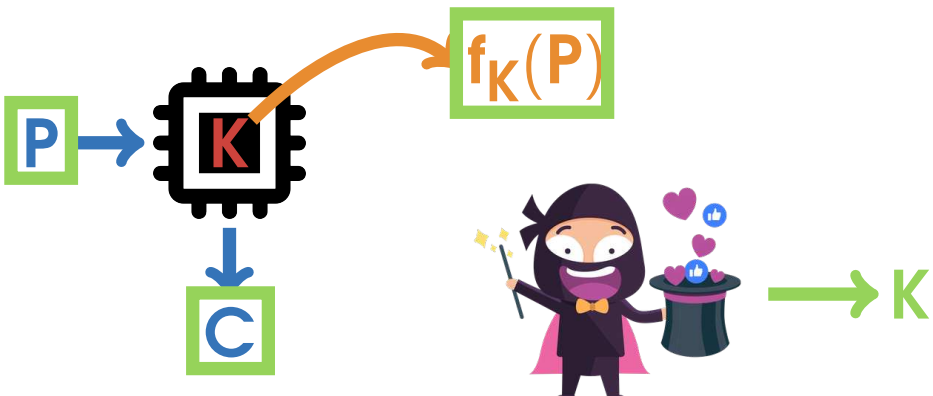
Side-Channel Analysis



Side-Channel Analysis

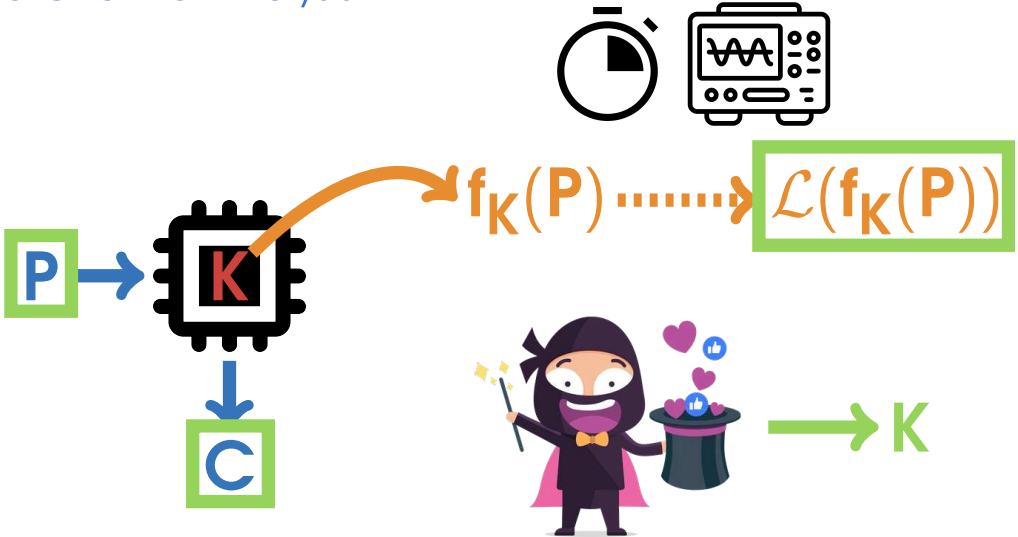


Side-Channel Analysis



Side-Channel Analysis

Side-Channel Analysis



FIDO Hardware Tokens

credits Yubico

- (2nd) Authentication Factor
- FIDO core crypto primitive is ECDSA:

  Elliptic Curve Digital Signature Algorithm

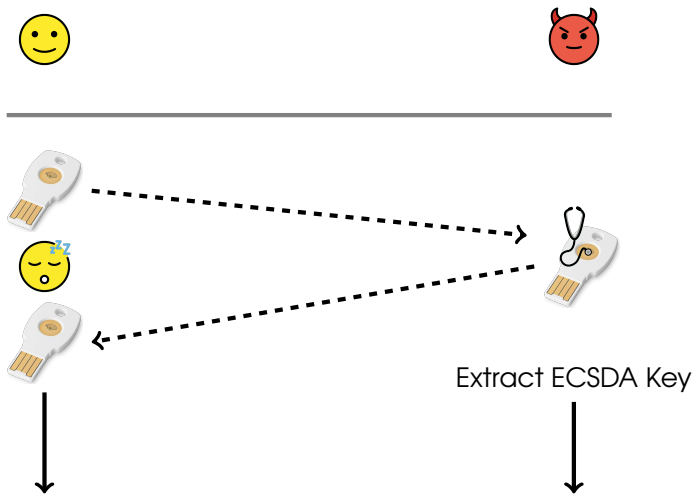
  - Generate ECDSA key-pairs
  - ECDSA Sign challenges
- Protect the ECDSA private keys

  ↪ Secure Element

FIDO U2F Protocol

- FIDO U2F: open standard for two-factor authentication
  - Hosted by FIDO alliance (historically developed by Google, Yubico and NXP)
- FIDO U2F protocol works in two steps - for each account:
  - *Registration* $\rightarrow$ ECDSA key pair generation
  - *Authentication* $\rightarrow$ ECDSA signature

Side-Channel Attack Scenario



ECDSA Signature Scheme

- Elliptic Curve E over $\mathbb{F}_p$ (base point $G_{(x,y)}$, order is N)
- Inputs: secret key d, the input message to sign $h = H(m)$
- randomly generate a nonce k in $\mathbb{Z}/N\mathbb{Z}$
- compute $Q_{(x,y)} = [k]G_{(x,y)}$
- denote by r the x-coordinate of Q: $r = Q_x$
- compute $s = k^{-1}(h + rd) \bmod N$
- return (r, s)

ECDSA Signature Scheme

- Elliptic Curve E over $\mathbb{F}_p$ (base point $G_{(x,y)}$, order is N)
- Inputs: secret key d, the input message to sign $h = H(m)$
- randomly generate a nonce k in $\mathbb{Z}/N\mathbb{Z}$
- compute $Q_{(x,y)} = [k]G_{(x,y)}$
  - randomly generate a random z in $\mathbb{Z}/p\mathbb{Z}$
  - random projection $G_{(x,y)} \to G_{(xz,yz,z)}$
  - compute $Q_{(x,y,z)} = [k]G_{(x,y,z)}$
  - inv projection $Q_{(x,y,z)} \to Q_{(xz^{-1},yz^{-1})}$
- denote by r the x-coordinate of Q: $r = Q_x$
- compute $s = k^{-1}(h + rd) \bmod N$
- return (r, s)

Thomas Roche and Victor Lomne and Camille Mutschler and Laurent Imbert

Usenix Security 2021

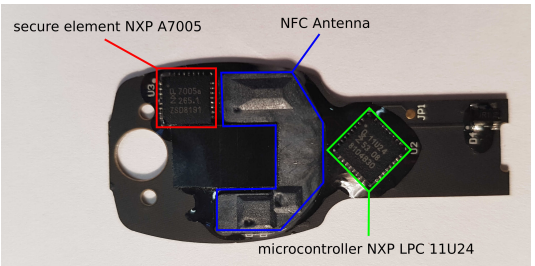
Product Description

- ***Google Titan Security Key***: hardware FIDO U2F token
- Hardware token to be used as 2FA for your Google account
  and many other services supporting FIDO U2F protocol
- 3 versions:
  - Left: micro-USB, NFC and BLE interfaces
  - Middle: USB type A and NFC interfaces
  - Right: USB type C interface

Google Titan Security Key Teardown

- Recto: HW manufacturer is **Feitian**

- Verso: secure element is ***NXP A7005a***



Other FIDO U2F products based on ***NXP A700x*** chip:

- Yubico Yubikey Neo
- Feitian K9, K13, K21, K40

Similarities with other NXP Products

- Several NXP JavaCard smartcards (JCOP) can be purchased on the web
- Those similar to ***NXP A700X*** are based on ***NXP P5x*** chips

NXP J3D081

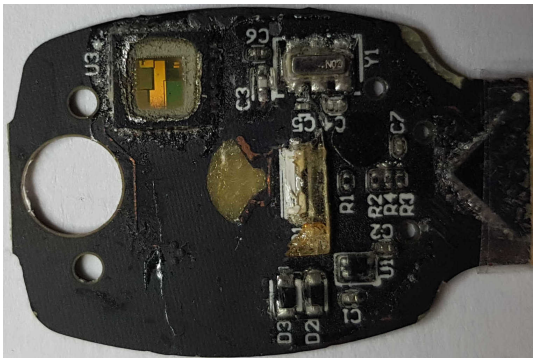
JCOP 2.4.2 R2
CC EAL5+ (2015)

- NXP J3D081_M59_DF and variants
- NXP J3A081 and variants
- NXP J2E081_M64 and variants
- NXP J3D145_M59 and variants
- NXP J3D081_M59 and variants
- NXP J3E145_M64 and variants
- NXP J3E081_M64_DF and variants

Rhea
2nd largest moon of Saturn after ***Titan***

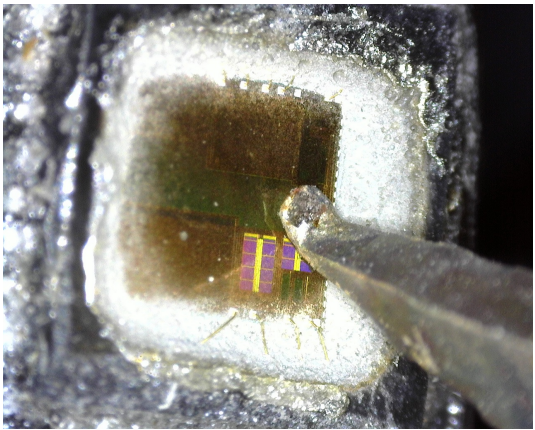
Titan / Rhea Package Openings

- ***Titan's NXP A700X***:
  - wet chemical opening
    aluminium tape + fuming nitric acid
  - ***Google Titan Security Key*** still alive !
- ***Rhea***:
  - mecanical opening
    scalpel + acetone
  - ***Rhea*** still alive !



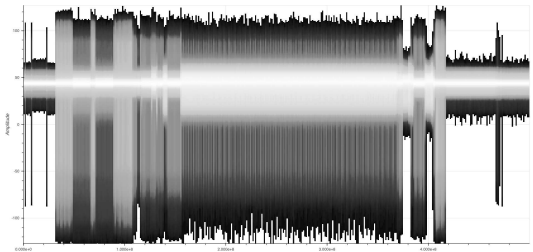
EM Side-Channel Acquisition Setup (about 12k€)

- **EM sensor:** Langer ICR HH 500-6 (diam. 500μm, freq. BW 2MHz to 6GHz)
- **Manual micro-manipulator:** Thorlabs PT3/M 3 axes (X-Y-Z)
- **Oscilloscope:** PicoScope 6404D, freq. BW 500MHz, SR 5GSa/s

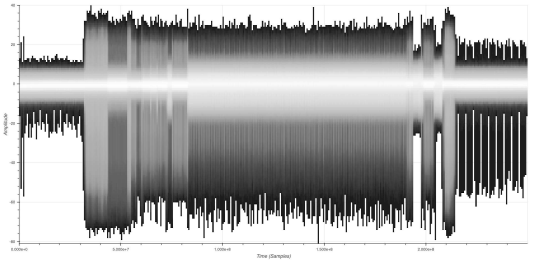


EM activity of ECDSA signature on Titan / Rhea

- ***Titan***

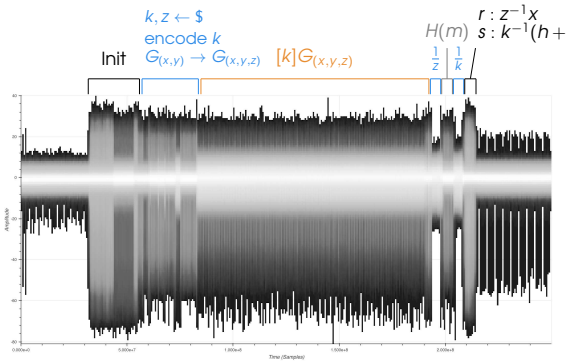


- ***Rhea***



- ECDSA signature EM activities on ***Titan*** and ***Rhea*** look very similar !

Rhea – Signature Alg. – EM Radiations



E = P256, SHA-256

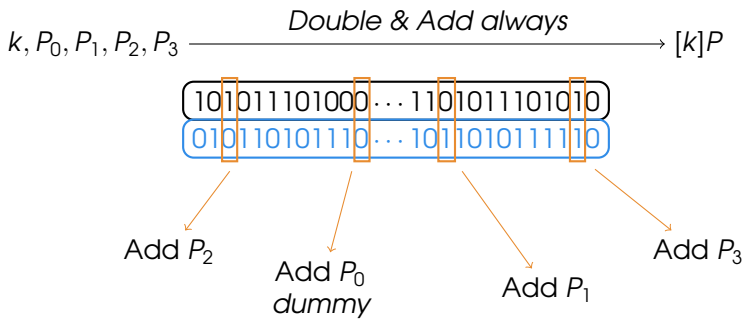
Scalar Multiplication $[k]G$

- Constant time algorithm
  Double-and-Add-Always
- 128 iterations for a 256-bit nonce k
  2 bits of k by iteration
- The scalar multiplication algorithm is a **left-to-right *comb method*** (of width 2)
- Scalars are not blinded.

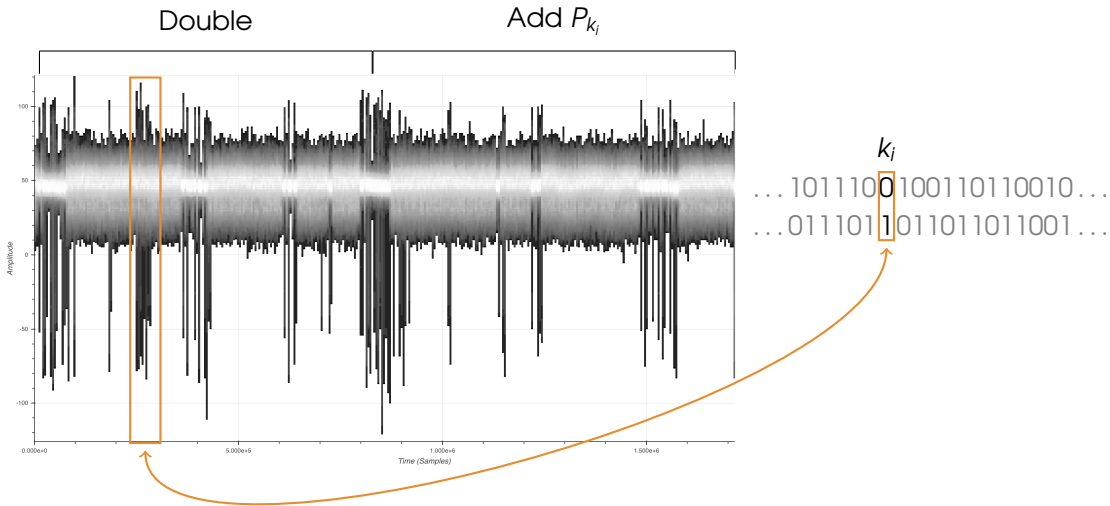
Left-To-Right Comb method (width 2)

Precomp: $P_0, P_1 = P, P_2 = [2^{128}]P,\ P_3 = P_2 + P_1$

$k =$ 101011101000 $\cdots$ 1101011101010 | 0101101011110 $\cdots$ 1011010111110

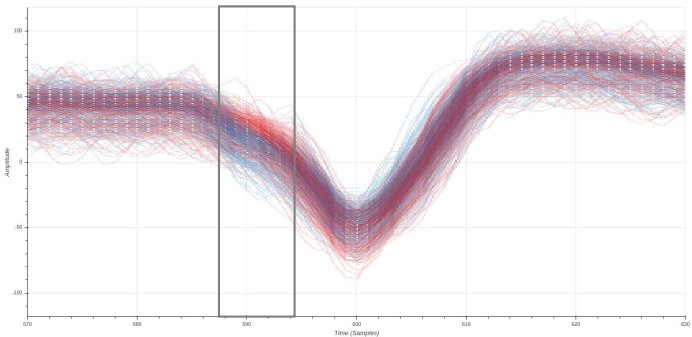


Rhea – Single Iteration – Leakage Area



Rhea – Leakage Illustration

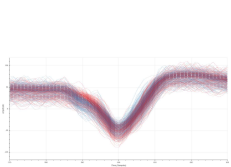
1000 Superposed Iterations – Zoom in Leakage Area

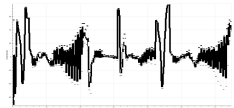


$k_i = \underline{0}\ x$

$k_i = \underline{1}\ x$

Single Trace Matching





99%

0
X

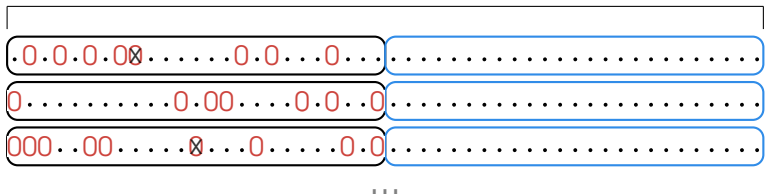
X
X

PoI extraction
Signal Processing

Unsupervised Clustering

256-bit Nonces

Identified 0 bits
27.5 by nonce
in average



Hidden Number Problems

- Recovering an ECDSA secret key given some partial knowledge on the nonces can be expressed as a Hidden Number Problem (HNP/EHNP)
- HNP and EHNP can be defined as games with an oracle
- The oracle reveals x and $f_m(\alpha x)$ for several random values of x

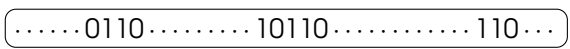
  The player should find the hidden value α
- HNP: f_m discloses the m most significant bits of αx

  $1101001010\cdots\cdots\cdots\cdots\cdots\cdots$
- EHNP: f_m discloses m bits of αx, not necessarily consecutive

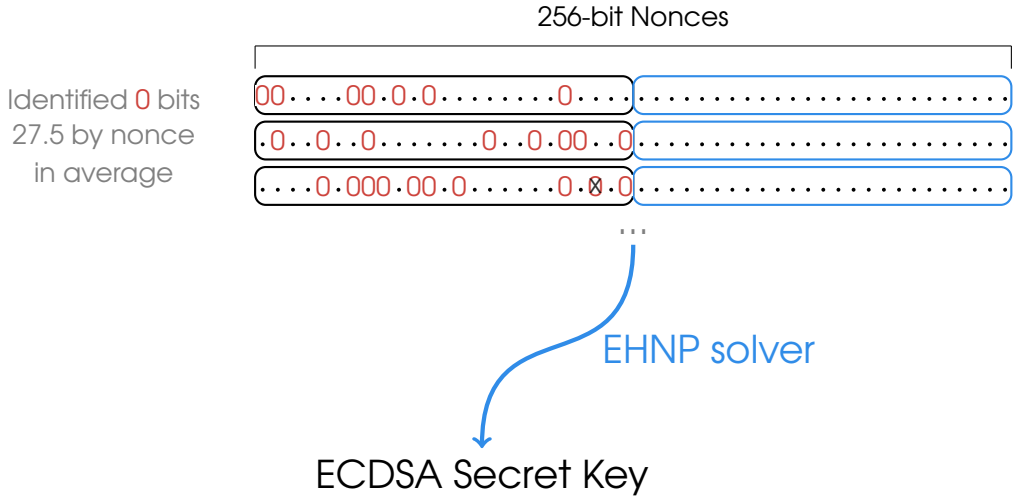
  $\cdots 1 \cdots 01 \cdots 0 \cdots 101 \cdots 0 \cdots 10 \cdots$

Solving (E)HNP

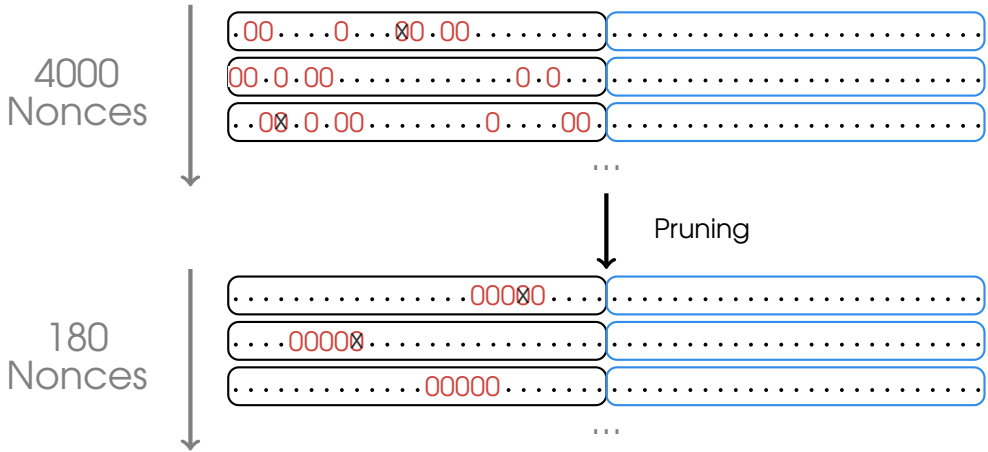
- (E)HNP can be reduced to instances of lattice-based problems (SVP, CVP) that may be solved using lattice reduction techniques (LLL, BKZ)
- # oracle queries and # known bits dictate the size of the lattice and the probability of success
- In practice EHNP can be solved when the m bits revealed by the oracle form blocks of sufficiently many consecutive bits



Solving (Extended) Hidden Number Problems



Rhea – Nonces Selection



Brute-Forcing the Key

- LLL reduction (for 80 signatures) takes about 100s
- 5 errors among 180 available signatures

$\hookrightarrow$ Brute-force attack on random subsets

Final Attack

- Acquisition of 4000 traces: $\sim 4h$
- Trace Processing: $\sim 4h$
- Brute-force attack: $\sim 20min$

Touchdown on Titan

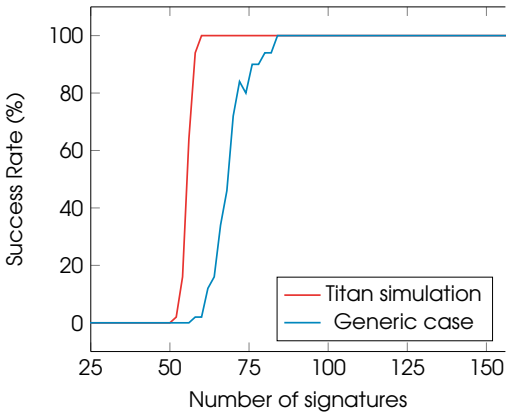
- Use Rhea parameters for PoI extraction
- Pruning: from 6000 signatures to 156
- 7 errors among 156 available signatures

$\hookrightarrow$ Brute-force attack on random subsets

Final Attack

- Acquisition of 6000 traces: $\sim 6h$
- Trace Processing: $\sim 6h$
- Brute-force attack: $\sim 30min$

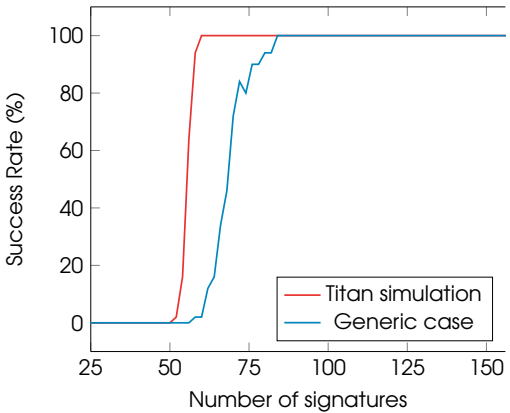
An Interesting Observation



Generic case *the known 5 bits can be at any position, between two unknown blocks*

Titan Simulation *the known 5 bits are restricted to the nonces upper-half*

An Interesting Observation

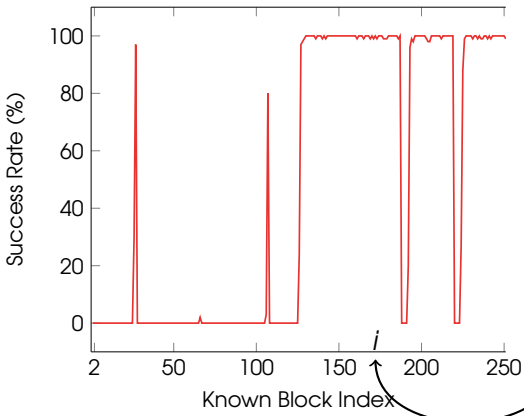


Generic case *the known 5 bits can be at any position, between two unknown blocks*

Titan Simulation *the known 5 bits are restricted to the nonces upper-half*

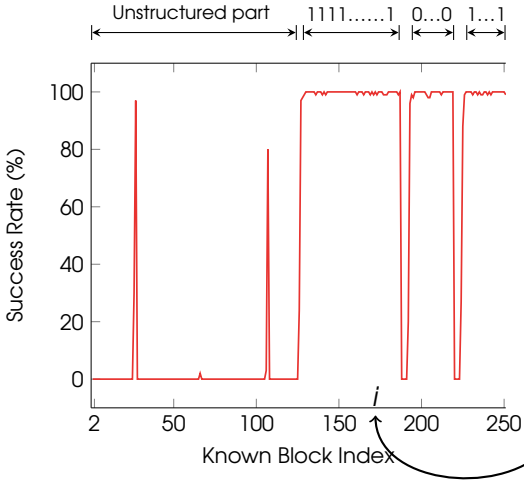
The position of the known block has an influence on the attack success rate

An Interesting Observation



Experiment *i*:
Attack success-rate (60 signatures)
with known block at bit position *i*

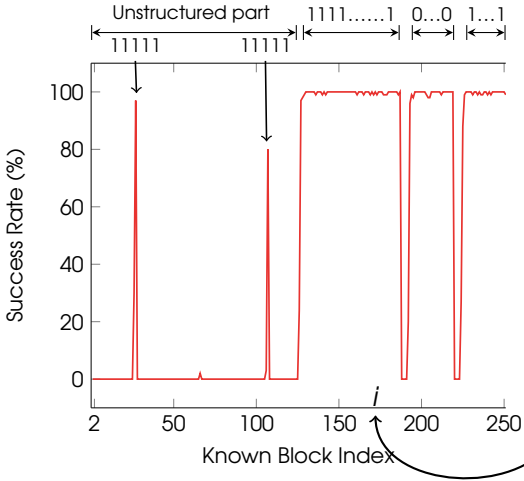
An Interesting Observation



Binary form of the curve order

Experiment *i*:
Attack success-rate (60 signatures)
with known block at bit position *i*

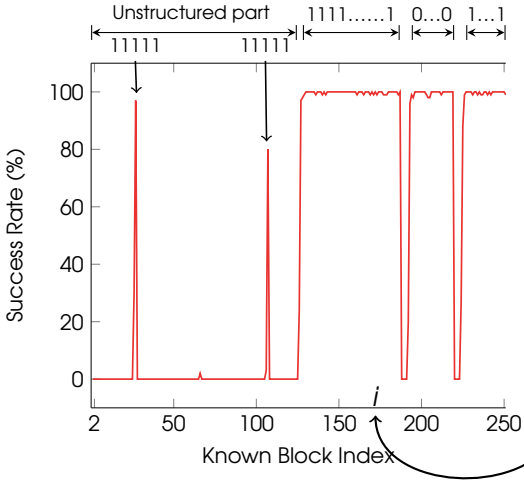
An Interesting Observation



Binary form of the curve order

Experiment *i*:
Attack success-rate (60 signatures)
with known block at bit position *i*

An Interesting Observation



Binary form of the curve order

Known block position w.r.t. Curve order impacts the inner structure of the lattice

- lattice volume unchanged
- short vector norm unchanged

Elliptic curves with structured orders containing large sequences of 0s or 1s are then easier to attack

Experiment *i*:
Attack success-rate (60 signatures) with known block at bit position *i*

Thomas Roche
To appear in IEEE S&P 2025

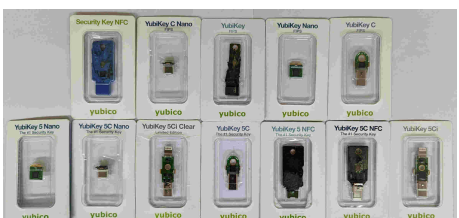
Agenda

Agenda

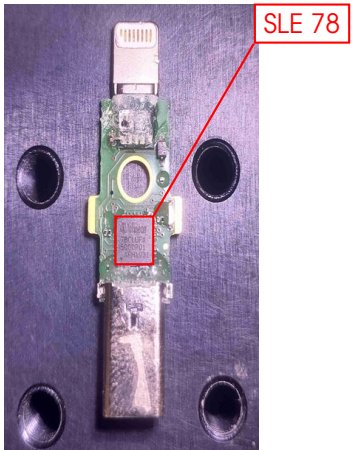
- Introduction
  - FIDO Hardware Tokens
  - Infineon SLE 78
  - Yubikey 5 Series
- A Side-Channel Vulnerability
  - Infineon ECDSA Observations
  - The Extended Euclidean Algorithm
  - Summary

Yubico FIDO Tokens

Most common security microcontrollers in FIDO Tokens are Infineon SLE78.

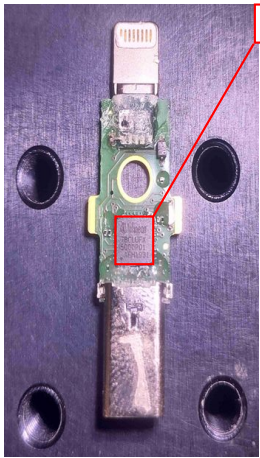


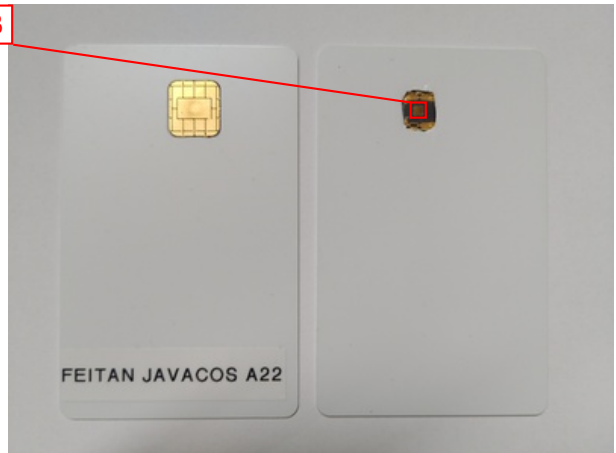
Yubikey 5Ci



credits Yubico

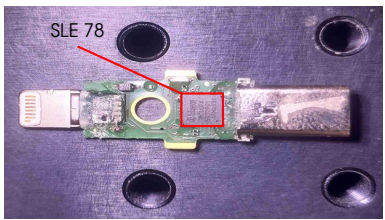
Yubikey 5Ci – FEITIAN A22 Open JavaCard



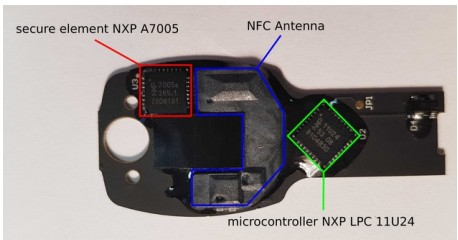


Infineon SLE 78

*"The **SLE 78 USB** is a cache-based pure **16-bit security controller** family designed to meet all secure USB token design requirements. (...) It enables certification levels up to **Common Criteria EAL6+ (high) and EMVCo**."* [1]



Yubikey 5Ci (SLE 78)

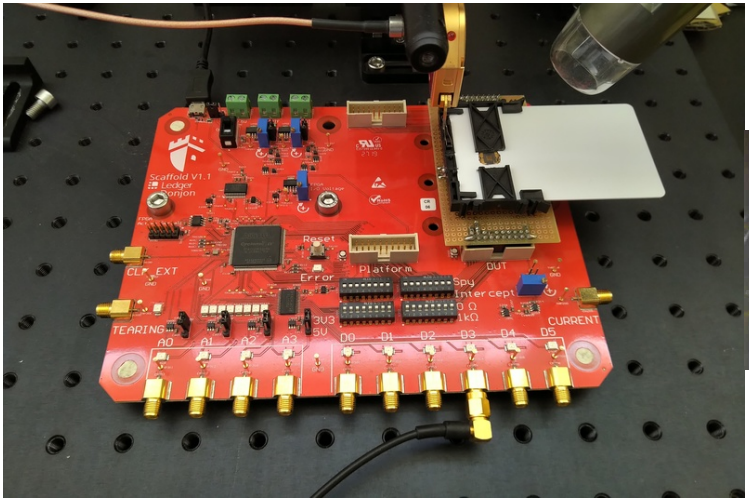


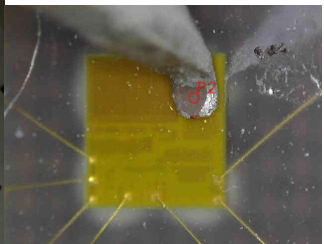
Google Titan Key (NXP A7005)

[1] https://www.infineon.com/cms/en/product/security-smart-card-solutions/security-controllers-for-usb-tokens/sle-78clufx5000ph/

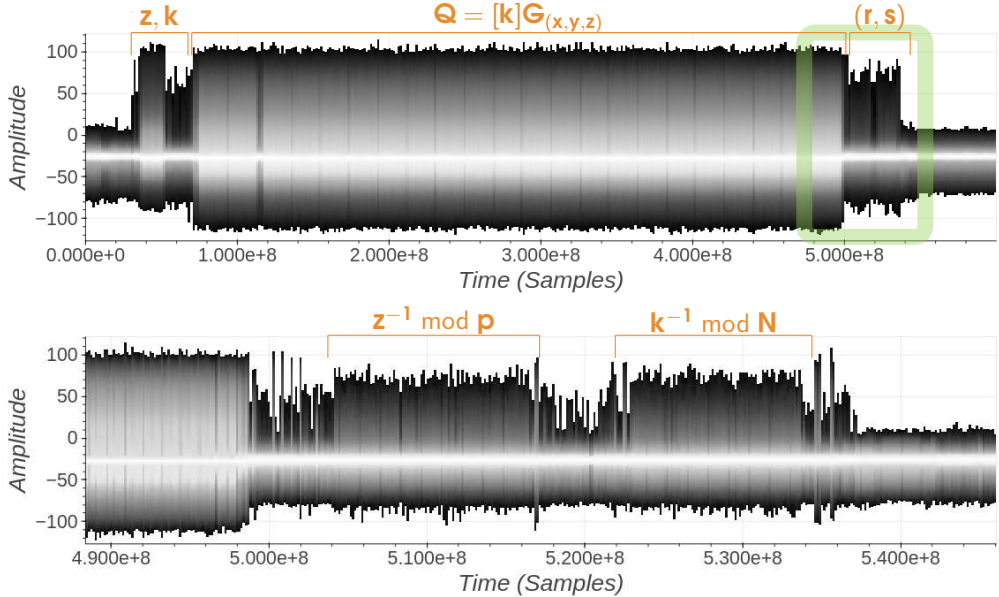
Agenda

FEITIAN A22 – EM Acquisitions

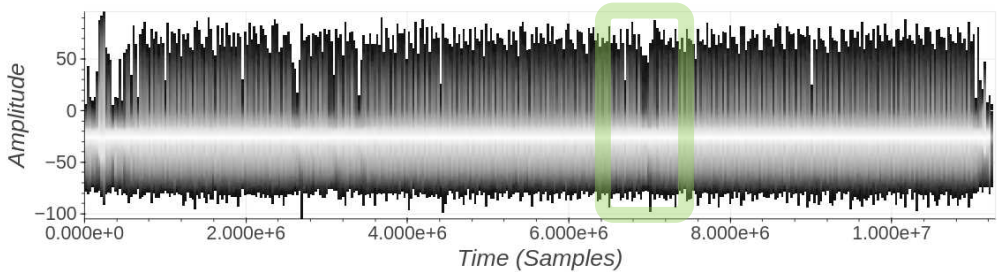


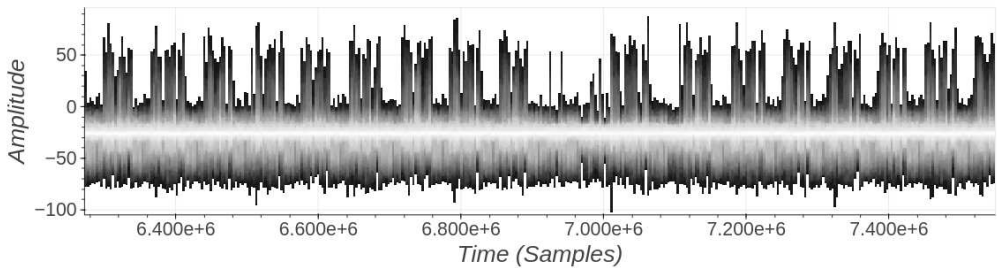


FEITIAN A22 – ECDSA Command – EM Radiations



FEITIAN A22 – k^{-1} mod N – EM Radiations





Extended Euclidean Algorithm

Input : v, n: two positive integers with $v \leq n$ and $gcd(v, n) = 1$
Output: $v^{-1} \bmod n$: the inverse of v modulo n

1 $r_0, r_1 \leftarrow n, v$
2 $t_0, t_1 \leftarrow 0, 1$
3 **while** $r_1 \neq 0$ **do**
4 $\quad q \leftarrow \mathtt{div}(r_0, r_1)$
5 $\quad r_0, r_1 \leftarrow r_1, r_0 - q.r_1$
6 $\quad t_0, t_1 \leftarrow t_1, t_0 - q.t_1$
7 **if** $t_0 < 0$ **then**
8 $\quad t_0 \leftarrow t_0 + n$
Return : t_0

Extended Euclidean Algorithm

Input : v, n: two positive integers with $v \leq n$ and $gcd(v, n) = 1$
Output: $v^{-1} \bmod n$: the inverse of v modulo n

1 $r_0, r_1 \leftarrow n, v$
2 $t_0, t_1 \leftarrow 0, 1$
3 **while** $r_1 \neq 0$ **do**
4 $\quad q \leftarrow \mathtt{div}(r_0, r_1)$
5 $\quad r_0, r_1 \leftarrow r_1, r_0 - q.r_1$
6 $\quad t_0, t_1 \leftarrow t_1, t_0 - q.t_1$
7 **if** $t_0 < 0$ **then**
8 $\quad t_0 \leftarrow t_0 + n$
Return : t_0

k is blinded with a 32-bit mask

$$m \xleftarrow{\$} \mathbb{Z}/2^{32}\mathbb{Z}^{\star}$$
$$k' = k \times m \bmod N$$
$$k'^{-1} = \mathtt{EEA}(k', N)$$
$$k^{-1} = k'^{-1} \times m \bmod N$$

Extended Euclidean Algorithm

Input : v, n: two positive integers with $v \leq n$ and $gcd(v, n) = 1$
Output: $v^{-1} \bmod n$: the inverse of v modulo n

1 $r_0, r_1 \leftarrow n, v$
2 $t_0, t_1 \leftarrow 0, 1$
3 **while** $r_1 \neq 0$ **do**
4 $\quad q \leftarrow \mathtt{div}(r_0, r_1)$
5 $\quad r_0, r_1 \leftarrow r_1, r_0 - q.r_1$
6 $\quad t_0, t_1 \leftarrow t_1, t_0 - q.t_1$
7 **if** $t_0 < 0$ **then**
8 $\quad t_0 \leftarrow t_0 + n$
Return : t_0

Input : a, b: two positive integers
Output : q: the quotient of the division of a by b

$r \leftarrow a$
$\ell \leftarrow len(r) - len(b)$
$q \leftarrow 0$
while $\ell \geq 0$ **do**
$\quad g \leftarrow \mathtt{sign}(r).2^{\ell}$
$\quad r \leftarrow r - g.b$
$\quad q \leftarrow q + g$
$\quad \ell \leftarrow len(r) - len(b)$
if $r < 0$ **then**
$\quad q \leftarrow q - 1$
$\quad r \leftarrow r + b$
Return : q

Extended Euclidean Algorithm

Input : v, n: two positive integers with $v \leq n$ and $gcd(v, n) = 1$
Output: $v^{-1} \bmod n$: the inverse of v modulo n

1 $r_0, r_1 \leftarrow n, v$
2 $t_0, t_1 \leftarrow 0, 1$
3 **while** $r_1 \neq 0$ **do**
4 $\quad q \leftarrow \mathtt{div}(r_0, r_1)$
5 $\quad r_0, r_1 \leftarrow r_1, r_0 - q.r_1$
6 $\quad t_0, t_1 \leftarrow t_1, t_0 - q.t_1$
7 **if** $t_0 < 0$ **then**
8 $\quad t_0 \leftarrow t_0 + n$
Return : t_0

Input : a, b: two positive integers
Output : q: the quotient of the division of a by b

$r \leftarrow a$
$\ell \leftarrow len(r) - len(b)$
$q \leftarrow 0$
while $\ell \geq 0$ **do**
$\quad g \leftarrow \mathtt{sign}(r).2^{\ell}$
$\quad r \leftarrow r - g.b$
$\quad q \leftarrow q + g$
$\quad \ell \leftarrow len(r) - len(b)$
if $r < 0$ **then**
$\quad q \leftarrow q - 1$
$\quad r \leftarrow r + b$
Return : q

Let's sum up

- Timing leakages in the EEA implementation that inverse ECDSA's nonce k.

Let's sum up

- Timing leakages in the EEA implementation that inverse ECDSA's nonce k.
- Nonce is blinded with a 32-bit multiplicative mask.

blinded nonce $\rightarrow$ nonce $\rightarrow$ private key

Let's sum up

- Timing leakages in the EEA implementation that inverse ECDSA's nonce k.
- Nonce is blinded with a 32-bit multiplicative mask.

  blinded nonce $\rightarrow$ nonce $\rightarrow$ private key
- The timing leakage is enough information to guess the blinded nonce.

Side-Channel Attack
on Ext. Euclidean Alg.

ninjalab.io/eucleak

Let's sum up

- Timing leakages in the EEA implementation that inverse ECDSA's nonce k.
- Nonce is blinded with a 32-bit multiplicative mask.

  blinded nonce $\rightarrow$ nonce $\rightarrow$ private key
- The timing leakage is enough information to guess the blinded nonce.

Side-Channel Attack
on Ext. Euclidean Alg.

ninjalab.io/eucleak

Agenda

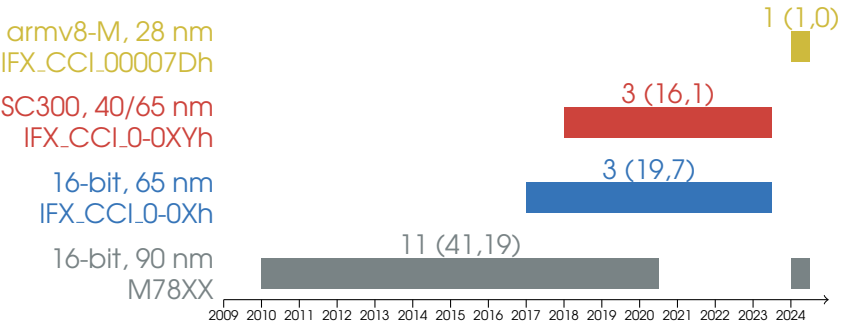
Introdu
- FIDO
- Infineon
- Yubikey 5 S

A Side-Ch
- Infineon ECDS
- The Extended
- Summary

Impact Analysis
- Infineon Security Microcontrollers
- Yubikey 5Ci
- Optiga Trust M
- Optiga TPM

Conclusions
- Summing up
- Mitigations
- Avenues Of Research
- Project Timeline

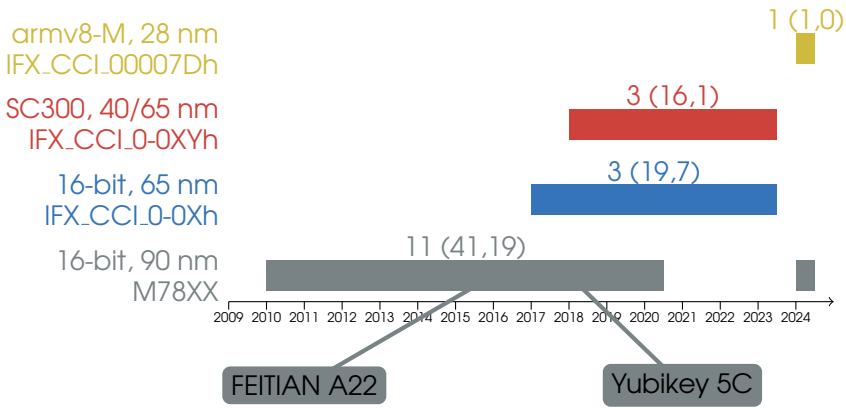
Infineon Security Microcontrollers (IC CC Certifications)



Legend: # IC (# Certification Reports, # Maintenance Reports)

Credits: www.bsi.bund.de, www.sec-certs.org

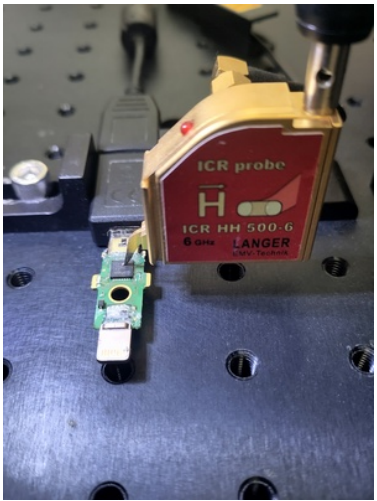
Infineon Security Microcontrollers (IC CC Certifications)



Legend: # IC (# Certification Reports, # Maintenance Reports)

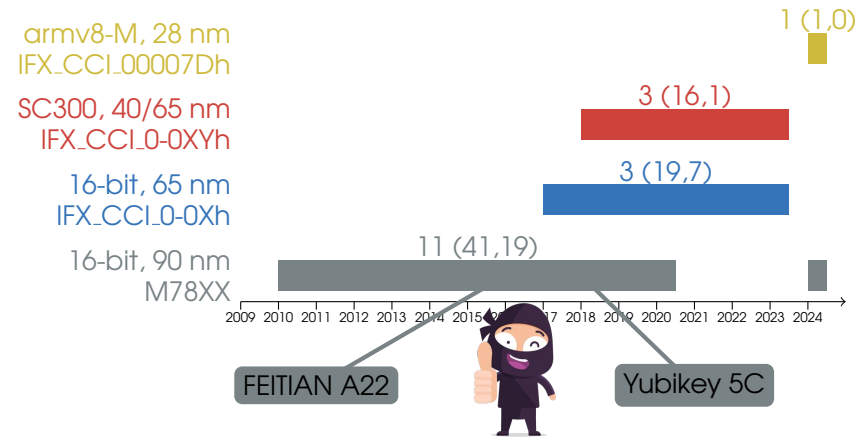
Credits: `www.bsi.bund.de`, `www.sec-certs.org`

Yubikey 5Ci – Side-channel Setup



credits Yubico

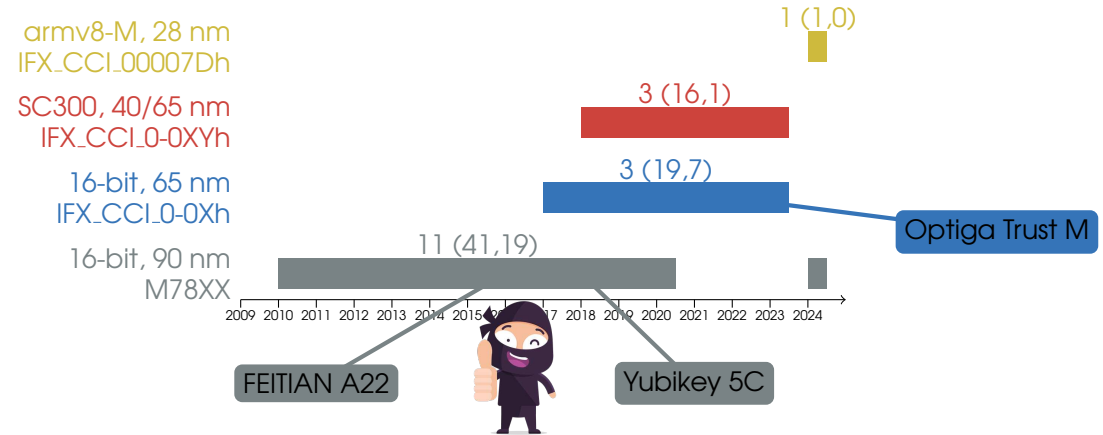
Infineon Security Microcontrollers (IC CC Certifications)



Legend: # IC (# Certification Reports, # Maintenance Reports)

Credits: `www.bsi.bund.de`, `www.sec-certs.org`

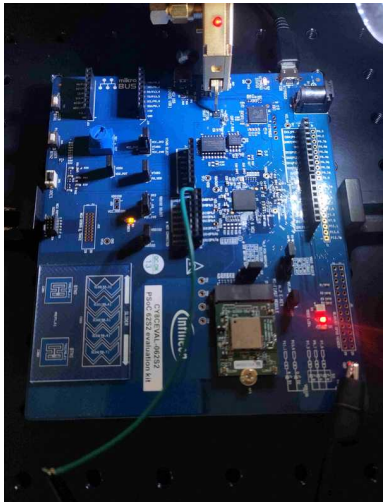
Infineon Security Microcontrollers (IC CC Certifications)

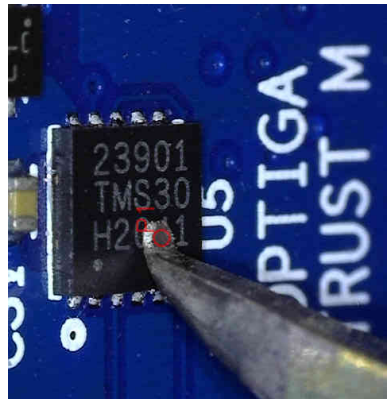


Legend: # IC (# Certification Reports, # Maintenance Reports)

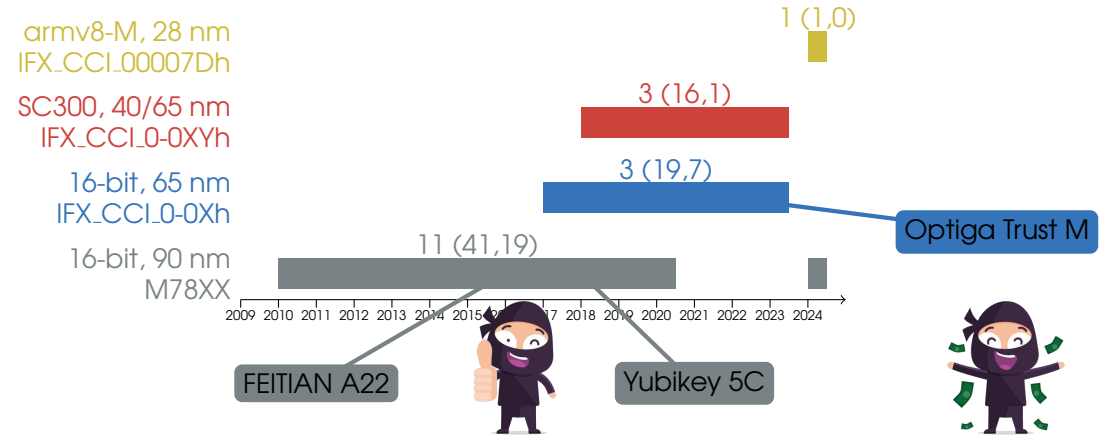
Credits: `www.bsi.bund.de`, `www.sec-certs.org`

Optiga Trust M – Side-channel Setup





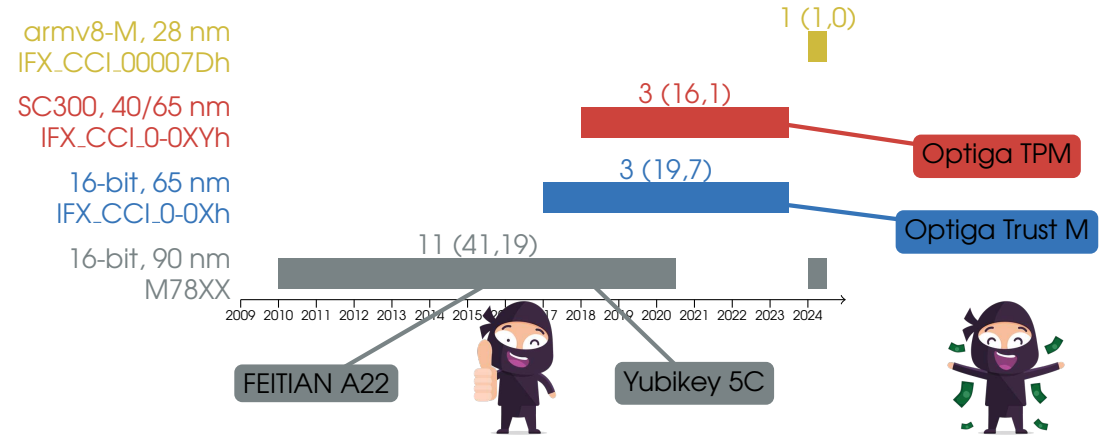
Infineon Security Microcontrollers (IC CC Certifications)



Legend: # IC (# Certification Reports, # Maintenance Reports)

Credits: `www.bsi.bund.de`, `www.sec-certs.org`

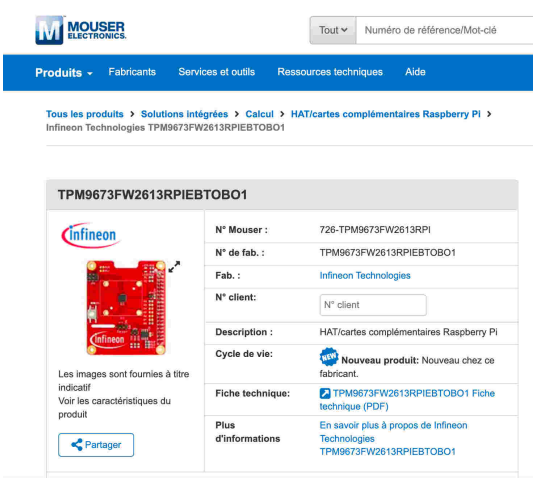
Infineon Security Microcontrollers (IC CC Certifications)



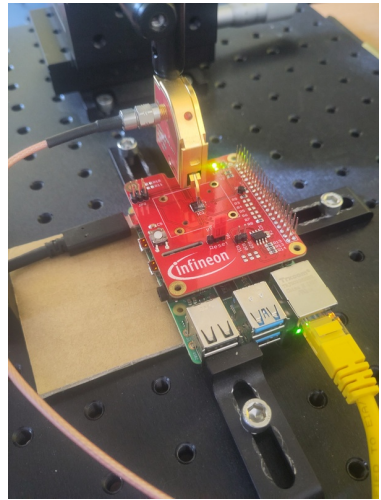
Legend: # IC (# Certification Reports, # Maintenance Reports)

Credits: www.bsi.bund.de, www.sec-certs.org

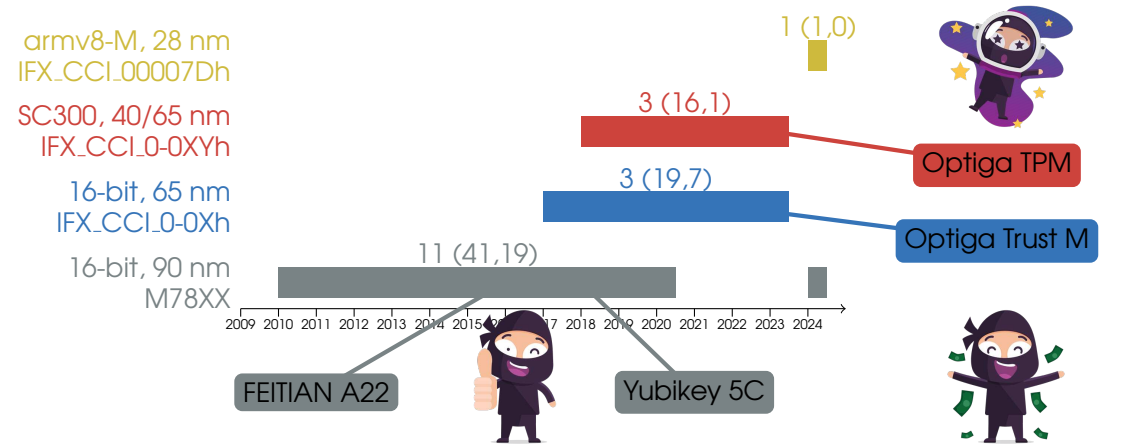
Optiga TPM – Evaluation Kit



https://github.com/Infineon/optiga-tpm



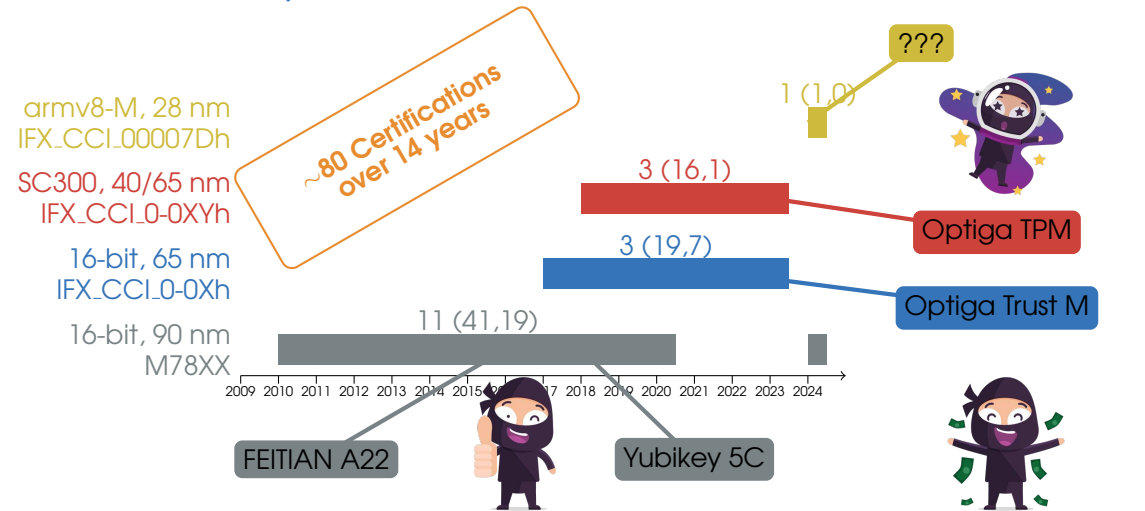
Infineon Security Microcontrollers (IC CC Certifications)



Legend: # IC (# Certification Reports, # Maintenance Reports)

Credits: `www.bsi.bund.de`, `www.sec-certs.org`

Infineon Security Microcontrollers (IC CC Certifications)



Legend: # IC (# Certification Reports, # Maintenance Reports)

Credits: `www.bsi.bund.de`, `www.sec-certs.org`

Agenda

Introduction
- FIDO
- Infineon
- Yubikey

A Side-Channel
- Infineon ECDSA
- The Extended
- Summary

Impact Analysis
- Infineon Security Microcontrollers
- Yubikey 5Ci
- Optiga Trust M
- Optiga TPM

Conclusions
- Summing up
- Mitigations
- Avenues Of Research
- Project Timeline

Let's sum up: Attack Requirements

- *Infineon security microcontroller* with *Infineon cryptolib*
- modular inversion of a secret (*eg.* ECDSA).
- The attacker must have physical access to the device:
  - open the device to access to the Infineon chip package;
  - EM probe + oscillo to capture the EM side-channel signal (few minutes).
- Later, the offline phase will take one hour to one day to retrieve the private key.

EUCLEAK

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

- Sovereign Documents
- Access Control
- Bank Cards

- Bitcoin HW Wallets
- 2FA HW Tokens

- SmartPhones
- Computers (TPMs)

- Smart Cars
- Smart Homes

...

φ Attacker

Side-Channel
Fault Injection
Invasive

Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs

EUCLEAK

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

- Sovereign Documents
- Access Control
- Bank Cards

- Bitcoin HW Wallets
- 2FA HW Tokens

- SmartPhones
- Computers (TPMs)

- Smart Cars
- Smart Homes

...

φ Attacker

Side-Channel
Fault Injection
Invasive

Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs

EUCLEAK

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

- Sovereign Documents
- Access Control
- Bank Cards

- Bitcoin HW Wallets
- 2FA HW Tokens

- SmartPhones
- Computers (TPMs)

- Smart Cars
- Smart Homes

...

≥ 14 years

φ Attacker

Side-Channel
Fault Injection
Invasive

Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs

Mitigations

At Infineon Level:

- Increase the size of the multiplicative mask to Elliptic Curve size
- Use a *constant time* algorithm for inversion

eg. BEEA or ModExp

At Application Level:

- Avoid ECDSA

eg. EdDSA or RSA

- Defense in Depth

eg. Activate PIN (or any biometrics) on the device

- Protocol Specific Mitigations

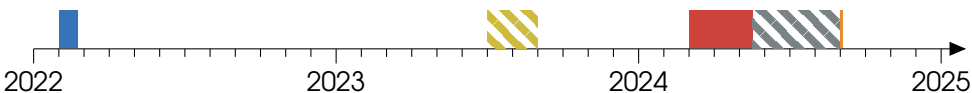
eg. Activate Counter in FIDO

Avenues Of Research

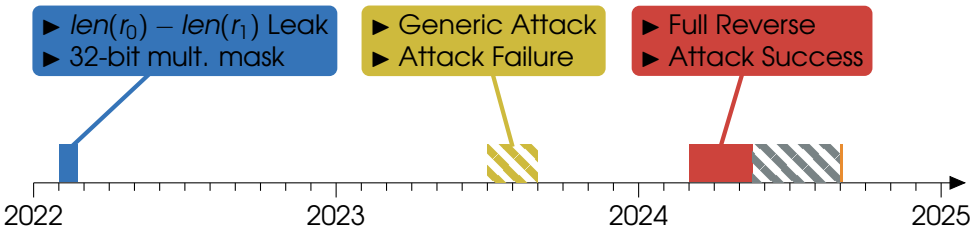
- Extend this work to RSA key generation
- Theoretical Analysis of the **EUCLEAK** generic attack
- Improve the attack in practice

  eg. single-trace attack or improve EM acquisitions
- Extend the generic attack to the Binary EEA case.

Project Timeline

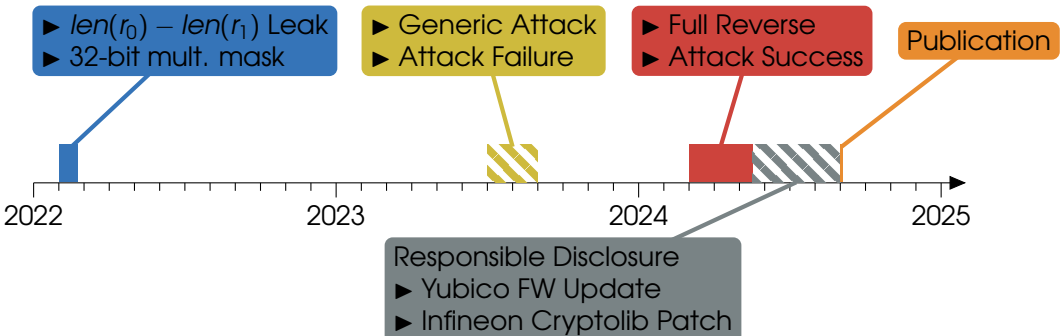


Project Timeline



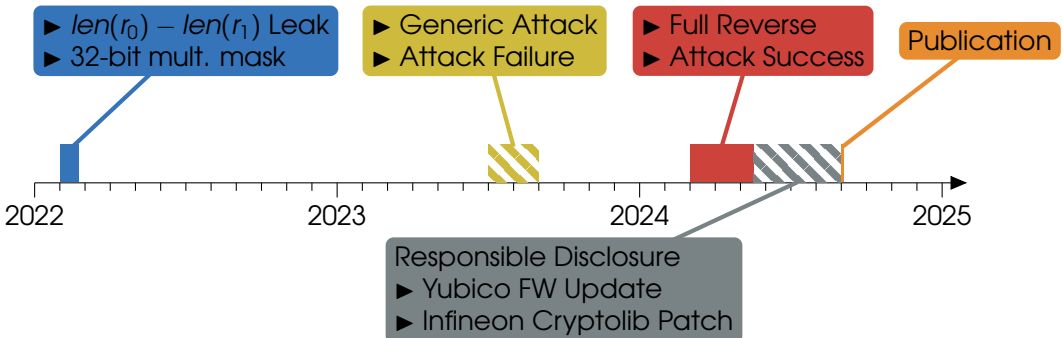
Project Timeline

ninjalab.io/eucleak
eprint.iacr.org/2024/1380
to appear in IEEE S&P 2025



Project Timeline

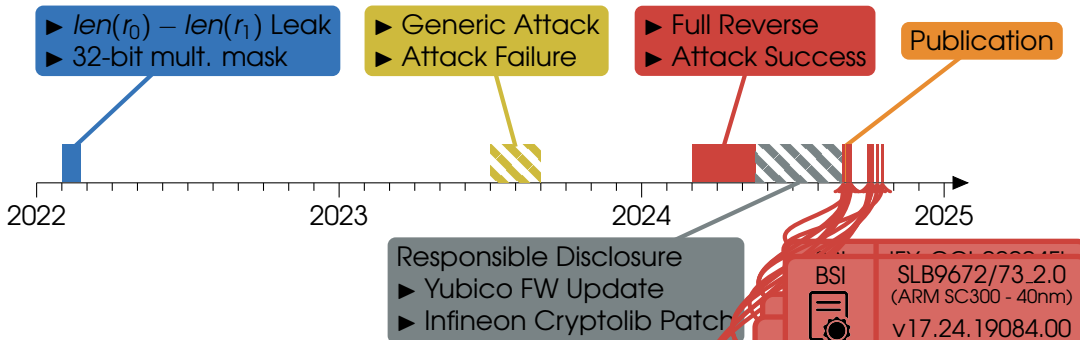
ninjalab.io/eucleak
eprint.iacr.org/2024/1380
to appear in IEEE S&P 2025



- Sept. 3rd 2024: Yubico Releases a Security Advisory
- Sept. 9th 2024: Public Post from BSI (in German)
- **No Public Statement from Infineon**
- **New CC Certificates**

Project Timeline

ninjalab.io/eucleak
eprint.iacr.org/2024/1380
to appear in IEEE S&P 2025



- Sept. 3rd 2024: Yubico Releases a Security Advisory
- Sept. 9th 2024: Public Post from BSI (in German)
- **No Public Statement from Infineon**
- **New CC Certificates**

Conclusions

Summing Up

Summing Up

A SIDE JOURNEY TO TITAN (Usenix'21)

- NXP Secure Element – First generation
- Unprotected nonce during the (constant time) scalar multiplication

EUCLEAK (ieee S&P'25)

- Infineon Secure Element – All recent generations
- Nonce protected with a 32-bit mask during the scalar modular inversion

Summing Up

A SIDE JOURNEY TO TITAN (Usenix'21)

- NXP Secure Element – First generation
- Unprotected nonce during the (constant time) scalar multiplication

EUCLEAK (ieee S&P'25)

- Infineon Secure Element – All recent generations
- Nonce protected with a 32-bit mask during the scalar modular inversion

TPM-FAIL (Usenix'20 by *Daniel Moghimi, Berk Sunar, Thomas Eisenbarth and Nadia Heninger*)

- Intel and STMicroelectronics TPMs – All generations
- Unprotected nonce during the scalar multiplication (not constant time)

Summing Up

A SIDE JOURNEY TO TITAN (Usenix'21)

- NXP Secure Element – First generation
- Unprotected nonce during the (constant time) scalar multiplication

EUCLEAK (ieee S&P'25)

- Infineon Secure Element – All recent generations
- Nonce protected with a 32-bit mask during the scalar modular inversion

TPM-FAIL (Usenix'20 by *Daniel Moghimi, Berk Sunar, Thomas Eisenbarth and Nadia Heninger*)

- Intel and STMicroelectronics TPMs – All generations
- Unprotected nonce during the scalar multiplication (not constant time)

MINERVA (TCHES'20 by *Jan Jancar, Vladimir Sedlacek, Petr Svenda and Marek Sys*)

- Inside Secure AT90SC
- Unprotected nonce during the scalar multiplication (not constant time)

Root Causes

- ECDSA ?

Root Causes

- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.

Root Causes

- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.
  - ↪ Sound and efficient countermeasures exist!

Root Causes

- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.
  - ↪ Sound and efficient countermeasures exist!
- Closed Sources ?

Root Causes

- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.
  - ↪ Sound and efficient countermeasures exist!
- Closed Sources ?
  - ↪ Lack of thorough scrutiny.
  - ↪ Developers rely on the implementation secrecy.

Root Causes

- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.
  - ↪ Sound and efficient countermeasures exist!
- Closed Sources ?
  - ↪ Lack of thorough scrutiny.
  - ↪ Developers rely on the implementation secrecy.
  - ↪ White-box Common Criteria certification process should catch these flaws.

Root Causes

- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.
  - ↪ Sound and efficient countermeasures exist!
- Closed Sources ?
  - ↪ Lack of thorough scrutiny.
  - ↪ Developers rely on the implementation secrecy.
  - ↪ White-box Common Criteria certification process should catch these flaws.
- Flawed Certification Process ?

Root Causes

- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.
  - ↪ Sound and efficient countermeasures exist!
- Closed Sources ?
  - ↪ Lack of thorough scrutiny.
  - ↪ Developers rely on the implementation secrecy.
  - ↪ White-box Common Criteria certification process should catch these flaws.
- Flawed Certification Process ?
  - ↪ Time constraints.
  - ↪ Evaluators' cognitive biases.

Root Causes

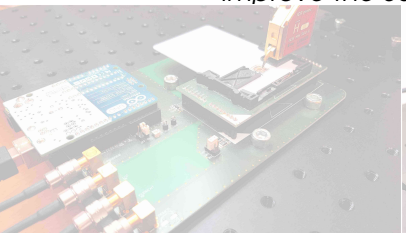
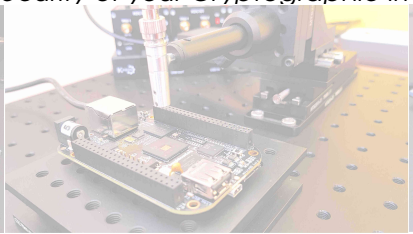
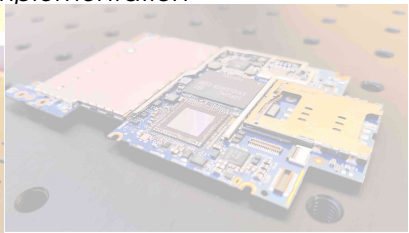
- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.
  - ↪ Sound and efficient countermeasures exist!
- Closed Sources ?
  - ↪ Lack of thorough scrutiny.
  - ↪ Developers rely on the implementation secrecy.
  - ↪ White-box Common Criteria certification process should catch these flaws.
- Flawed Certification Process ?
  - ↪ Time constraints.
  - ↪ Evaluators' cognitive biases.
  - ↪ CC is the most stringent security certification process.
  - ↪ Real exploit of a side-channel attack on certified products == None.

Root Causes

- ECDSA ?
  - ↪ certainly the most fragile crypto primitive *w.r.t.* SCA.
  - ↪ Sound and efficient countermeasures exist!
- Closed Sources ?
  - ↪ Lack of thorough scrutiny.
  - ↪ Developers rely on the implementation secrecy.
  - ↪ White-box Common Criteria certification process should catch these flaws.
- Flawed Certification Process ?
  - ↪ Time constraints.
  - ↪ Evaluators' cognitive biases.
  - ↪ CC is the most stringent security certification process.
  - ↪ Real exploit of a side-channel attack on certified products == None.
- Side-Channel Attacks ?

NinjaLab

Improve the Security of your Cryptographic Implementation

https://ninjalab.io
contact@ninjalab.io

NinjaLab
12 rue Boussairolles
34000 MONTPELLIER
FRANCE