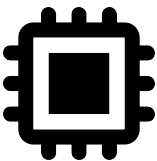
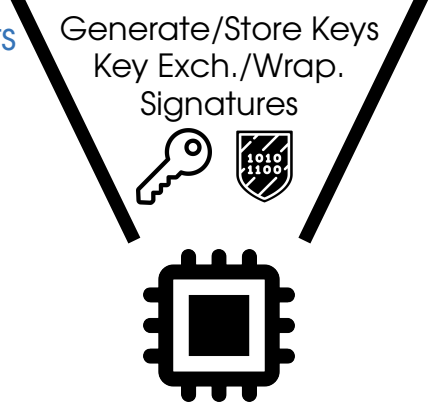
Thomas Roche
NinjaLab

C&ESAR
Rennes, FR – Nov. 20th, 2024

Secure Elements



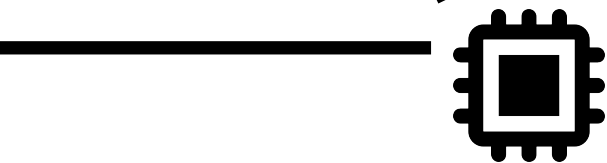
Secure Elements



Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

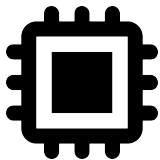
Remote Attacker



Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker



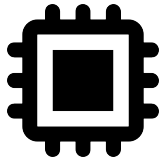
Simple HW
Simple SW
Simple I/O
Formal Methods

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

φ Attacker

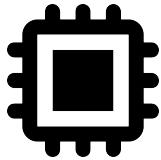


Simple HW
Simple SW
Simple I/O
Formal Methods

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker



φ Attacker

Simple HW
Simple SW
Simple I/O
Formal Methods

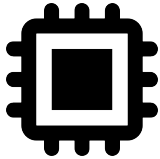
Side-Channel
Fault Injection
Invasive

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

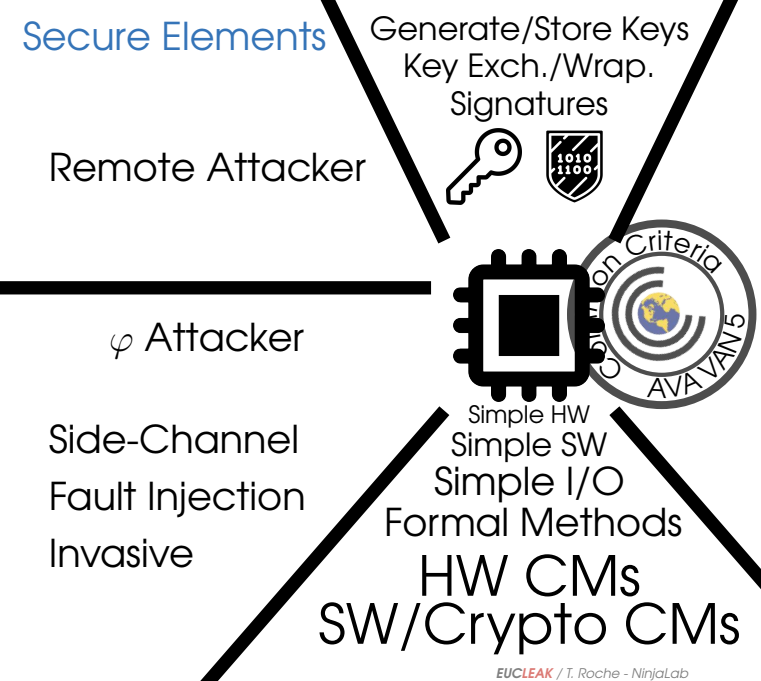
φ Attacker

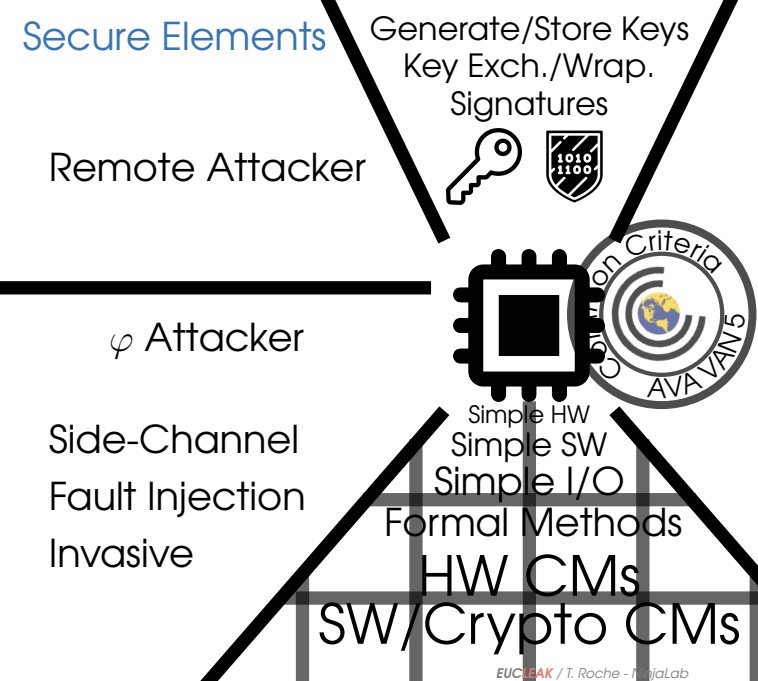


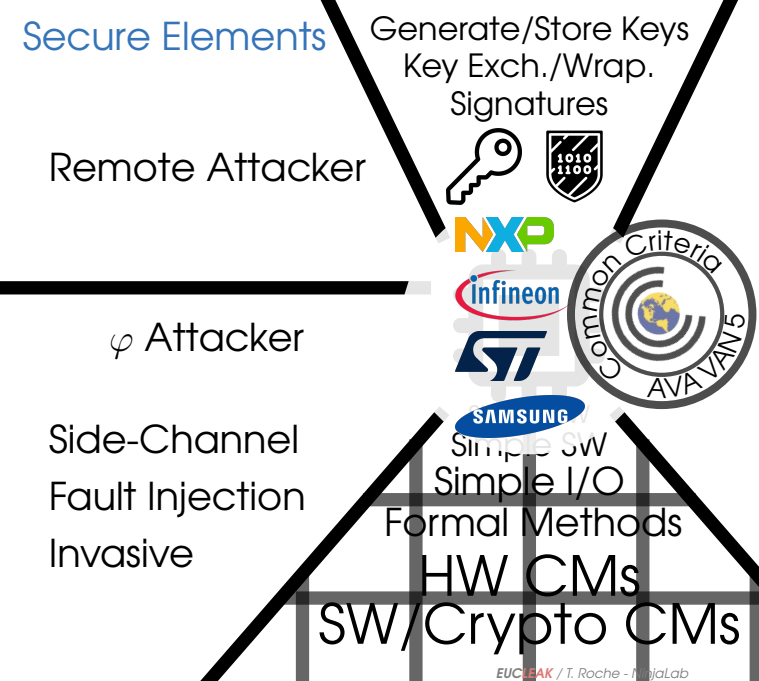
Simple HW
Simple SW
Simple I/O
Formal Methods

Side-Channel
Fault Injection
Invasive

HW CMs
SW/Crypto CMs


Secure Elements
Generate/Store Keys
Key Exch./Wrap.
Signatures
Remote Attacker
1010
1100
Common Criteria
AVA VAN 5
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
Side-Channel
Fault Injection
Invasive
Common Criteria
AVA VAN 5
Simple HW
Simple SW
Simple I/O
Formal Methods
HW CMs
SW/Crypto CMs


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

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

- Sovereign Documents
- Access Control
- Bank Cards

Remote Attacker

NXP

infineon

Common Criteria
AVA VAN 5

φ Attacker

SAMSUNG

Side-Channel
Fault Injection
Invasive

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

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

- Sovereign Documents
- Access Control
- Bank Cards

- Bitcoin HW Wallets
- 2FA HW Tokens

Remote Attacker

φ Attacker

Side-Channel
Fault Injection
Invasive

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

Secure Elements

Generate/Store Keys
Key Exch./Wrap.
Signatures

Remote Attacker

- Sovereign Documents
- Access Control
- Bank Cards

- Bitcoin HW Wallets
- 2FA HW Tokens

φ Attacker

- SmartPhones
- Computers (TPMs)

Side-Channel
Fault Injection
Invasive

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

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

- SmartPhones
- Computers (TPMs)

- Smart Cars
- Smart Homes

...

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

Agenda

Agenda

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

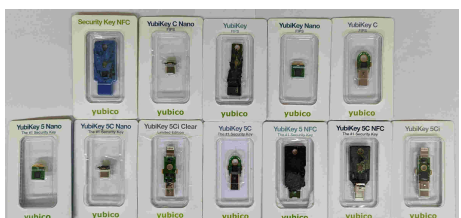
  ↪ Secure Element

A Side Journey To Titan

- In 2021 NinjaLab published *A Side Journey to Titan* (Usenix Security'21)
  SCA vulnerability in NXP'P5x security MCU ECC cryptolib.
  ↪Side-Channel Key-Recovery Attack on ECDSA
  ↪Clone FIDO token *Google Titan Security Key*
- NXP'P5x security microcontrollers are already old devices (last CC certification in 2015)
  ↪Most common security microcontrollers in FIDO Tokens are Infineon SLE78.

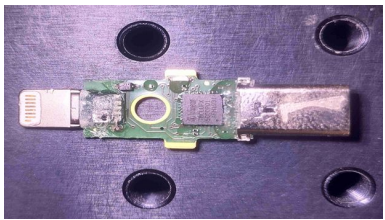
A Side Journey To Titan

- In 2021 NinjaLab published *A Side Journey to Titan* (Usenix Security'21)
  SCA vulnerability in NXP'P5x security MCU ECC cryptolib.
  ↪Side-Channel Key-Recovery Attack on ECDSA
  ↪Clone FIDO token *Google Titan Security Key*
- NXP'P5x security microcontrollers are already old devices (last CC certification in 2015)
  ↪Most common security microcontrollers in FIDO Tokens are Infineon SLE78.

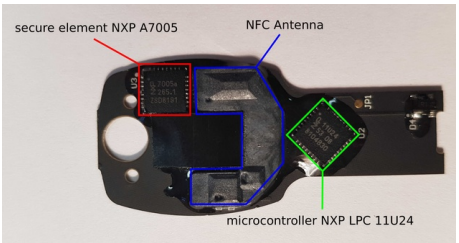


Infineon SLE 78

*The **SLE 78 USB** is a cache-based pure **16-bit security controller** family designed to meet all secure USB token design requirements. Its outstanding digital security concept Integrity Guard offers comprehensive error detection, a self-checking dual CPU and a fully encrypted data path including encrypted calculation in the CPU. 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/

FEITIAN A22 Open JavaCard

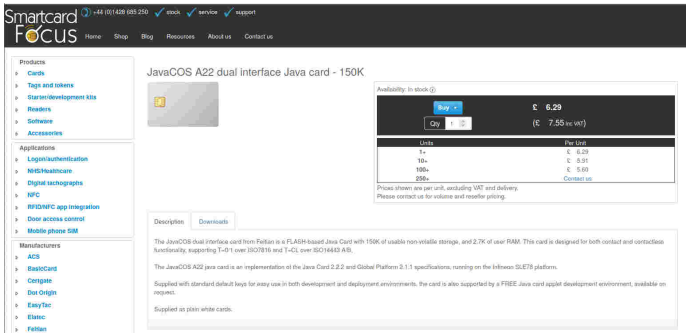
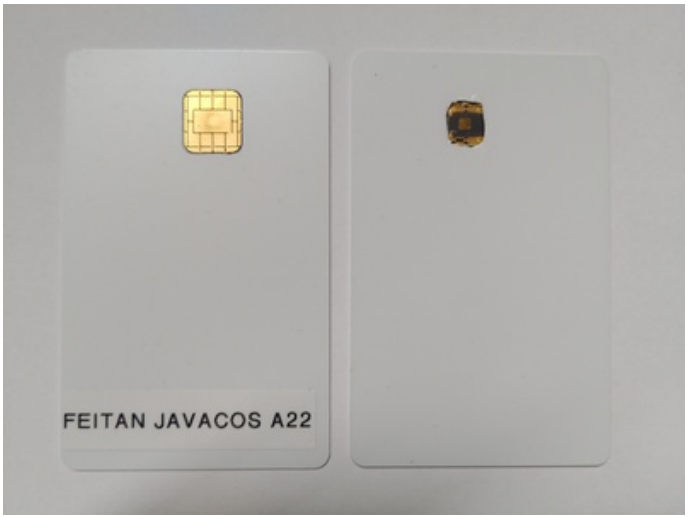


Figure: FEITIAN A22 – Screenshot from SmartCard Focus

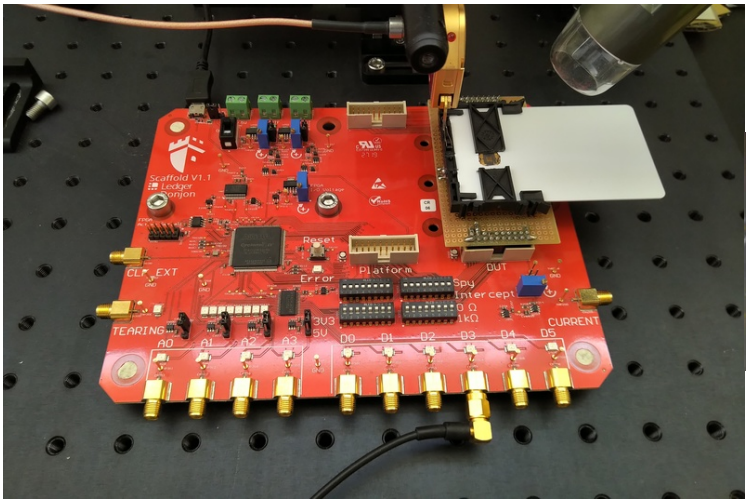
- Develop and push our own JavaCard applet
  ↪ECDSA Signature & Verification
- certified EAL5+ under Common Criteria in 2018[2]
- Infineon asymmetric crypto lib version 1.02.013

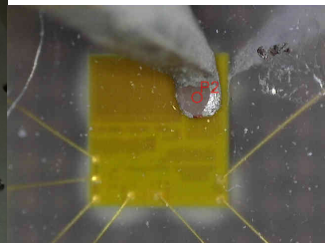
[2] https://www.commoncriteriaportal.org/files/epfiles/SERTIT-091CRFeitianv1.0.pdf

FEITIAN A22 Open JavaCard

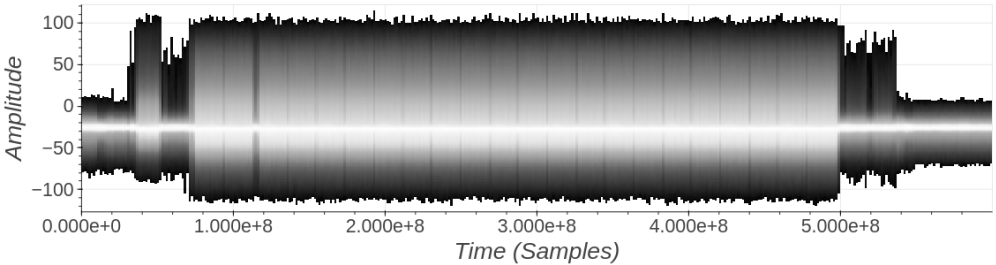


FEITIAN A22 – EM Acquisitions





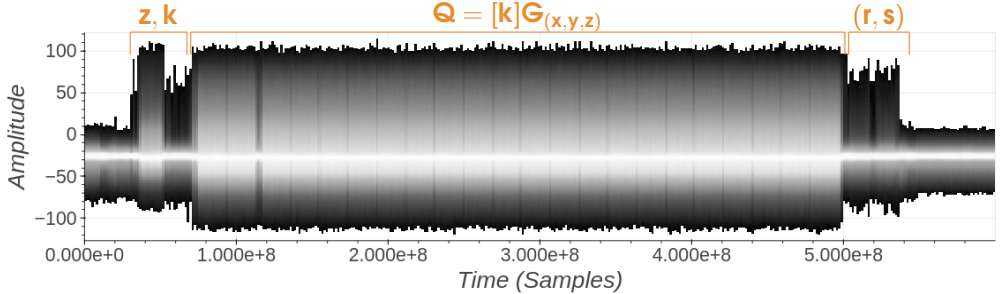
FEITIAN A22 – ECDSA Command – EM Radiations



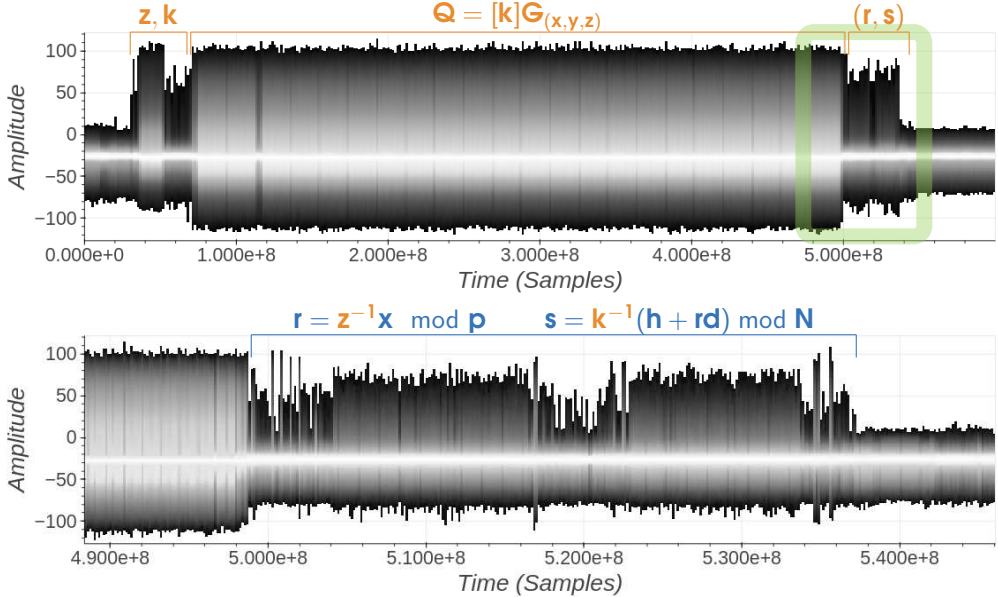
ECDSA Signature Scheme

- 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 $\mathbb{Z}/N\mathbb{Z}$
- randomly generate a random z in $\mathbb{Z}/p\mathbb{Z}$
- random projection $G_{(x,y)} \rightarrow G_{(xz,yz,z)}$
- compute $Q_{(x,y,z)} = [k]G_{(x,y,z)}$
- inv projection $Q_{(x,y,z)} \rightarrow Q_{(xz^{-1},yz^{-1})}$
- denote by r the x-coordinate of Q: $r = Q_x$
- compute $s = k^{-1}(h + rd) \bmod N$
- return (r, s)

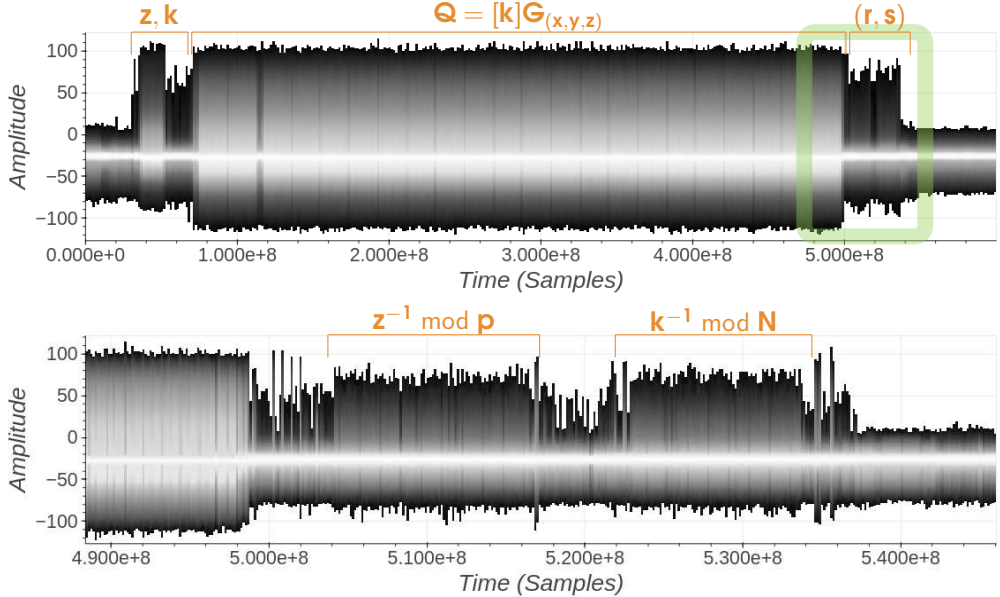
FEITIAN A22 – ECDSA Command – EM Radiations



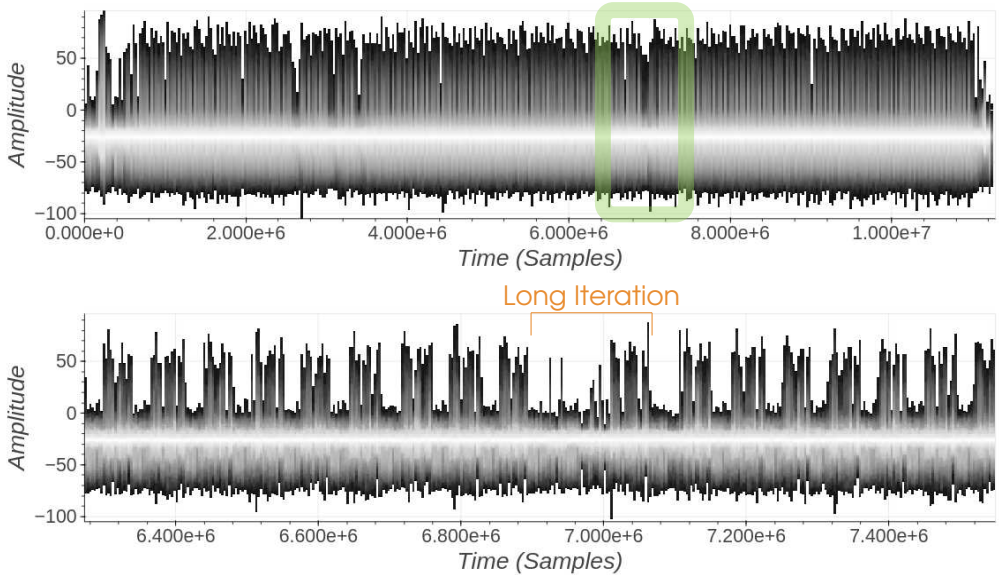
FEITIAN A22 – ECDSA Command – EM Radiations



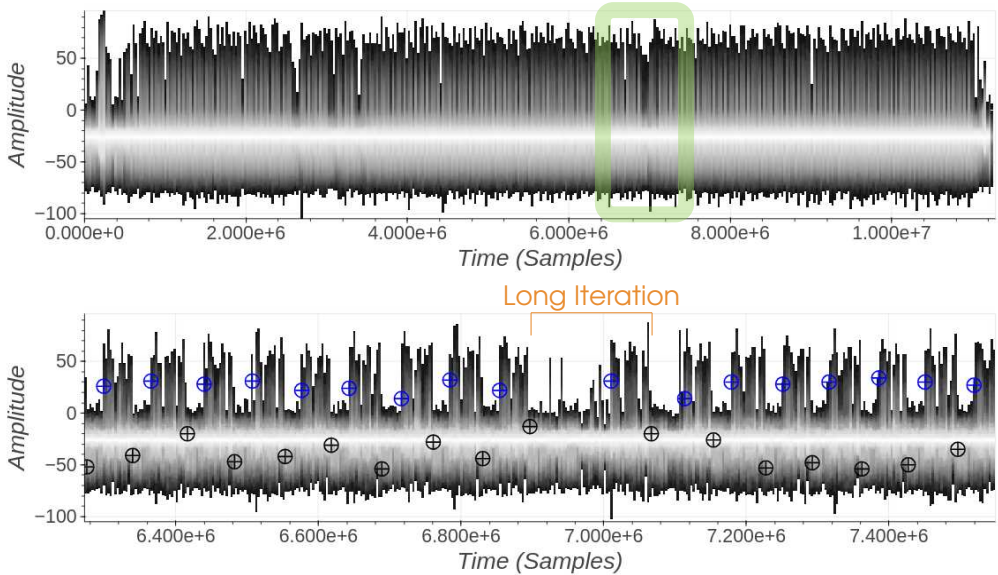
FEITIAN A22 – ECDSA Command – EM Radiations



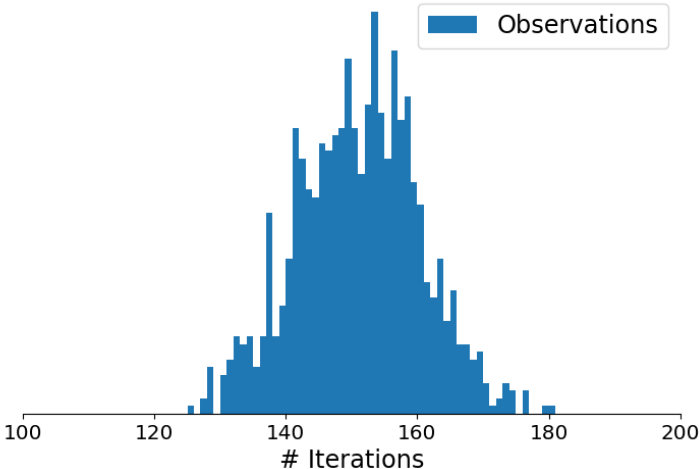
FEITIAN A22 – k^{-1} mod N – EM Radiations



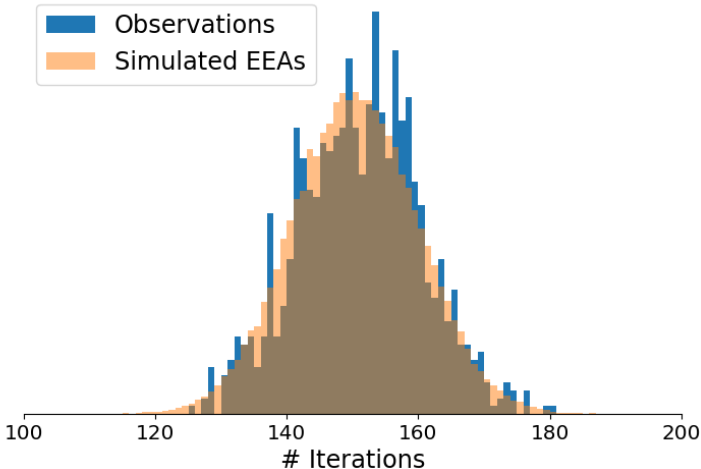
FEITIAN A22 – k^{-1} mod N – EM Radiations



FEITIAN A22 – k^{-1} mod N – Iterations Count Distribution



FEITIAN A22 – k^{-1} mod N – Iterations Count Distribution



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

 # Iterations does not match with $k^{-1} \bmod N$

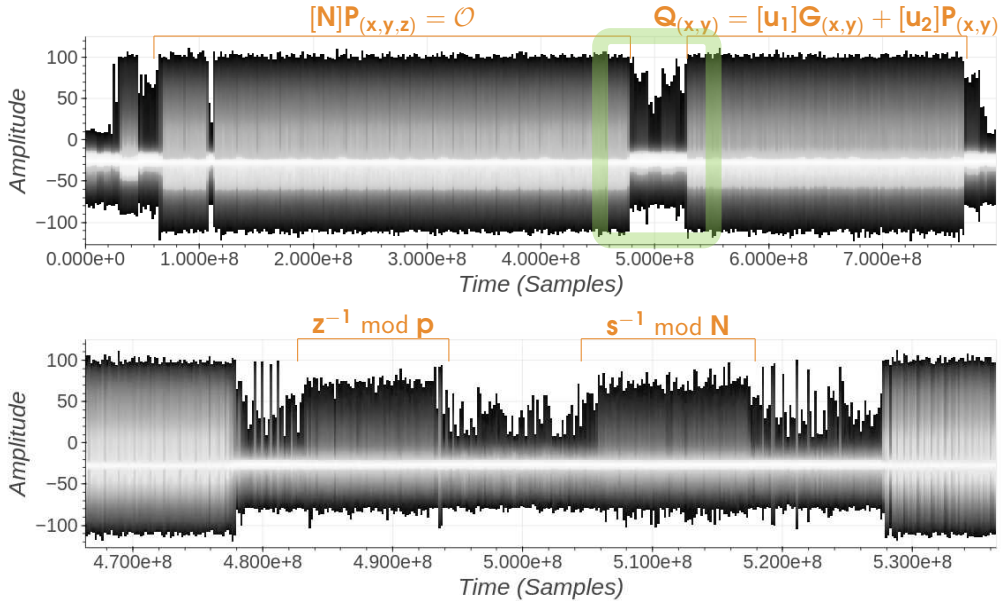
$\hookrightarrow$ k might be masked

Agenda

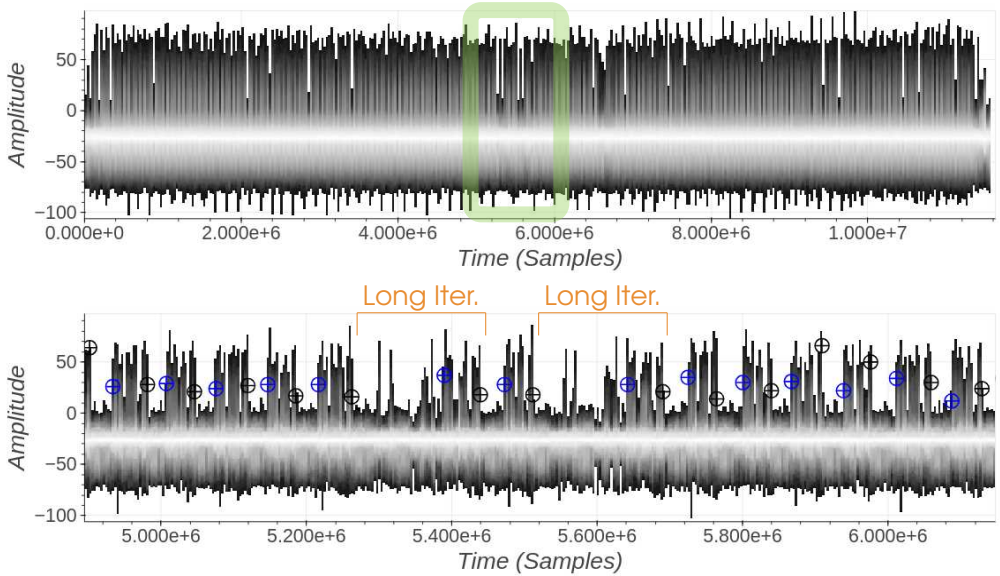
ECDSA Signature Verification Scheme

- Elliptic Curve base point is $G_{(x,y)}$, Elliptic Curve order is N
- Inputs: public key $P_{(x,y)}$, the input message $h = H(m)$, the signature (r, s)
- check that $P \neq \mathcal{O}$
- check that $P \in E$
- check that $[N]P = \mathcal{O}$
- Let $u_1 = hs^{-1} \bmod N$, $u_2 = rs^{-1} \bmod N$
- compute $Q_{(x,y)} = [u_1]G_{(x,y)} + [u_2]P_{(x,y)}$
- return True iff $r = Q_x \bmod N$

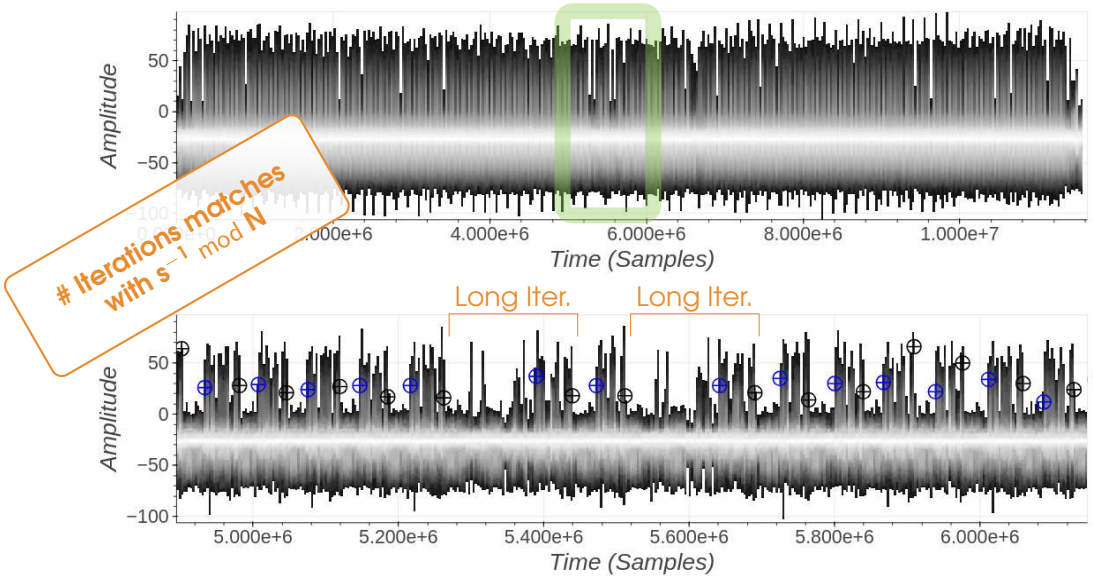
FEITIAN A22 – ECDSA Verif Command – EM Radiations



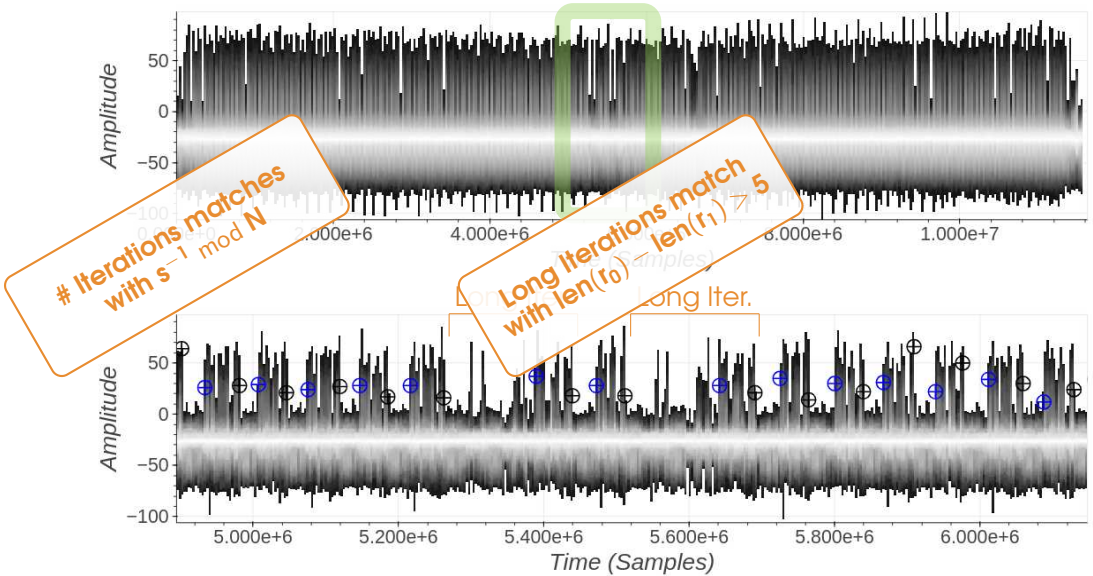
FEITIAN A22 – s^{-1} mod N – EM Radiations



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

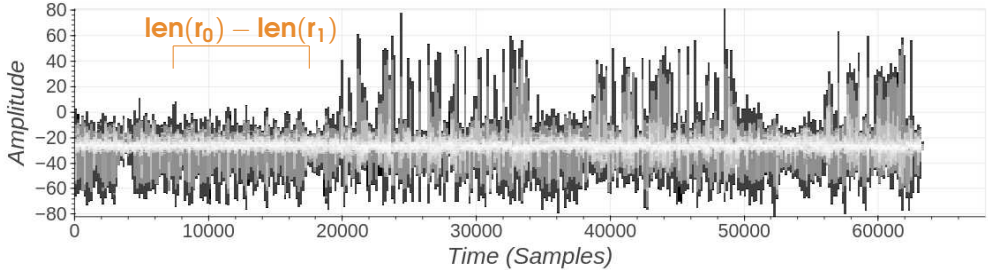


FEITIAN A22 – $s^{-1} \bmod N$ – EM Radiations

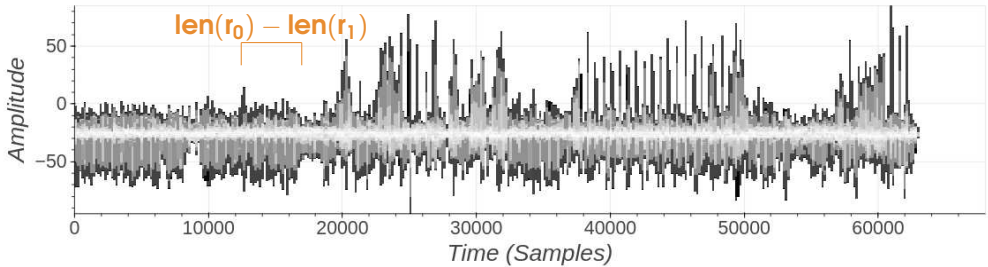


FEITIAN A22 – $s^{-1} \bmod N$ – Single Iteration

Odd



Even



Extended Euclidean Algorithm – Summary

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 – Summary

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

 # Iterations does match with $s^{-1} \bmod N$
Timing Leakage on $\mathrm{len}(r_0) - \mathrm{len}(r_1)$
Odd iterations $\neq$ Even iterations

A Masked Modular Inversion

$$
\begin{aligned}
k' &= k \times m \bmod N \\
k'^{-1} &= \texttt{EEA}(k', N) \\
k^{-1} &= k'^{-1} \times m \bmod N
\end{aligned}
$$

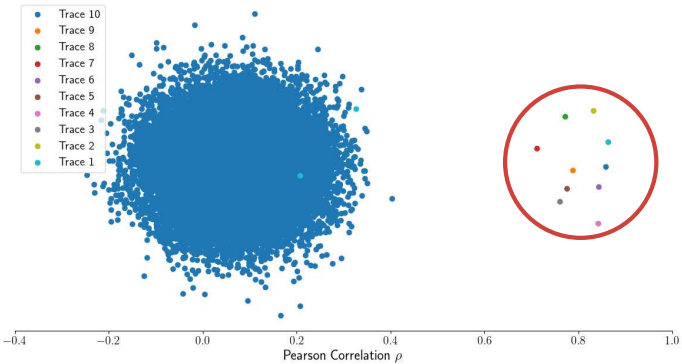
A Masked Modular Inversion

$$
\begin{aligned}
k' &= k \times m \bmod N \\
k'^{-1} &= \texttt{EEA}(k', N) \\
k^{-1} &= k'^{-1} \times m \bmod N
\end{aligned}
$$

- Hypothesis: the mask can be brute-forced
  (otherwise there is no reason to continue the investigation)
- Brute-force the mask:
  - For each value $\hat{m}$, compute $\hat{k'} = k \times \hat{m} \bmod N$
  - Predict the sequence of $\{\hat{\ell}_i = \mathit{len}(r_0) - \mathit{len}(r_1)\}_i$
  - compare $\{\hat{\ell}_i\}$ with $\{\ell_i\}_i$ (the observations)
  - Keep $\hat{m}$ if the sequences match well enough

A Masked Modular Inversion – Brute-Force Results

For the selected masks $\hat{m}$
Pearson Correlation between $\{\hat{\ell}_i = \mathit{len}(r_0) - \mathit{len}(r_1)\}_i$ and $\{\ell_i\}_i$



m is always a 32-bit odd integer!

Let's sum up

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

Let's sum up

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

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

Let's sum up

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

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

  Theoretically

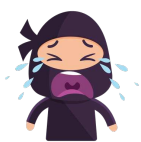
Let's sum up

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

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

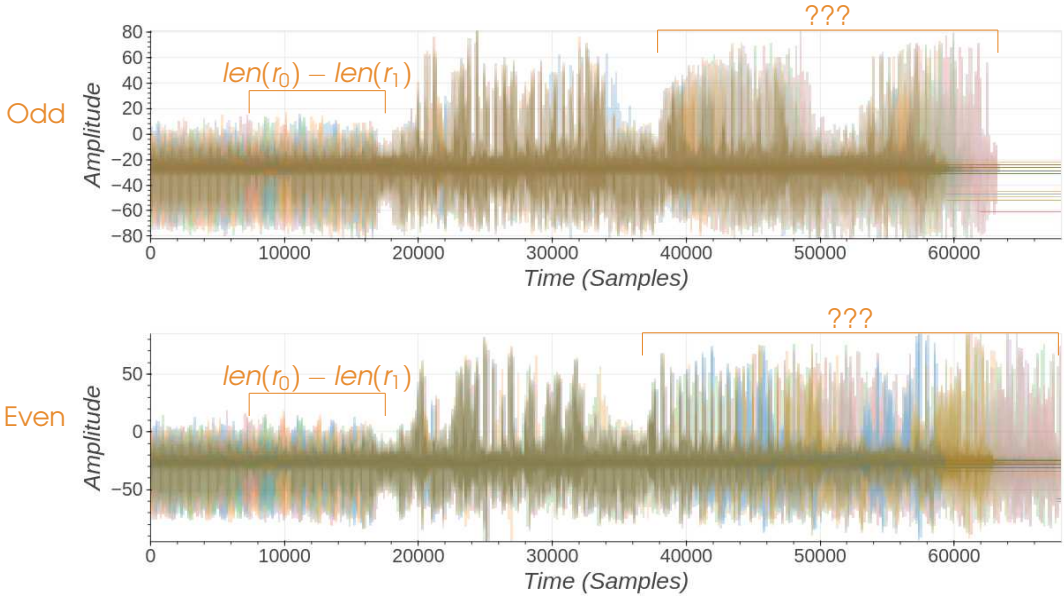
  Theoretically

No Key-Recovery Attack Was Found!



Agenda

FEITIAN A22 – $s^{-1} \bmod N$ – More Timing Leakages



A Weird Euclidean Division Algorithm

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

1 $r \leftarrow a$
2 $\ell \leftarrow len(r) - len(b)$
3 $q \leftarrow 0$
4 **while** $\ell \geq 0$ **do**
5 $\quad g \leftarrow \texttt{sign}(r).2^{\ell}$
6 $\quad r \leftarrow r - g.b$
7 $\quad q \leftarrow q + g$
8 $\quad \ell \leftarrow len(r) - len(b)$
9 **if** $r < 0$ **then**
10 $\quad q \leftarrow q - 1$ `// q is the quotient`
11 $\quad r \leftarrow r + b$ `// r is the remainder`

Return : q

Summary of The Sensitive Leakage

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

```
1  r ← a
2  ℓ ← len(r) − len(b)
3  q ← 0
4  while ℓ ≥ 0 do
5  |   g ← sign(r).2^ℓ
6  |   r ← r − g.b
7  |   q ← q + g
8  |   ℓ ← len(r) − len(b)
9  if r < 0 then
10 |   q ← q − 1
11 |   r ← r + b
   Return : q
```

$\ell = len(r_0) - len(r_1)$ leaks

Loop Iter. leaks

$\ell = 0$ leaks

Odd: $\texttt{sign}(r)$ leaks

Even: $\texttt{sign}(r)$ leaks iff $\ell > 0$

// q is the quotient
// r is the remainder

Let's sum up

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

Let's sum up

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

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

Let's sum up

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

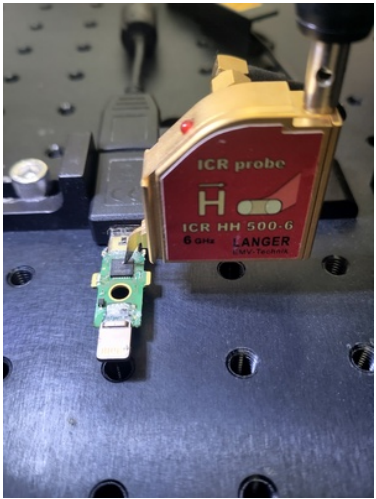
  blinded nonce $\rightarrow$ nonce $\rightarrow$ private key
- The timing leakage is enough information to guess the blinded nonce.

Side-Channel Attack
on Ext. Euclidean Alg.

ninjalab.io/eucleak

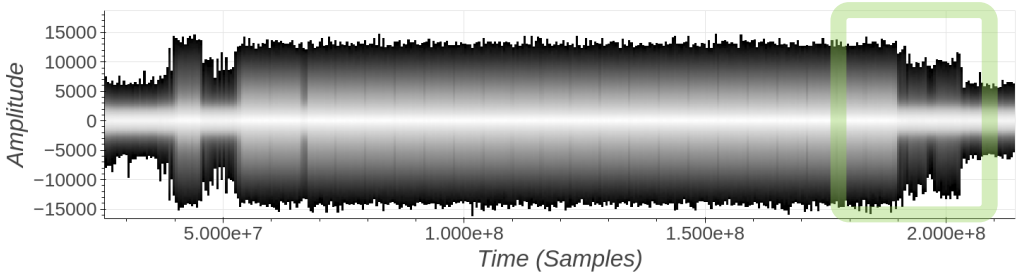
Agenda

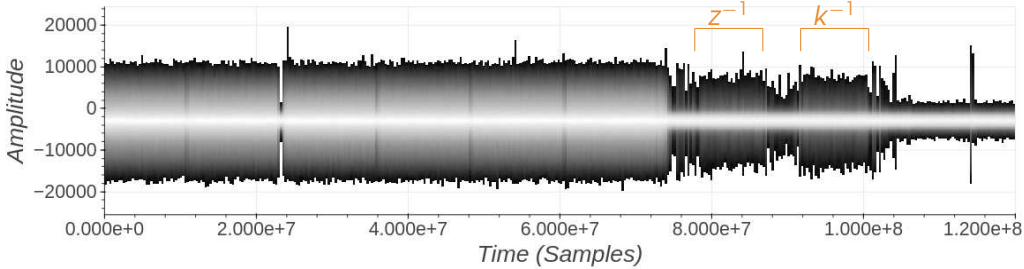
Yubikey 5Ci – EM Acquisitions



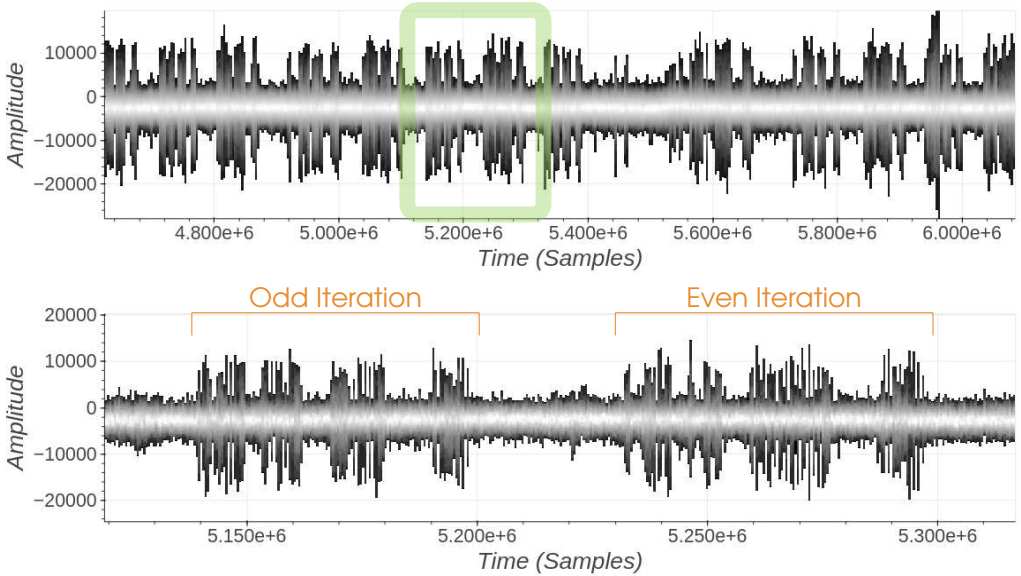
credits Yubico

Yubikey 5Ci – ECDSA Command – EM Radiations





Yubikey 5Ci – k^{-1} mod N – EM Radiations



Attack in Practice

- Secret key d is unknown
- Select EEA executions were $len(r_0) - len(r_1) \leq 5$ for the first half of the EEA
  $\hookrightarrow$ from 200 side-channel traces, 6 are selected
- From all iterations of the 6 side-channel traces, the leakage is extracted.
  semi-automated
- The attack is successful for 5 out of the 6 EEA traces.

Attack in Practice

- Secret key d is unknown
- Select EEA executions were $len(r_0) - len(r_1) \leq 5$ for the first half of the EEA
  $\hookrightarrow$ from 200 side-channel traces, 6 are selected
- From all iterations of the 6 side-channel traces, the leakage is extracted.
  semi-automated
- The attack is successful for 5 out of the 6 EEA traces.

- The pruning step can be avoided with more effort on the learning phase.
- The leakage extraction could be improved in both
  - Automation
  - Robustness

Agenda

Introdu
- FIDO H
- Infineon SL
- FEITIAN A
- Infineon
- The Extended Algorithm

A Side-Chann bility in EEA
- ECDSA Signa
- Infineon ECDSA verification
- First Observations
- Summary
- A Masked Modular Inversion

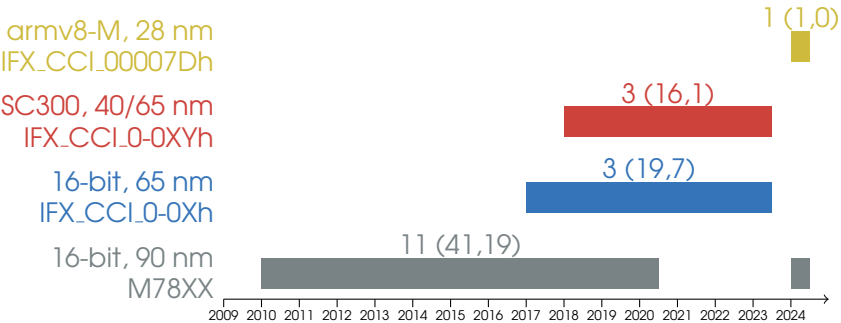
Full Reverse-Engineering of Infineon EEA
- Heuristical Approaches
- Summary of The Sensitive Leakage

Yubikey 5C
- Aquisition Setup
- First Side-Channel Traces
- Attack in Practice

Impact Analysis
- Infineon Security Microcontrollers
- Optiga Trust M
- Optiga TPM

Conclusions
- Conclusions
- Mitigations
- Project Timeline
- Project Timeline

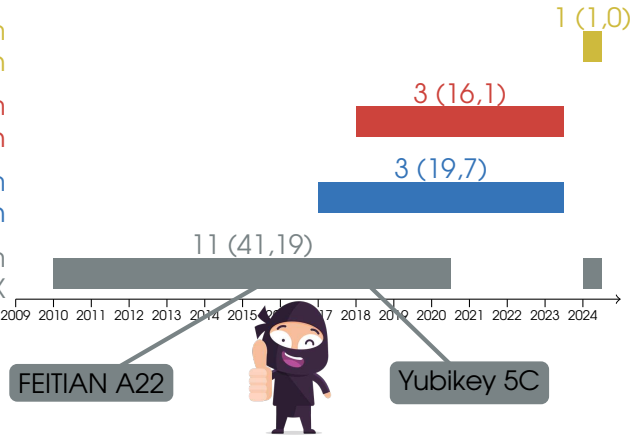
Infineon Security Microcontrollers (IC CC Certifications)



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

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

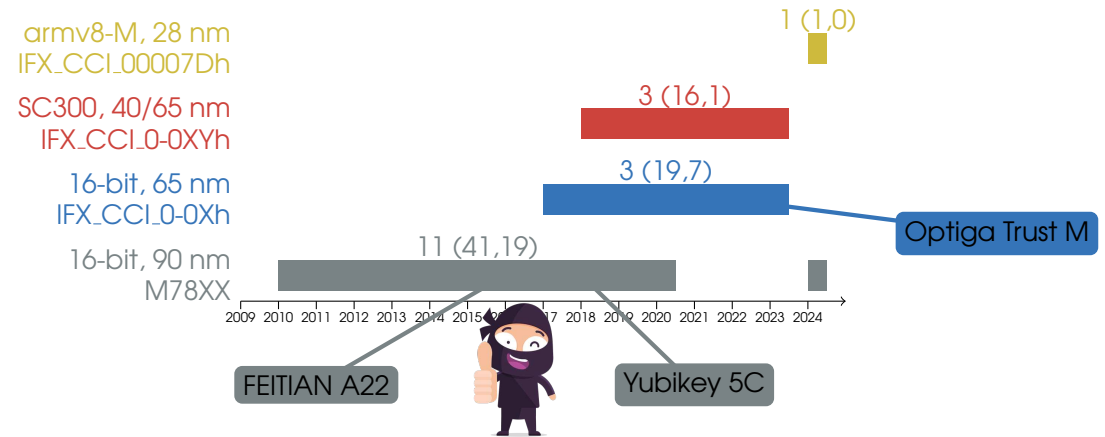
Infineon Security Microcontrollers (IC CC Certifications)



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

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

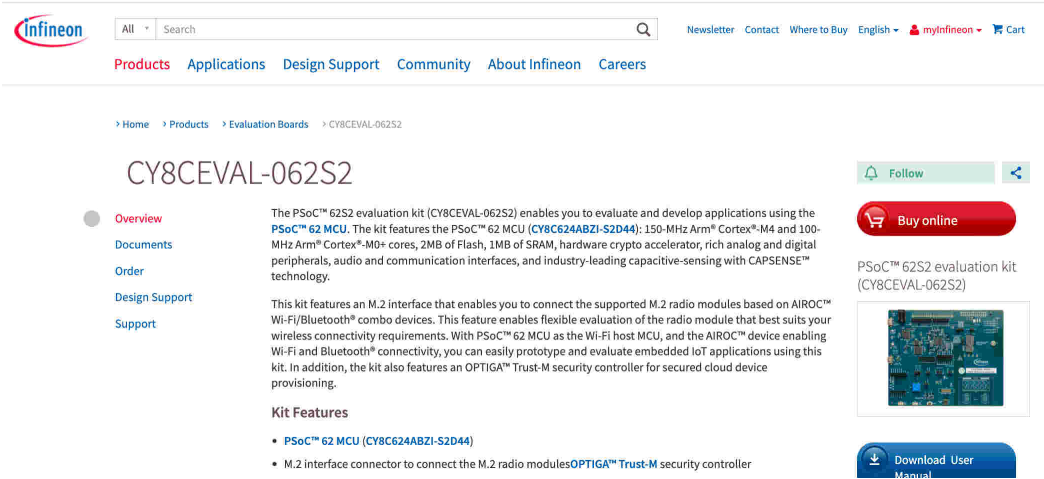
Infineon Security Microcontrollers (IC CC Certifications)



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

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

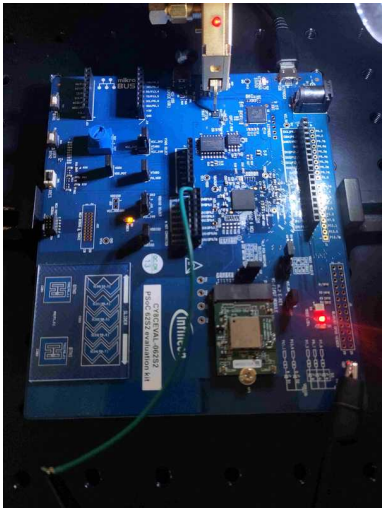
Optiga Trust M – Evaluation Kit



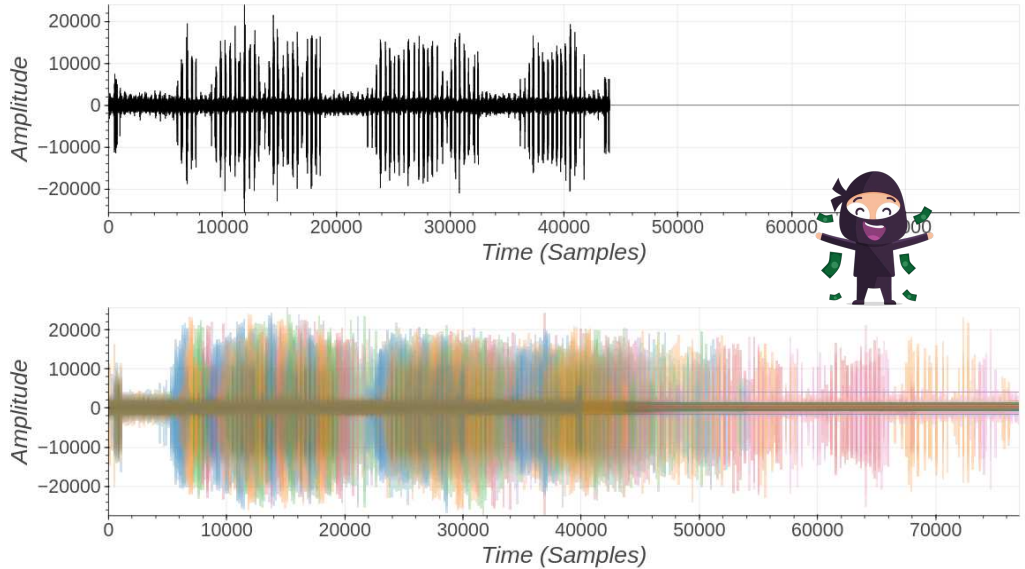
https://github.com/Infineon/optiga-trust-m

https://github.com/Infineon/mtb-example-optiga-crypto

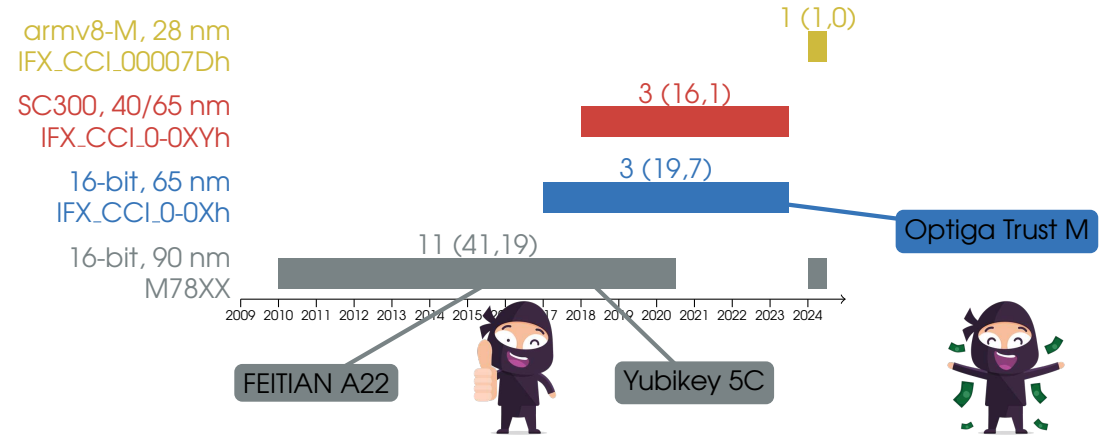
Optiga Trust M – Side-channel Setup



Optiga Trust M – $s^{-1} \bmod N$ – Single Iteration



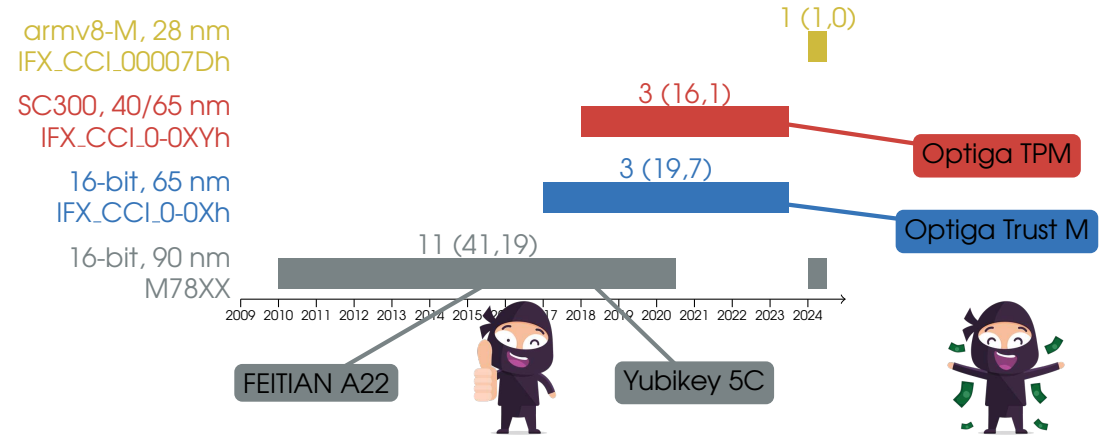
Infineon Security Microcontrollers (IC CC Certifications)



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

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

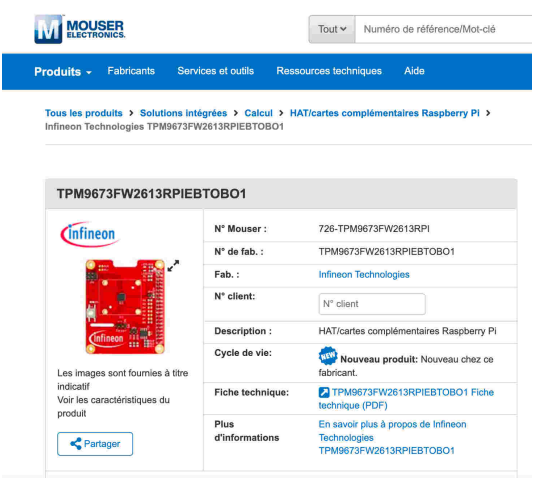
Infineon Security Microcontrollers (IC CC Certifications)



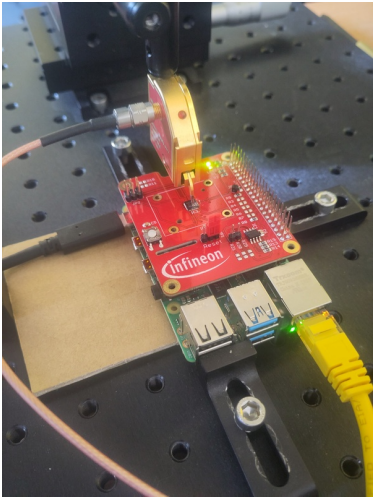
Legend: # IC (# Certification Reports, # Maintenance Reports)

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

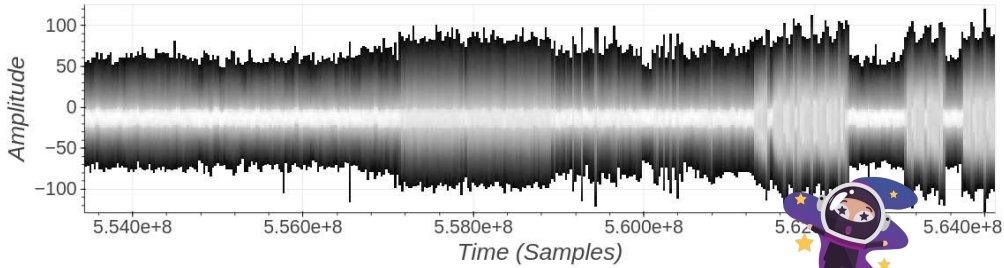
Optiga TPM – Evaluation Kit

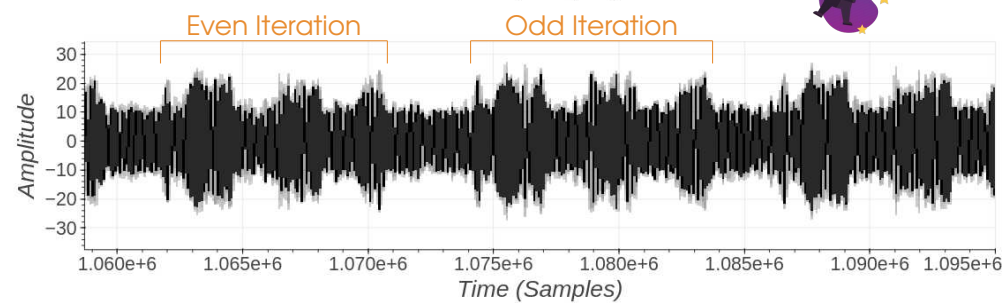


https://github.com/Infineon/optiga-tpm

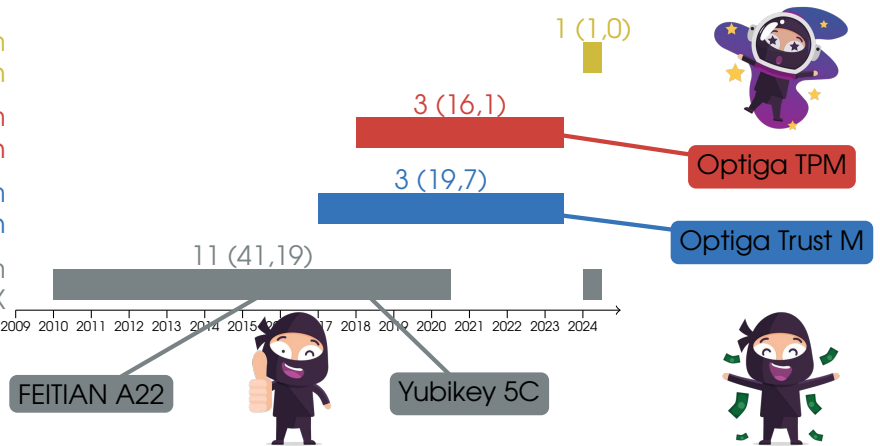


Optiga TPM – s^{-1} mod N – EM Radiations





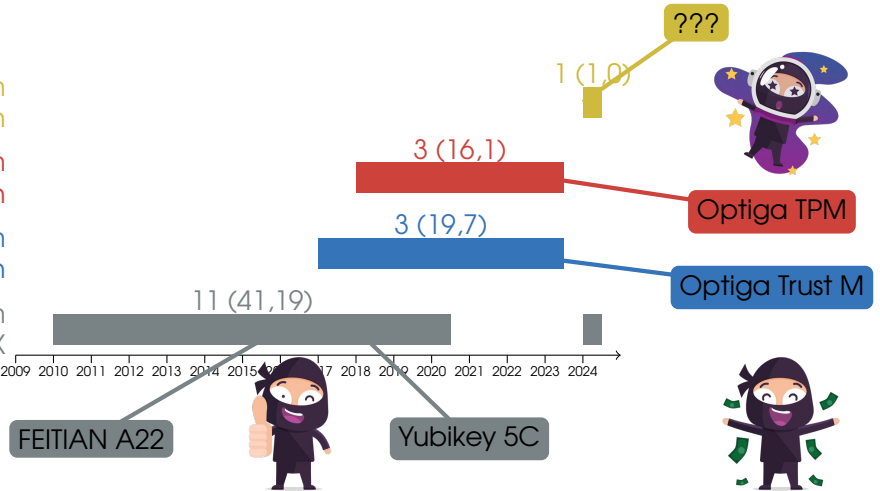
Infineon Security Microcontrollers (IC CC Certifications)



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

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

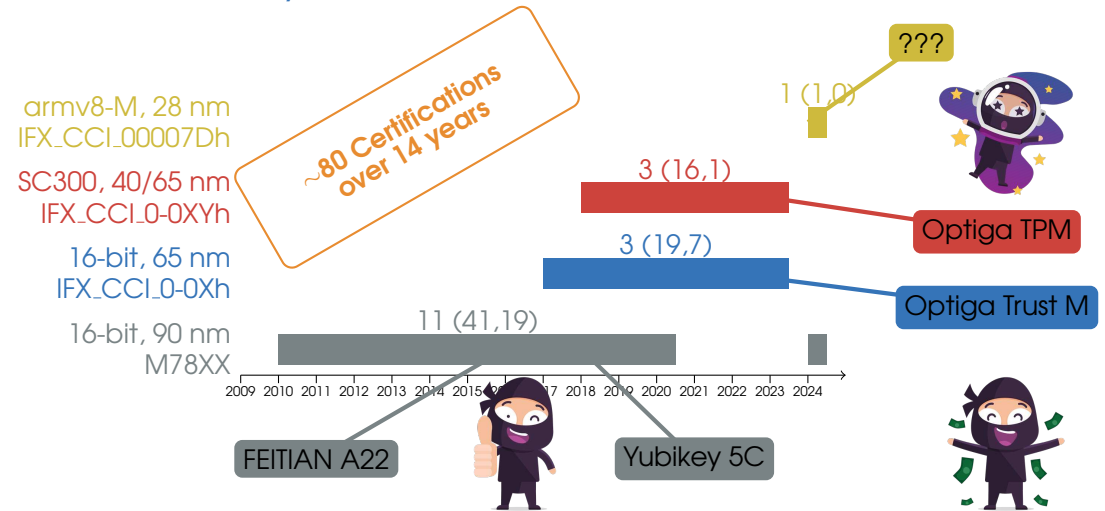
Infineon Security Microcontrollers (IC CC Certifications)



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

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

Infineon Security Microcontrollers (IC CC Certifications)



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

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

Agenda

Let's sum up: Attack Requirements

- *Infineon security microcontroller* with *Infineon cryptlib*
- 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

...

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

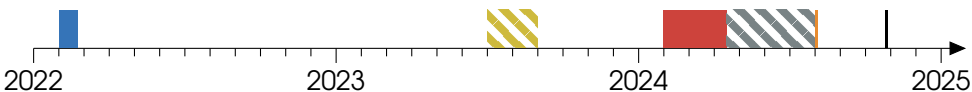
- Defense in Depth

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

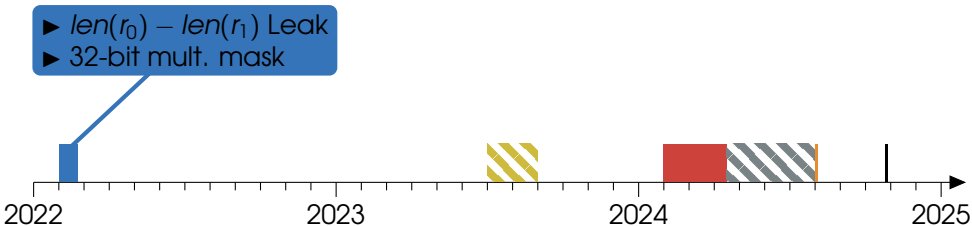
- Protocol Specific Mitigations

eg. Activate Counter in FIDO

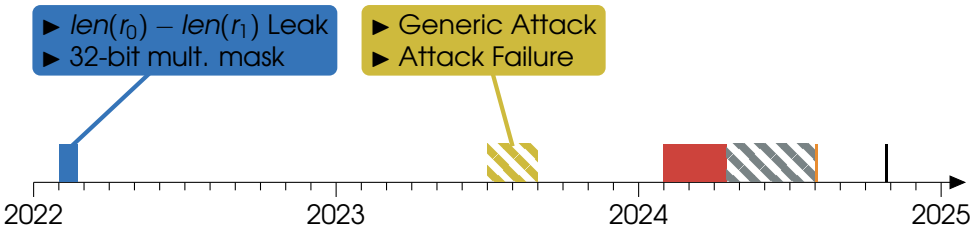
Project Timeline



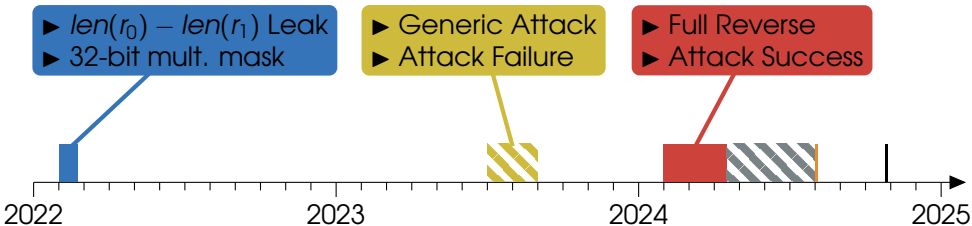
Project Timeline



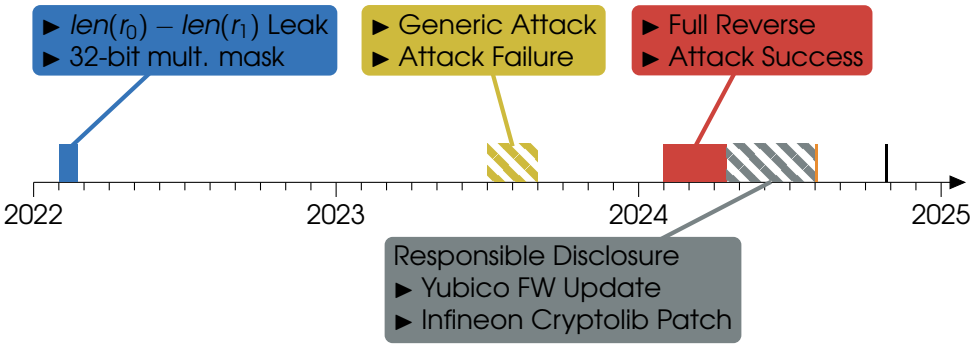
Project Timeline



Project Timeline

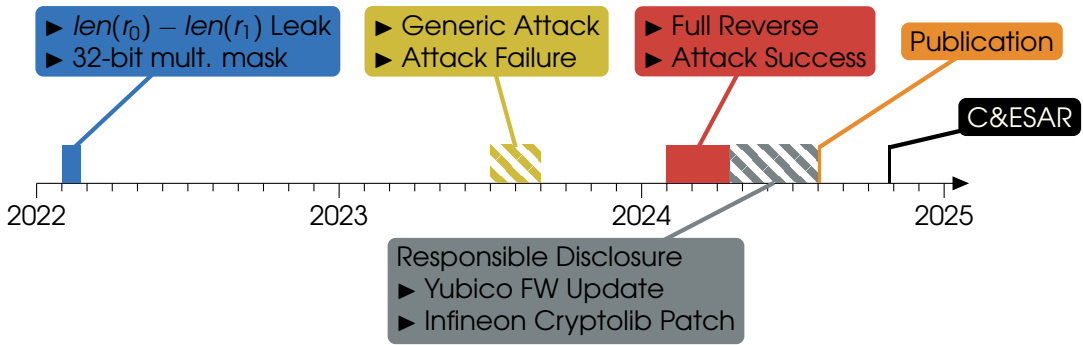


Project Timeline



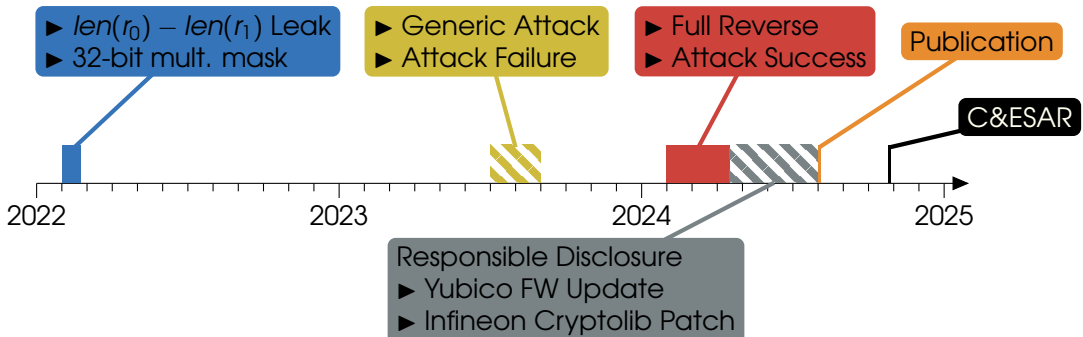
Project Timeline

ninjalab.io/eucleak
eprint.iacr.org/2024/1380



Project Timeline

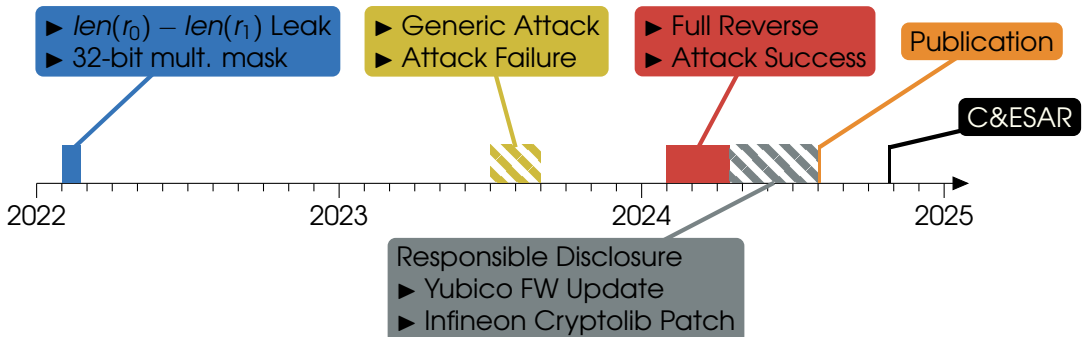
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



- Sept. 3rd 2024: Yubico Releases a Security Advisory

Project Timeline

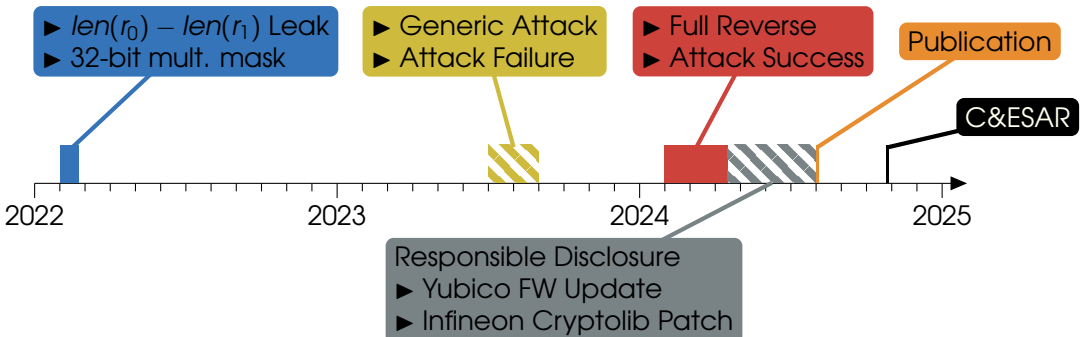
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



- Sept. 3rd 2024: Yubico Releases a Security Advisory
- Sept. 9th 2024: Public Post from BSI (in German)

Project Timeline

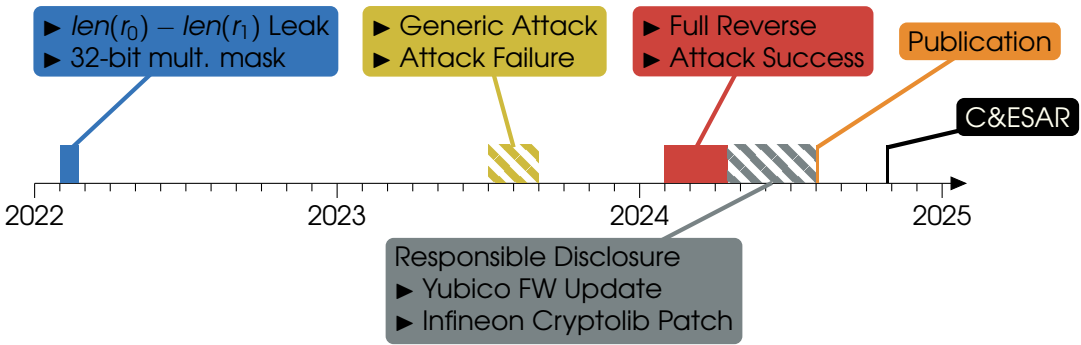
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



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

Project Timeline

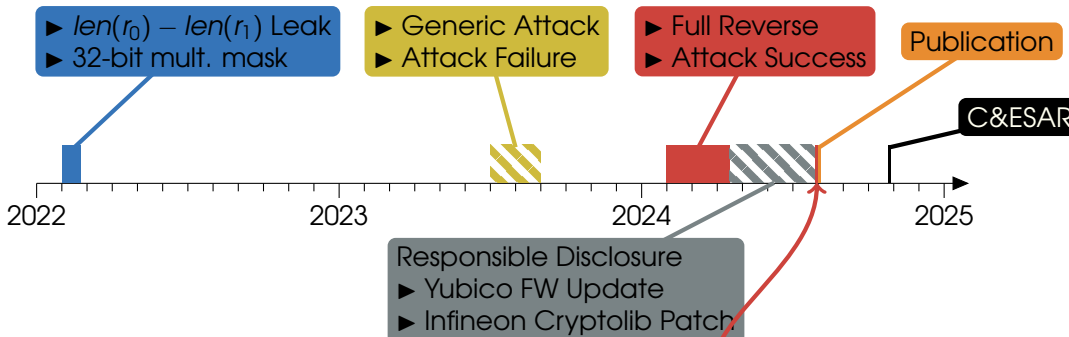
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



- 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

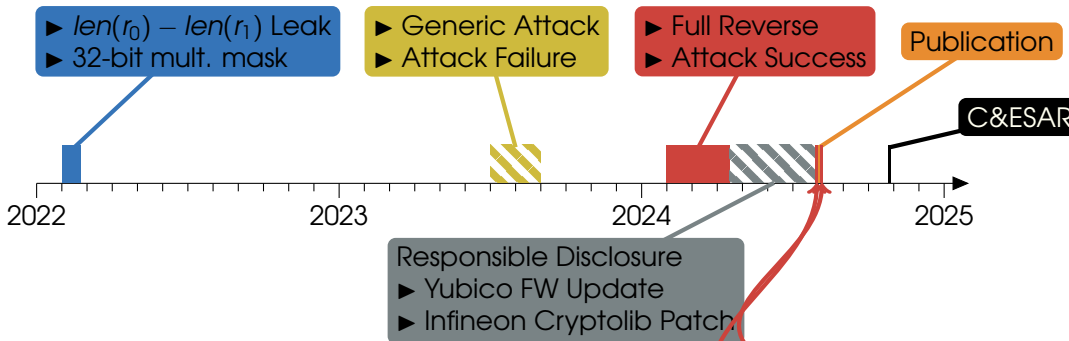


- Sept. 3rd 2024: Yubico Releases a Security Advisory
- Sept. 9th 2024: Public Post from BSI (in German)
- **No Public Statement from Infineon**
- **New CC Certificates**

NSCIB	IFX_CCI_0000XYh (ARM SC300 - 65nm)
AVA_VAN 5	Vulnerable ECC Library

Project Timeline

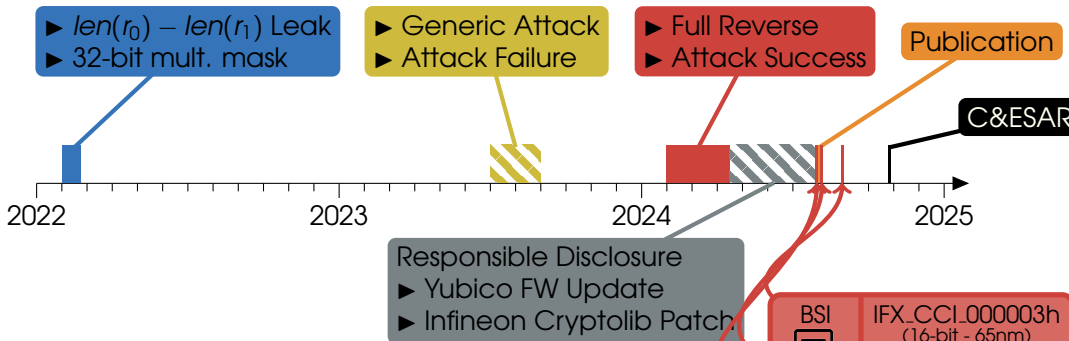
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



- 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

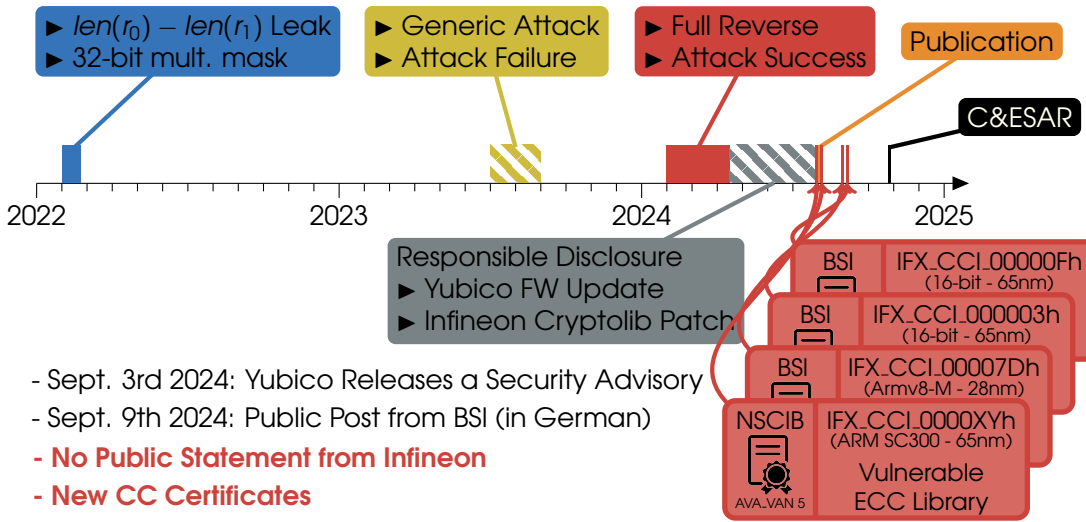
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



- 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

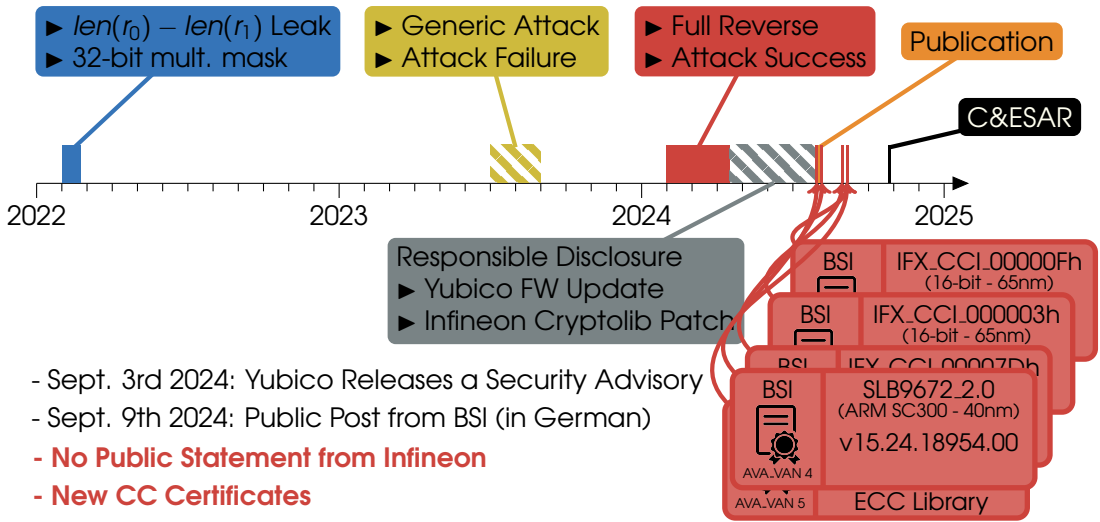
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



- 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

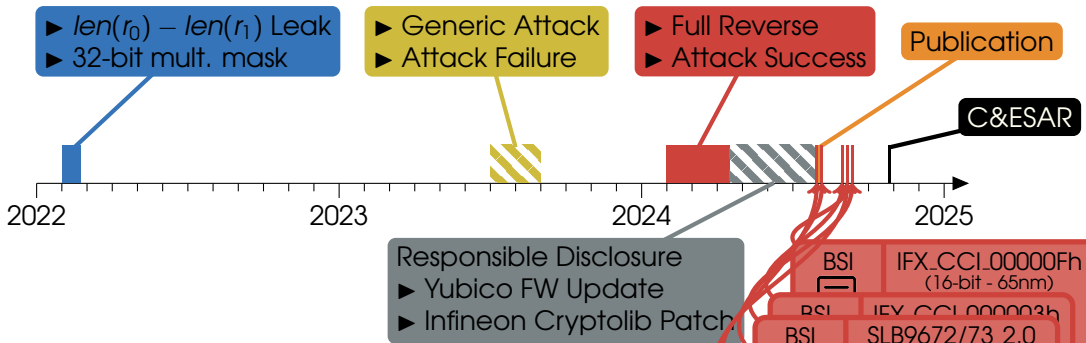
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



- 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

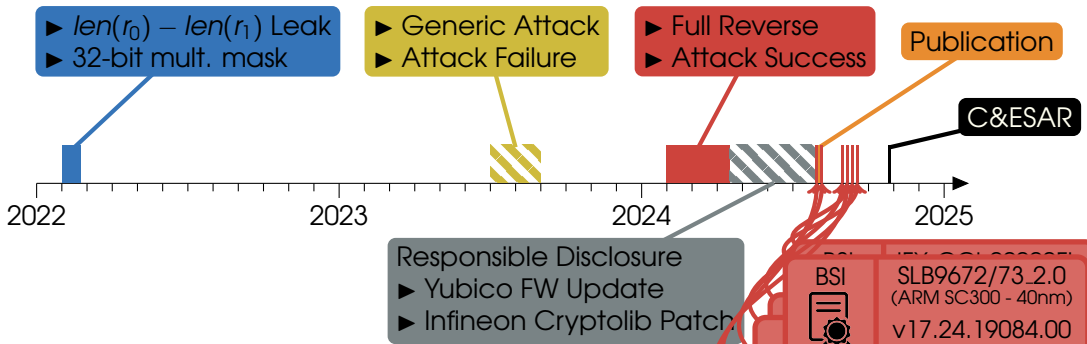
ninjalab.io/eucleak
eprint.iacr.org/2024/1380



- 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



- 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**

Infineon Security Microcontrollers – EC CryptoLibs – AFAWK

Family	Affected EC lib Versions	New EC lib versions
16-bit, 90 nm	1.1.18, 1.02.008, 1.02.013, 1.03.006, 2.03.008, 2.07.003	None
16-bit, 65 nm	2.06.003, 2.07.003, 2.08.007, 3.33.003	2.09.002
SC300, 40/65 nm	2.08.006, 3.03.003, 3.04.001	3.05.002
armv8-M, 28 nm	4.06.002	4.08.001
OptigaTPM	v15.20, v15.21, v15.22, v15.23, v16.10, v16.12, v26.10, v17.10, v17.12, v17.13, v27.10, v27.13	v15.24 v16.24, v26.24 v17.24, v27.24