TPM-Fail: TPM meets Timing and Lattice Attacks

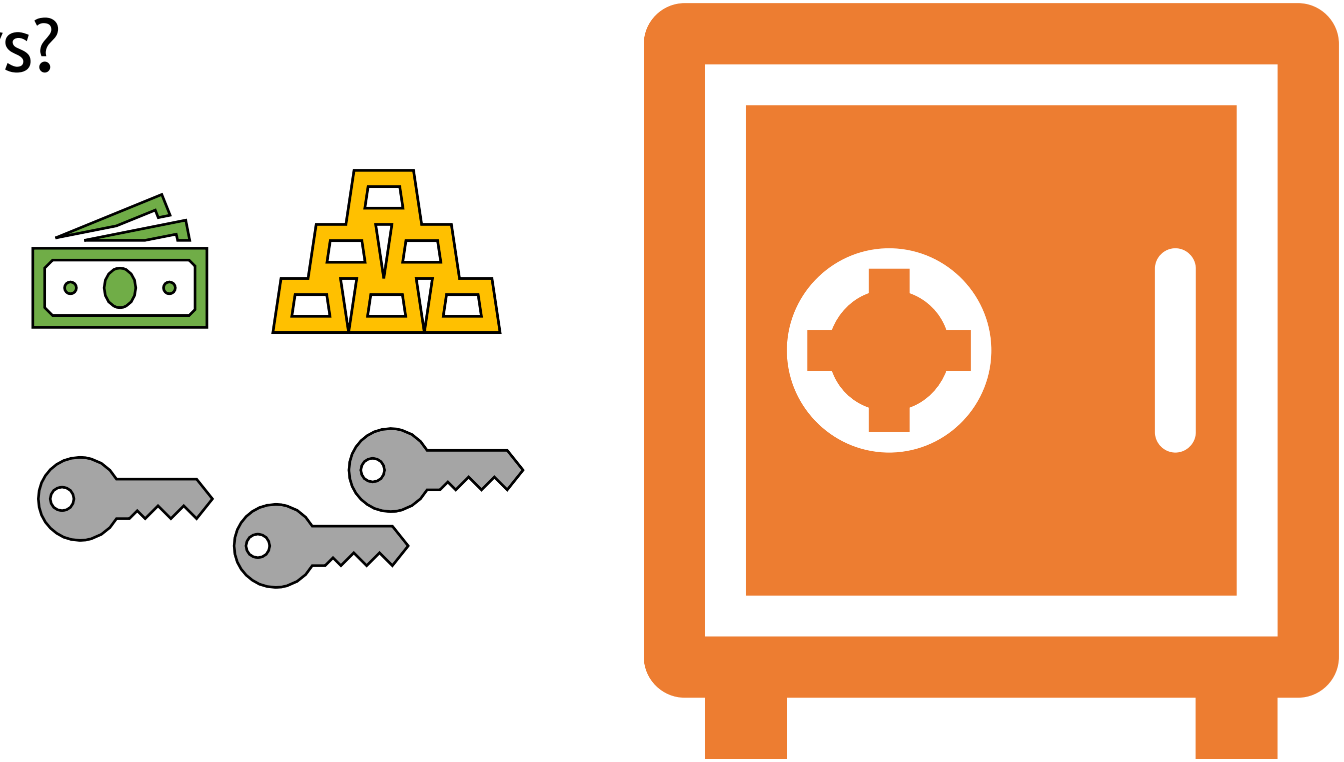
- **Daniel Moghimi**
- Berk Sunar
- Thomas Eisenbarth
- Nadia Heninger

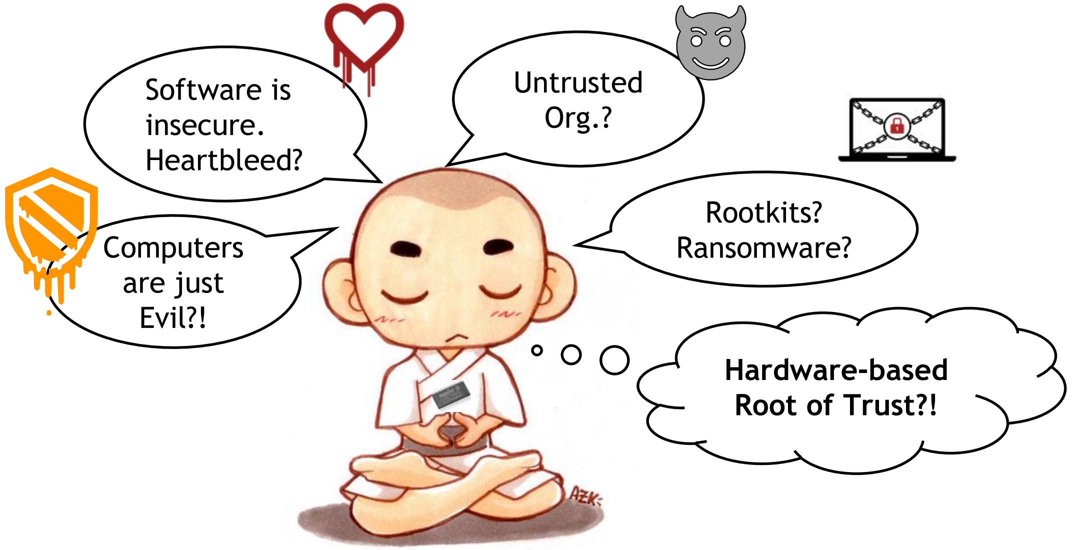
29TH USENIX SECURITY SYMPOSIUM

AUGUST 12–14, 2020

Trusted Platform Module (TPM)

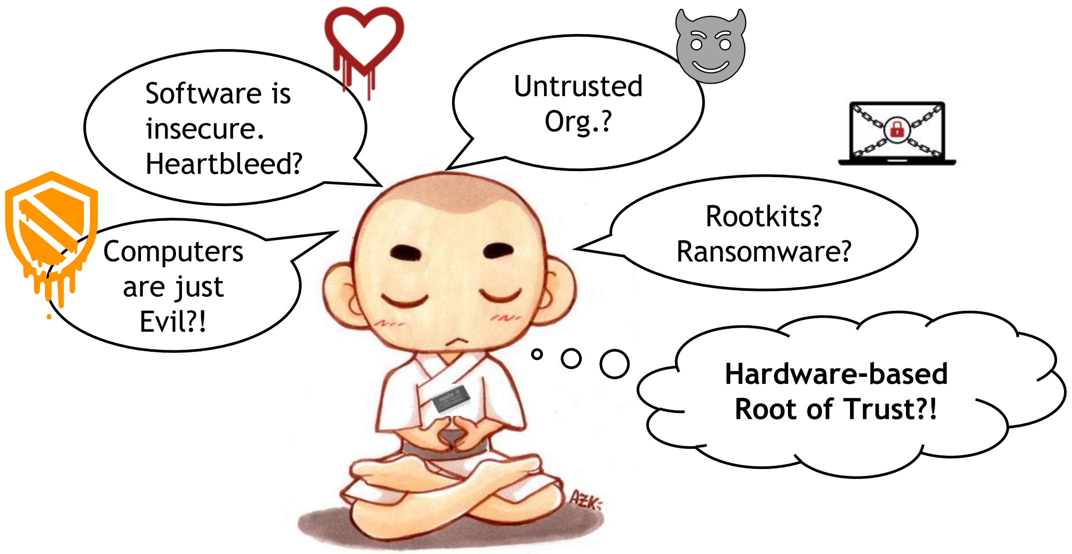
- Security Chip for Computers?
- Tamper Resistant
- Side-Channel Resistant
- Crypto Co-processor

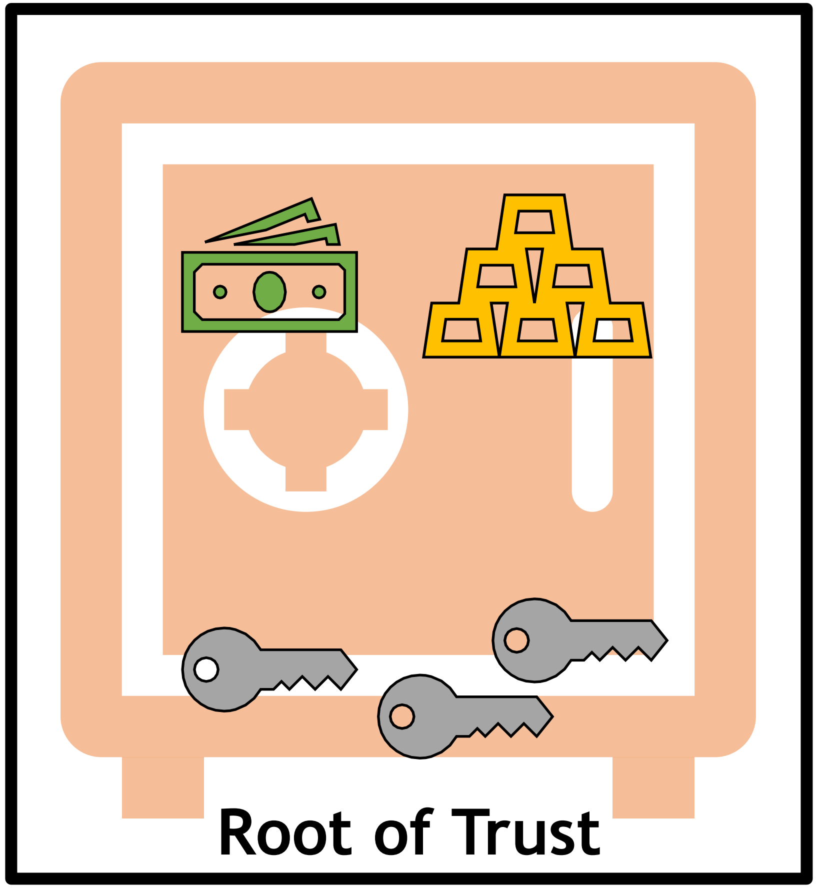




Trusted Platform Module (TPM)

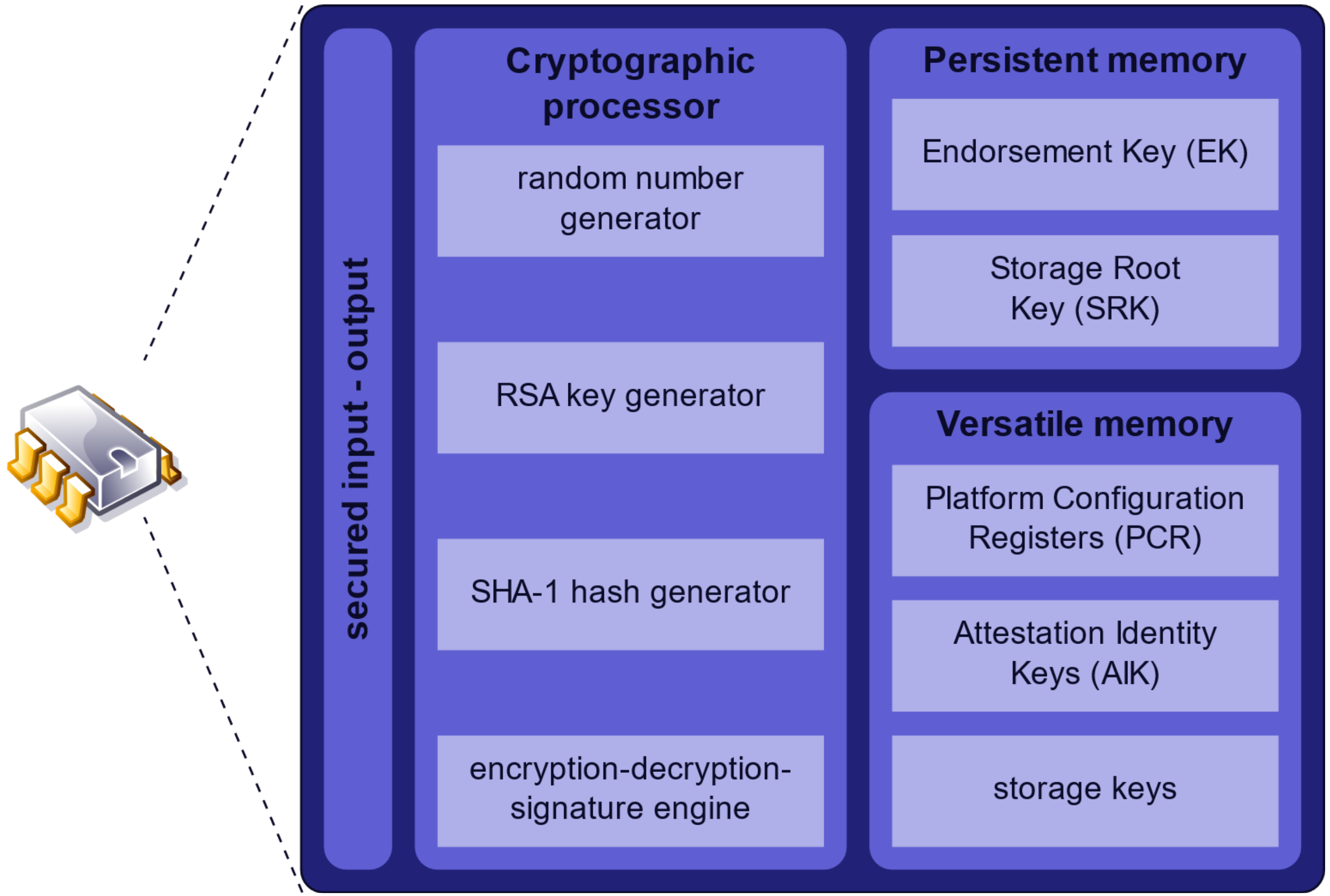
- Security Chip for Computers?
- Tamper Resistant
- Side-Channel Resistant
- Crypto Co-processor





Trusted Platform Module (TPM)

- Cryptographic Co-processor, specified by **Trusted Computing Group**
  - Secure Storage
  - Integrity Measurement
  - TRNG
  - Hash Functions
  - Encryption
  - **Digital Signatures**



TPM – Digital Signatures

- Applications
  - Trusted Execution of Signing Operations
  - Remote Attestation
- TPM 2.0 supports Elliptic-Curve Digital Signature
  - ECDSA
  - ECSchnorr
  - ECDAA (Anonymous Remote Attestation)

Trusted Computing Group – EAL 4+ Moderate

- https://trustedcomputinggroup.org/membership/certification/

TPM Security Evaluation

TCG members are required to demonstrate successful Common Criteria certification of their TPM product.

For the TPM 1.2 Family, the Common Criteria Security Assurance Level is at EAL4+ Moderate, in accordance to the PC Client TPM 1.2 Protection Profile by the TCG.

For the TPM 2.0 Family, the Common Criteria Security Assurance Level is at EAL4+ Moderate, in accordance to the PC Client TPM 2.0 Protection Profile by the TCG.

- https://trustedcomputinggroup.org/membership/certification/tpm-certified-products/

TPM Certified Products

TCG Certified Programs | TNC Certified Products List | Storage Certified Products List

Search:

Company Name	Product Name	Product Revision	Specification	Details	Security Evaluation	Cert. Status	Cert. Complete Date
STMicroelectronics	TPM ST33TPHF2X	1.256, 1.257, 2.256	Version 2.0 - Revision 1.38		Completed	Completed	2019.10.18
STMicroelectronics	TPM ST33GTPMA	3.256, 6.526	Version 2.0 - Revision 1.38		Completed	Completed	2019.10.18
Nuvoton Technologies Corporation (NTC)	TPM NPCT75x	7.4.0.0	Version 1.2 - Revision 116		Complete	Complete	2019.08.14
Nuvoton Technologies Corporation (NTC)	TPM NPCT75x	7.2.1.0	Version 2.0 - Revision 1.38		Complete	Complete	2019.01.18
Infineon Technologies	TPM SLI9670 TPM SLM9670	13.11	Version 2.0 - Revision 1.38		Complete	Complete	2018.12.18
Infineon Technologies	TPM SLB9670	7.85	Version 2.0 -		Complete	Complete	2018.10.29

- ST33TPHF2ESPI Data Brief: https://www.st.com/resource/en/data_brief/st33tphf2espi.pdf

- ST33TPHF2ESPI CC Evaluation: https://www.ssi.gouv.fr/uploads/2018/10/anssi-cible-cc-2018_41en.pdf

Intrinsic countermeasures for cryptographic algorithm against side channel attacks like timing attacks (TA), SPA and DPA.

Detection of abnormal behavior of the following operational conditions:

- High voltage supply
- Glitches

Detection of abnormal TOE behavior:

- MPU error
- TRNG failure

Are TPMs really side-channel resistant?

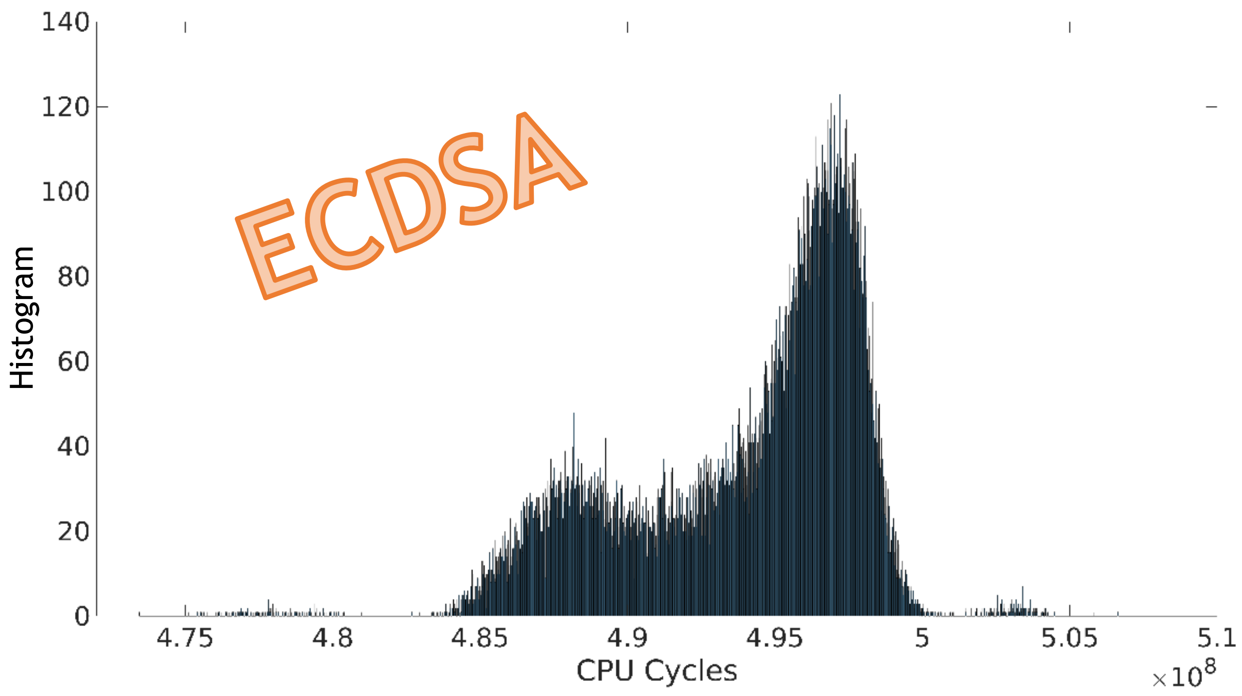
High-resolution Timing Test

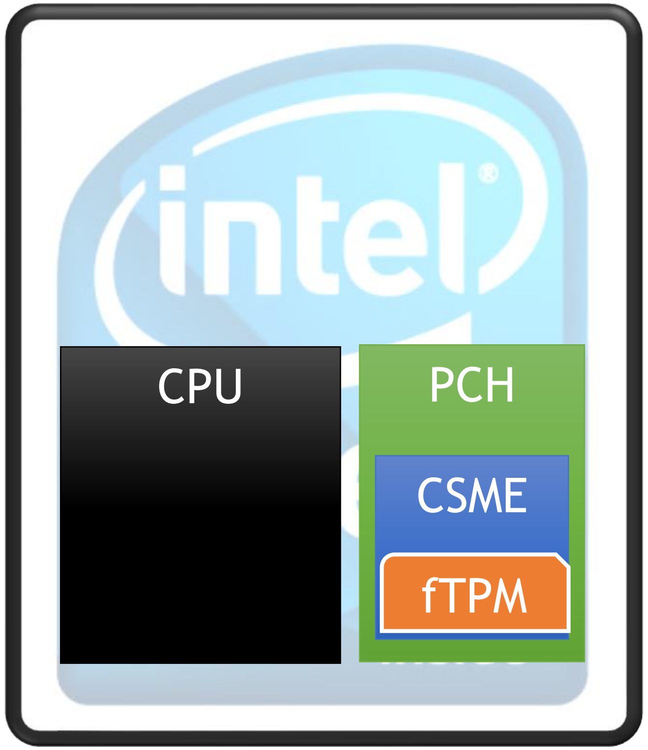
- TPM frequency ~= 32-120 MHz
- CPU Frequency is more than 2 GHz

100x faster!!

High-resolution Timing Test – Intel PTT (fTPM)

- Intel Platform Trust Technology (PTT)
  - Integrated firmware-TPM inside the CPU package
  - Runs on top of Converged Security and Management Engine (CSME)

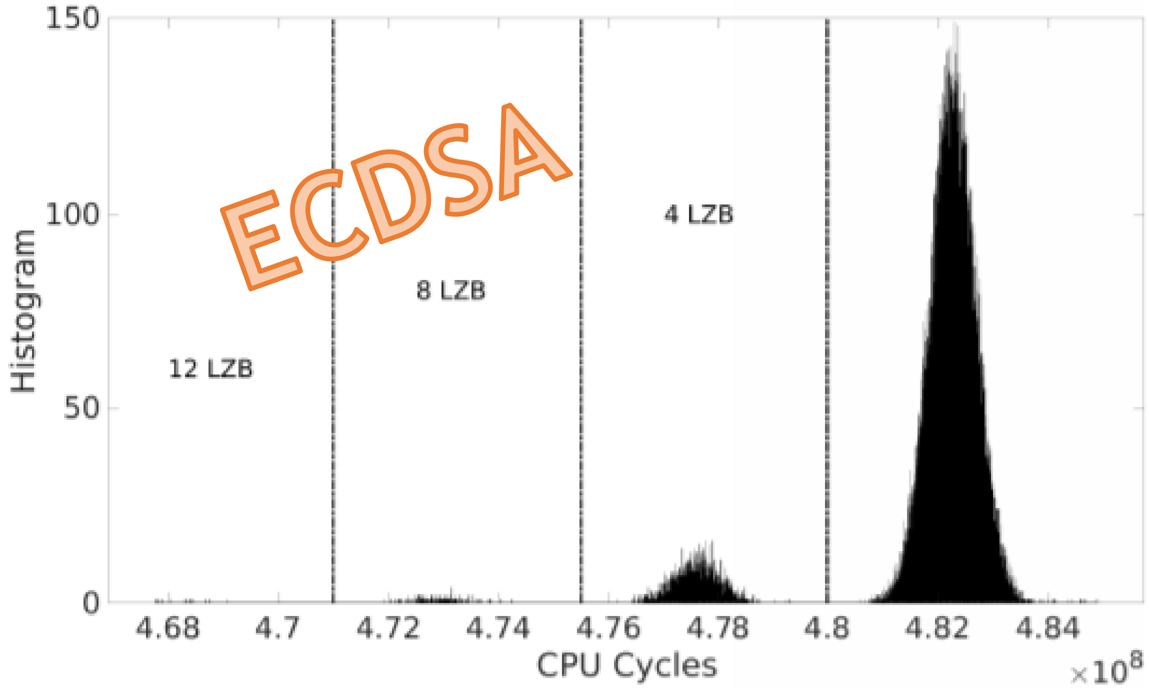


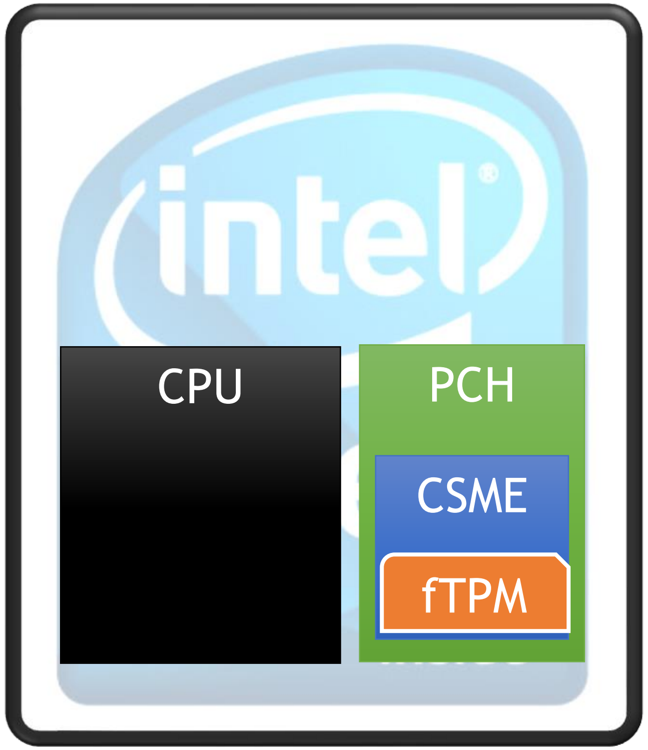


High-resolution Timing Test – Intel PTT (fTPM)

- Linux TPM Command Response Buffer (CRB) driver
- Kernel Driver to increase the Resolution

```
t = rdtsc ();
iowrite32 (CRB_START_INVOKE, &g_priv->regs_t->ctrl_start);
while ((ioread32(&g_priv->regs_t->ctrl_start) &
        CRB_START_INVOKE) == CRB_START_INVOKE);
tscrequest [requestcnt++] = rdtsc () - t;
```





High-resolution Timing Test – ECDSA Nonce Leakage

29TH USENIX SECURITY SYMPOSIUM

- Intel fTPM: 4-bit Window Nonce Length Leakage
  - ECDSA
  - ECSChnorr
  - BN-256 (ECDAA)

$ECDSA\ Sign$:
$(x_1, y_1) = k_i \times G$
$r_i = x_1 \bmod n$
$s_i = k_i^{-1}(z + r_i d) \bmod n$

Nonce

01010001001111111...111
00001001001111111...111
11010001001111111...111
00000000001111111...111
00000000000001111...111

4.67 4.72 4.76 4.8 4.84 t

High-resolution Timing Test – ECDSA Nonce Leakage

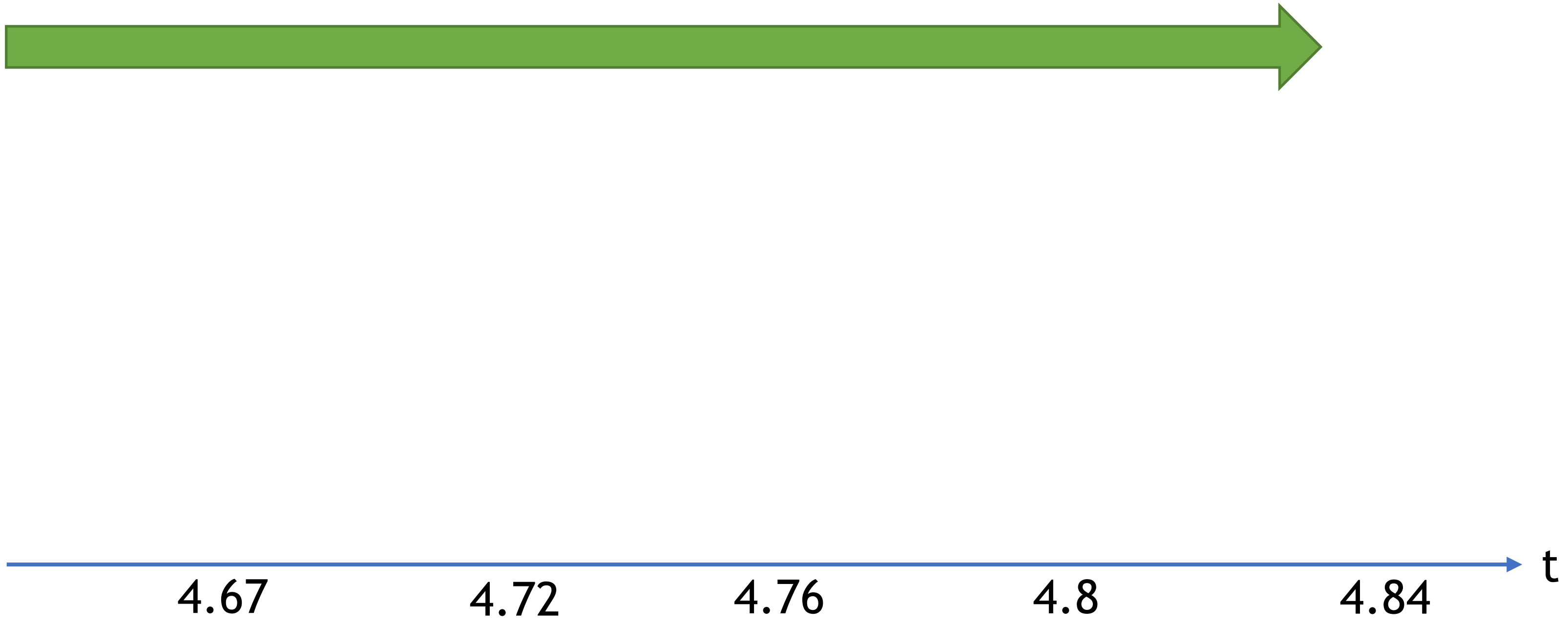
- Intel fTPM: 4-bit Window Nonce Length Leakage
  - ECDSA
  - ECSChnorr
  - BN-256 (ECDAA)

ECDSA Sign:

$(x_1, y_1) = k_i \times G$

$r_i = x_1 \bmod n$

$s_i = k_i^{-1}(z + r_i d) \bmod n$



Nonce

01010001001111111...111
00001001001111111...111
11010001001111111...111
00000000001111111...111
00000000000001111...111

High-resolution Timing Test – ECDSA Nonce Leakage

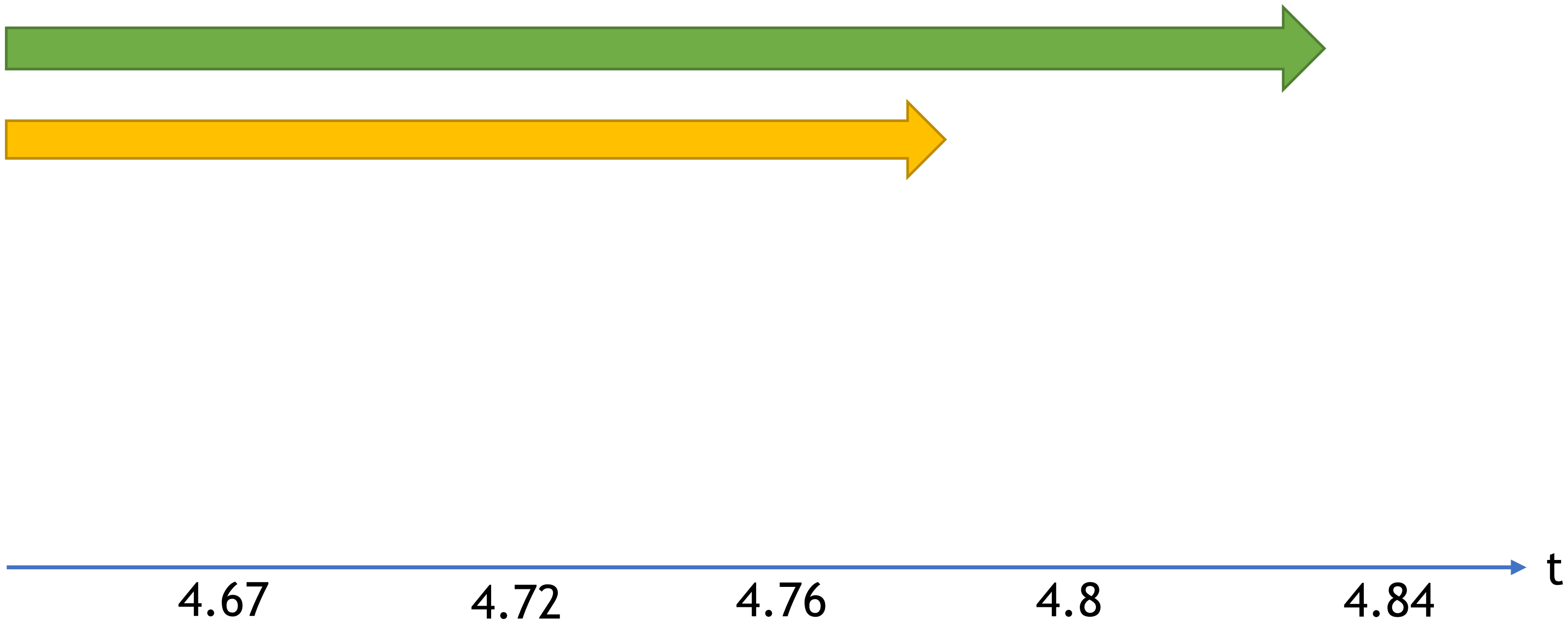
- Intel fTPM: 4-bit Window Nonce Length Leakage
  - ECDSA
  - ECSChnorr
  - BN-256 (ECDAA)

$ECDSA\ Sign$:

$(x_1, y_1) = k_i \times G$

$r_i = x_1 \bmod n$

$s_i = k_i^{-1}(z + r_i d) \bmod n$



Nonce

01010001001111111...111
00001001001111111...111
11010001001111111...111
00000000001111111...111
00000000000001111...111

High-resolution Timing Test – ECDSA Nonce Leakage

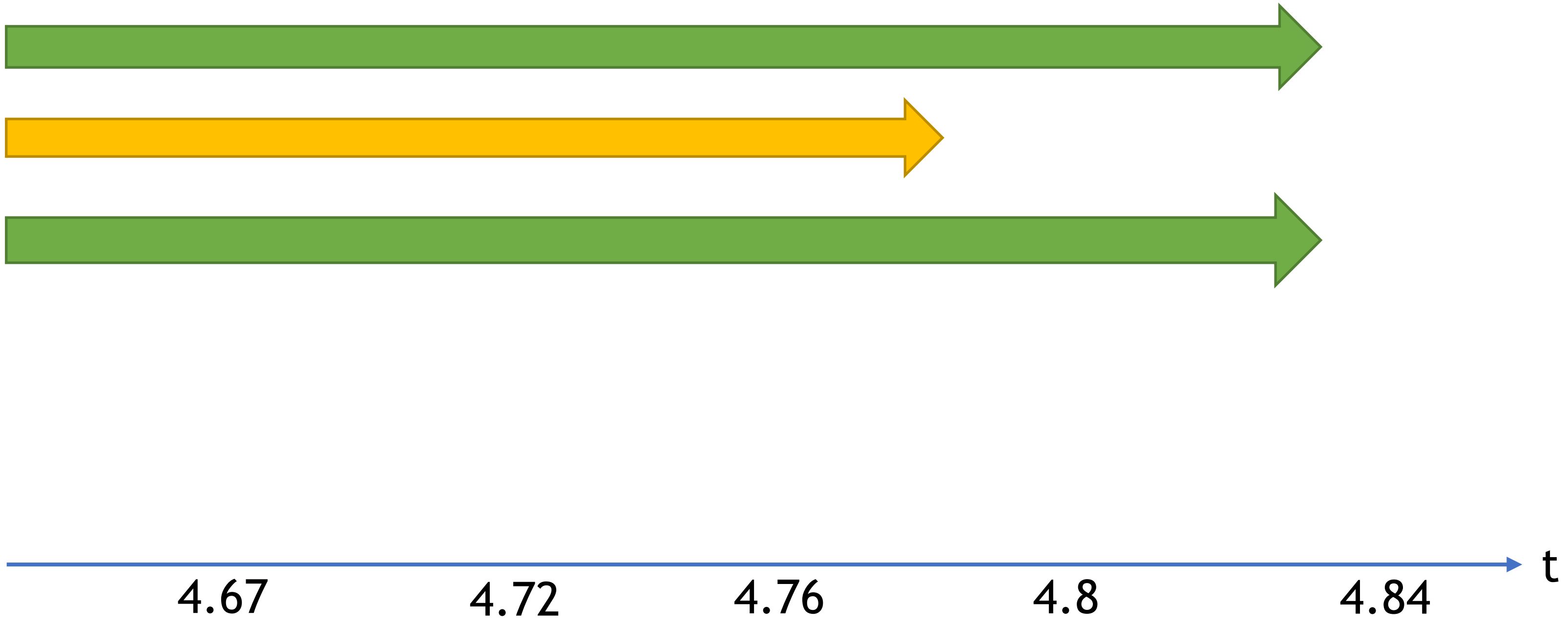
- Intel fTPM: 4-bit Window Nonce Length Leakage
  - ECDSA
  - ECSChnorr
  - BN-256 (ECDAA)

ECDSA Sign:

$(x_1, y_1) = k_i \times G$

$r_i = x_1 \; mod \; n$

$s_i = {k_i}^{-1}(z + r_i d) \; mod \; n$



Nonce

- 0101000100111111...111
- 0000100100111111...111
- 1101000100111111...111
- 0000000000111111...111
- 0000000000001111...111

High-resolution Timing Test – ECDSA Nonce Leakage

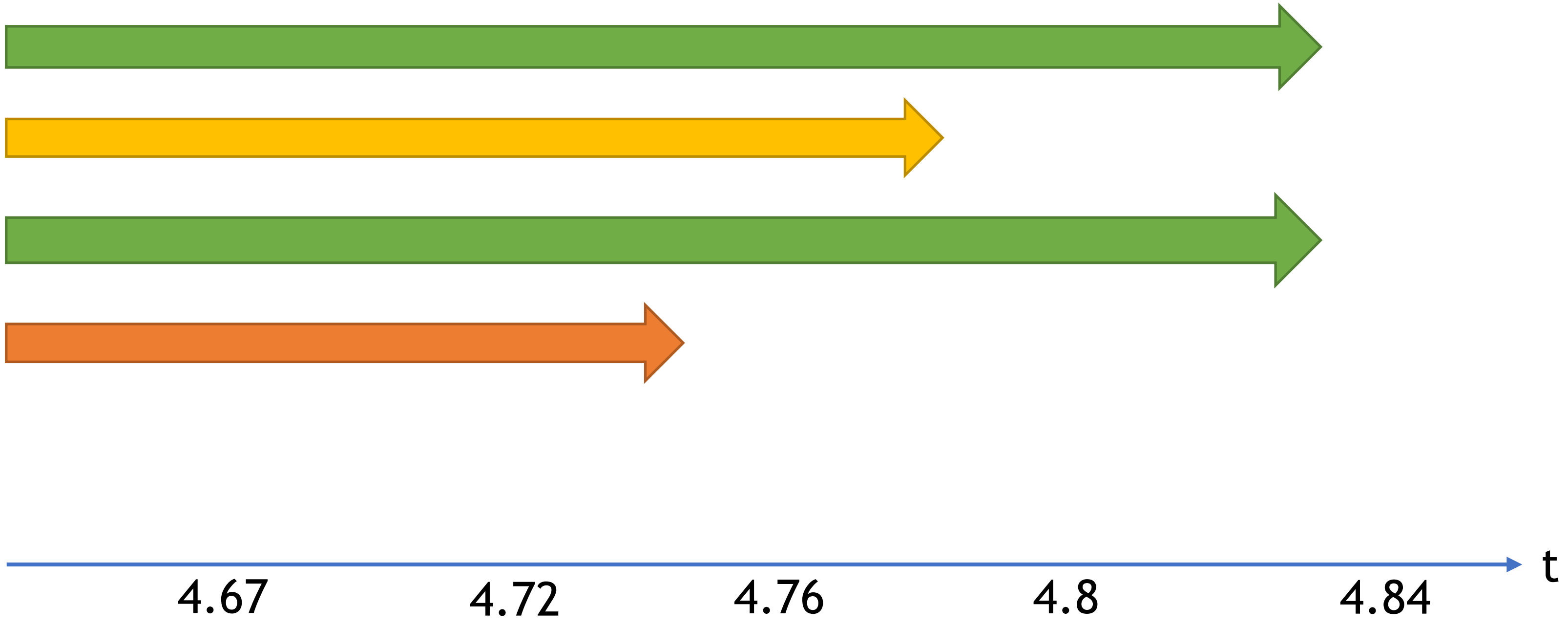
- Intel fTPM: 4-bit Window Nonce Length Leakage
  - ECDSA
  - ECSChnorr
  - BN-256 (ECDAA)

$ECDSA\ Sign$:

$(x_1, y_1) = k_i \times G$

$r_i = x_1 \ mod\ n$

$s_i = k_i^{-1}(z + r_i d) \ mod\ n$



Nonce

01010001001111111...111

00001001001111111...111

11010001001111111...111

00000000001111111...111

00000000000001111...111

High-resolution Timing Test – ECDSA Nonce Leakage

- Intel fTPM: 4-bit Window Nonce Length Leakage
  - ECDSA
  - ECSChnorr
  - BN-256 (ECDAA)

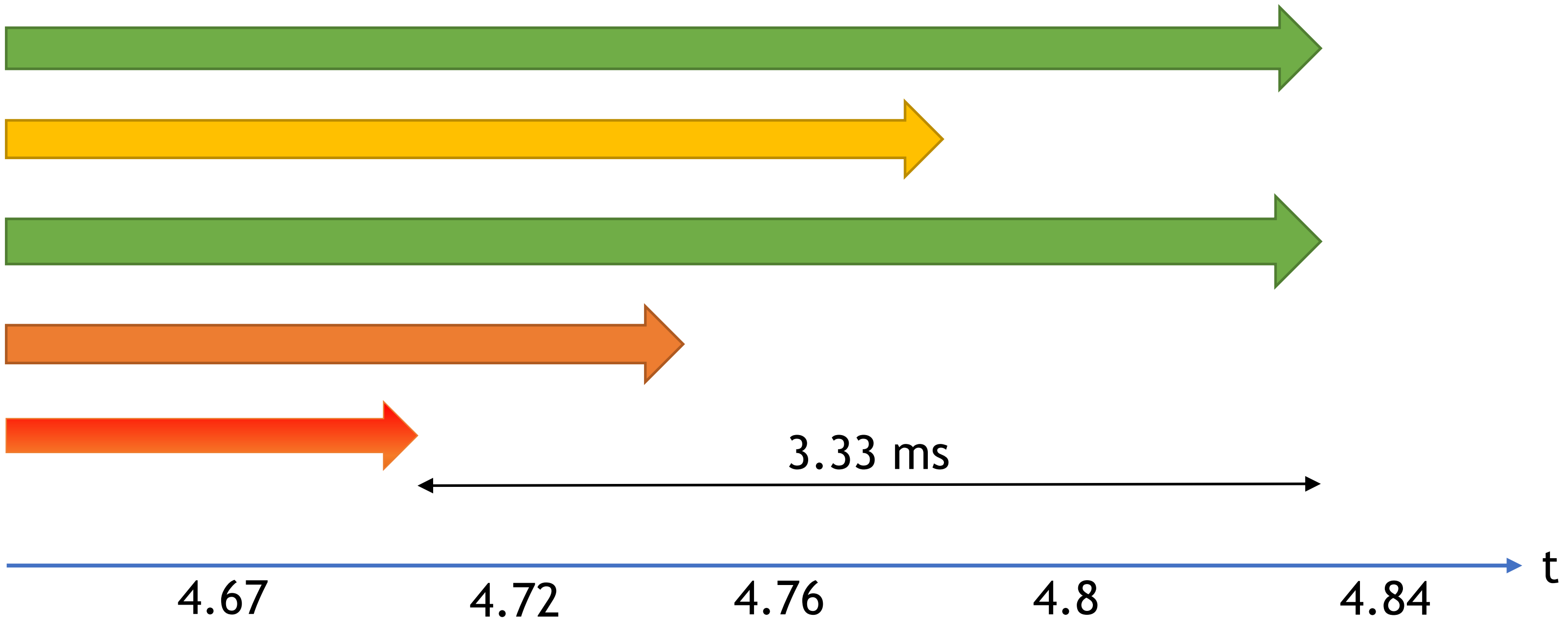
ECDSA Sign:

$(x_1, y_1) = k_i \times G$

$r_i = x_1 \; mod \; n$

$s_i = k_i^{-1}(z + r_i d) \; mod \; n$

Nonce



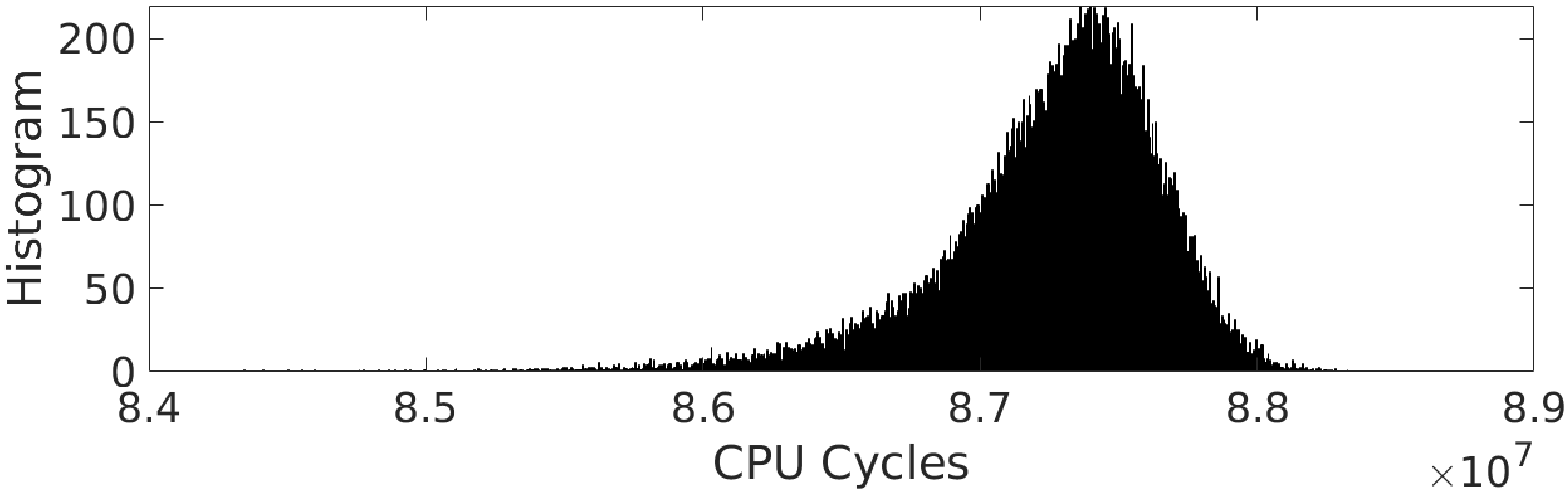
0101000100111111...111
0000100100111111...111
1101000100111111...111
0000000000111111...111
0000000000001111...111

High-resolution Timing Test – Analysis Of Devices

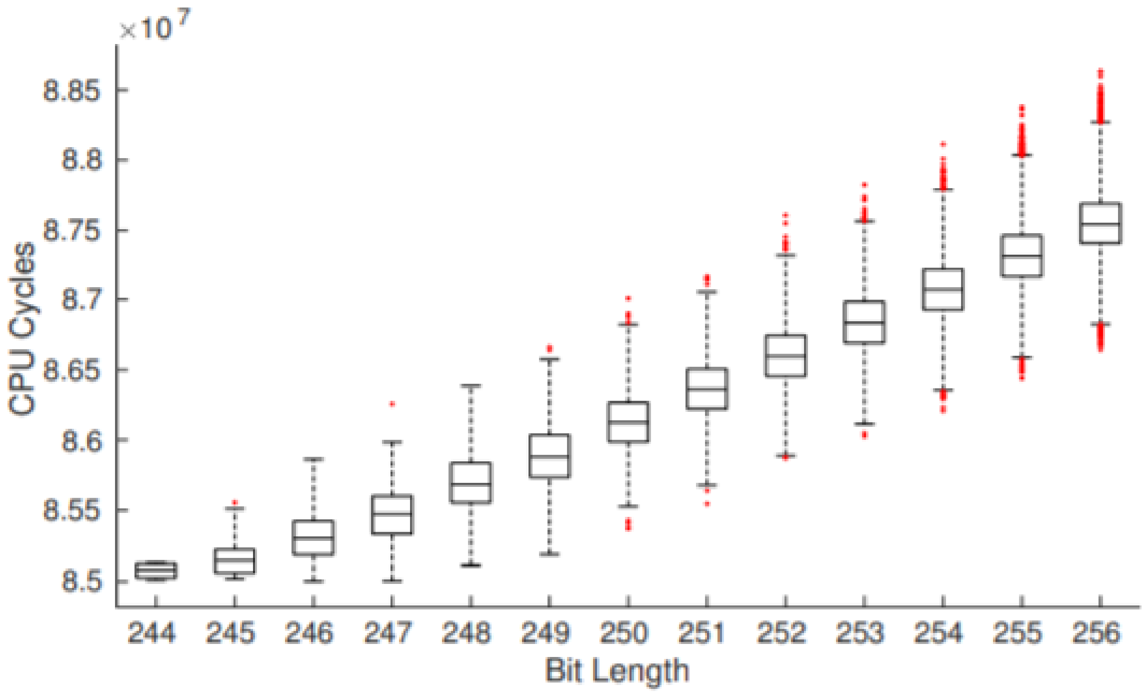
- RSA and ECDSA timing test on 3 dedicated TPM and Intel fTPM
- Various non-constant behaviour for both RSA and ECDSA

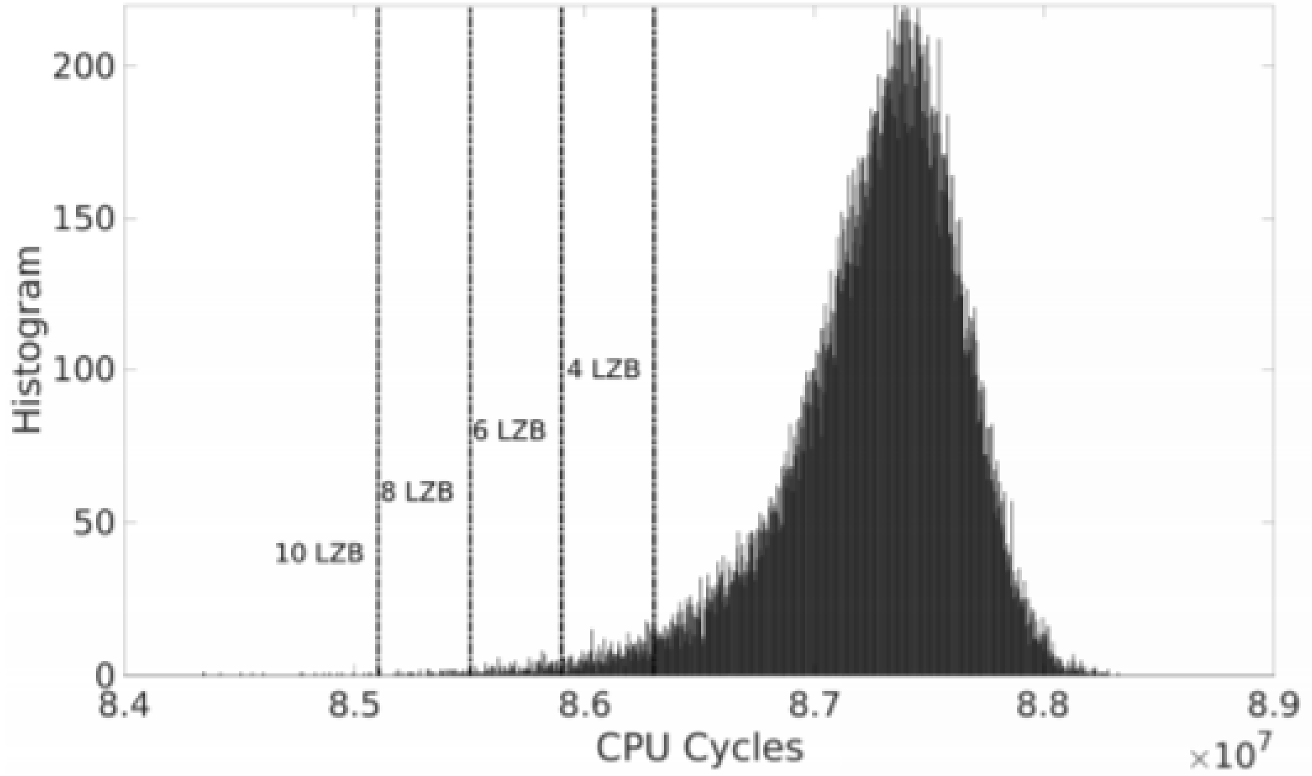
Machine	CPU	Vendor	TPM	Firmware/Bios
NUC 8i7HNK	Core i7-8705G	Intel	PTT (fTPM)	NUC BIOS 0053
NUC 7i3BNK	Core i3-7100U	Intel	PTT (fTPM)	NUC BIOS 0076
Asus GL502VM	Core i7-6700HQ	Intel	PTT (fTPM)	Latest OEM
Asus K501UW	Core i7 6500U	Intel	PTT (fTPM)	Latest OEM
Dell XPS 8920	Core i7-7700	Intel	PTT (fTPM)	Dell BIOS 1.0.4
Dell Precision 5510	Core i5-6440HQ	Nuvoton	rls NPCT	NTC 1.3.2.8
Lenovo T580	Core i7-8650U	STMicro	ST33TPHF2ESPI	STMicro 73.04
NUC 7i7DNKE	Core i7-8650U	Infineon	SLB 9670	NUC BIOS 0062

STMicroelectronics - ECDSA



- STMicroelectronics’ TPM: Bit-by-Bit Nonce Length Leakage





TPM-Fail – Recovering Private ECDSA Key

- TPM is programmed with an unknown key
- We already have a template for t_i.

1. Collect list of signatures (r_i, s_i) and timing samples t_i.
2. Filter signatures based on t_i and keeps (r_i, s_i) with a known bias.
3. Lattice-based attack to recover private key d, from signatures with biased nonce k_i.

Lattice and Hidden Number Problem

29TH USENIX SECURITY SYMPOSIUM

- $s = k^{-1}(z + dr) \ mod \ n \rightarrow \ k_i^{-1} \ - s_i^{-1} r_i d - s_i^{-1} z \equiv 0 \ mod \ n$

Lattice and Hidden Number Problem

- $s = k^{-1}(z + dr) \bmod n \rightarrow k_i^{-1} - s_i^{-1} r_i d - s_i^{-1} z \equiv 0 \bmod n$
- $A_i = -s_i^{-1} r_i, B_i = -s_i^{-1} z \rightarrow k_i + A_i d + B_i = 0$

Lattice and Hidden Number Problem

- $s = k^{-1}(z + dr) \, mod \, n \rightarrow k_i^{-1} - s_i^{-1} r_i d - s_i^{-1} z \equiv 0 \, mod \, n$
- $A_i = -s_i^{-1} r_i, B_i = -s_i^{-1} z \rightarrow k_i + A_i d + B_i = 0$
- Let X be the upper bound on $\mathrm{k_i}$ and $(d, k_{0,} k_1 \dots, k_n)$ is unknown

Boneh and Venkatesan[1]

[1] Boneh D, Venkatesan R. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. InAnnual International Cryptology Conference 1996 Aug 18 (pp. 129-142). Springer, Berlin, Heidelberg.

Lattice and Hidden Number Problem

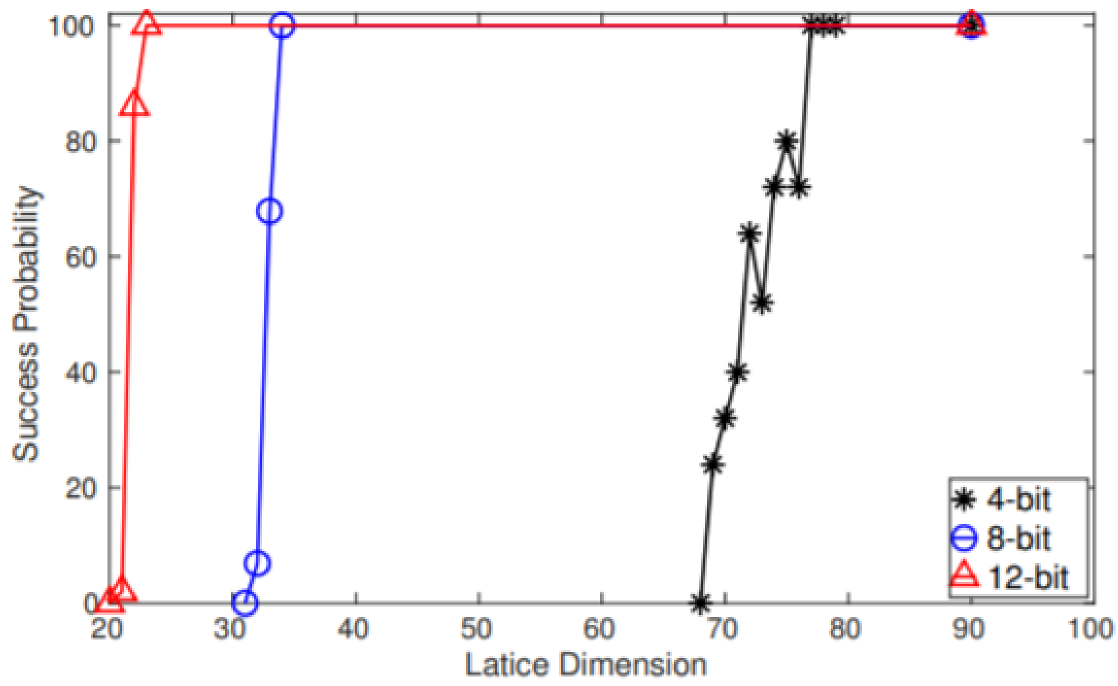
- $s = k^{-1}(z + dr) \bmod n \rightarrow k_i^{-1} - s_i^{-1} r_i d - s_i^{-1} z \equiv 0 \bmod n$
- $A_i = -s_i^{-1} r_i, B_i = -s_i^{-1} z \rightarrow k_i + A_i d + B_i = 0$
- Let X be the upper bound on $\mathrm{k_i}$ and $(d, k_0, k_1 \ldots, k_n)$ is unknown
- Lattice Construction:

$$\begin{bmatrix} n & & & & & \\ & n & & & & \\ & & \ddots & & & \\ & & & n & & \\ A_1 & A_2 & \ldots & A_t & \frac{X}{n} & \\ B_1 & B_2 & \ldots & B_t & & X \end{bmatrix}$$

TPM-Fail – Key Recovery Results

- Intel fTPM
  - ECDSA, ECSchnorr and BN-256 (ECDAA)
  - Three different threat model System, User, Network
- STMicroelectronics TPM
  - CC EAL4+ Certified
  - Give you the key in 80 minutes

Threat Model	TPM	Scheme	#Sign.	Time
Local System	ST TPM	ECDSA	39,980	80 mins
Local System	fTPM	ECDSA	1,248	4 mins
Local System	fTPM	ECSchnorr	1,040	3 mins
Local User	fTPM	ECDSA	15,042	18 mins



Remote Timing Attacks are Practical

David Brumley
Stanford University
dbrumley@cs.stanford.edu

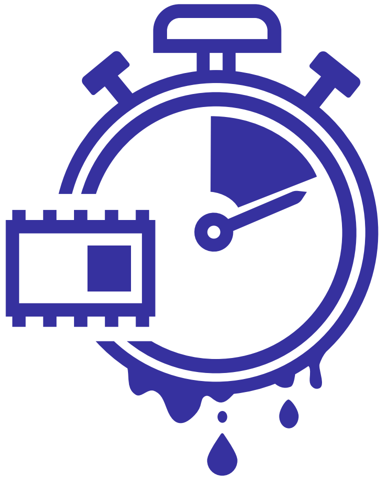
Dan Boneh
Stanford University
dabo@cs.stanford.edu

Abstract

Timing attacks are usually used to attack weak computing devices such as smartcards. We show that timing attacks apply to general software systems. Specifically, we devise a timing attack against OpenSSL. Our experiments show that we can extract private keys from an OpenSSL-based web server running on a machine in the local network. Our results demonstrate that timing attacks against network servers are practical and therefore security systems should defend against them.

The attacking machine and the server were in different buildings with three routers and multiple switches between them. With this setup we were able to extract the SSL private key from common SSL applications such as a web server (Apache+mod_SSL) and a SSL-tunnel.

Interprocess. We successfully mounted the attack between two processes running on the same machine. A hosting center that hosts two domains on the same machine might give management access to the admins of each domain. Since both domain are hosted on the same machine, one admin could use



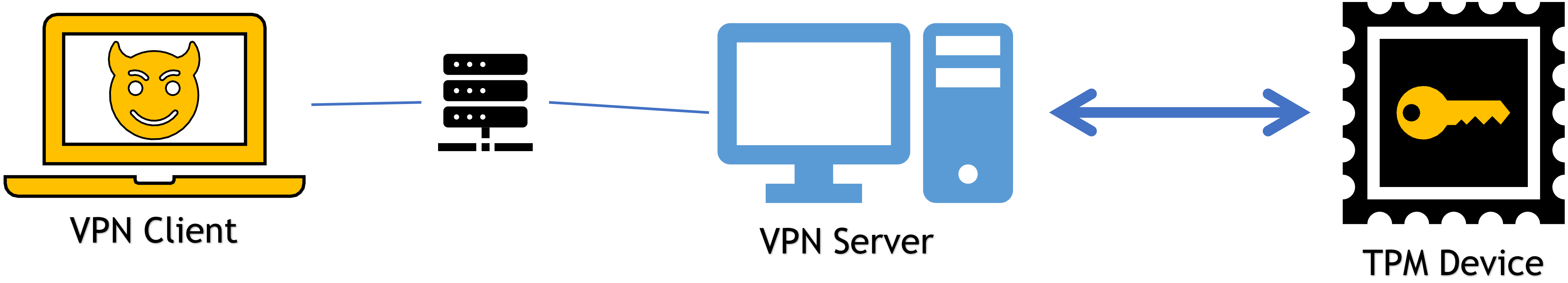
TPMs are extremely slow

Timing difference for each window	(4.76e8 - 4.72e8)/3600e6 * 1000 = **1.11 ms**
ping 192.168.1.x	average rtt **0.713 ms**
ping 1.1.1.1 (Cloudflare DNS)	average rtt **19.312 ms**

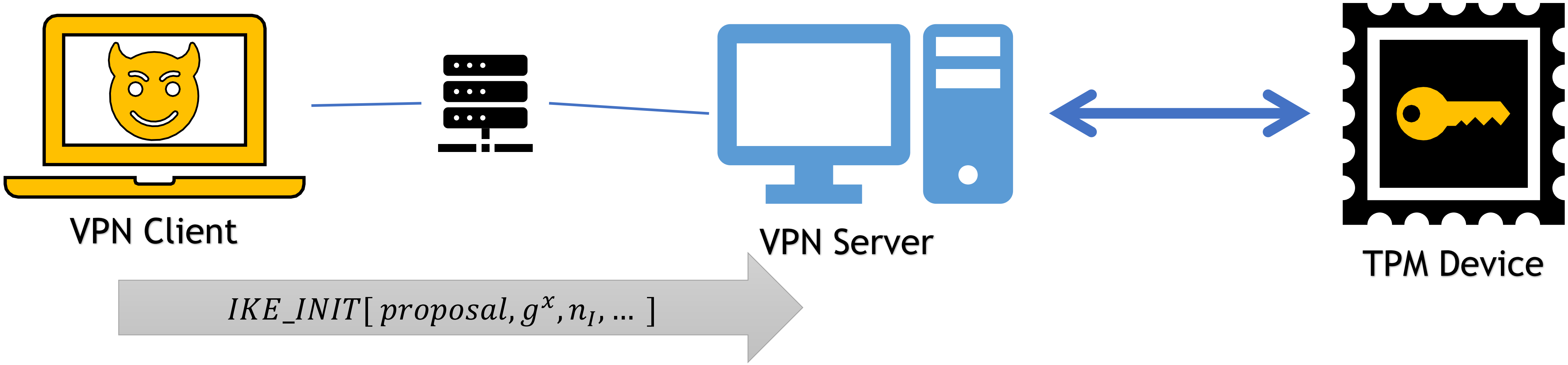
Remote Timing Attacks are Practical!!

Brumley D, Boneh D. Remote timing attacks are practical. Computer Networks. 2005 Aug 5;48(5):701-16.iv

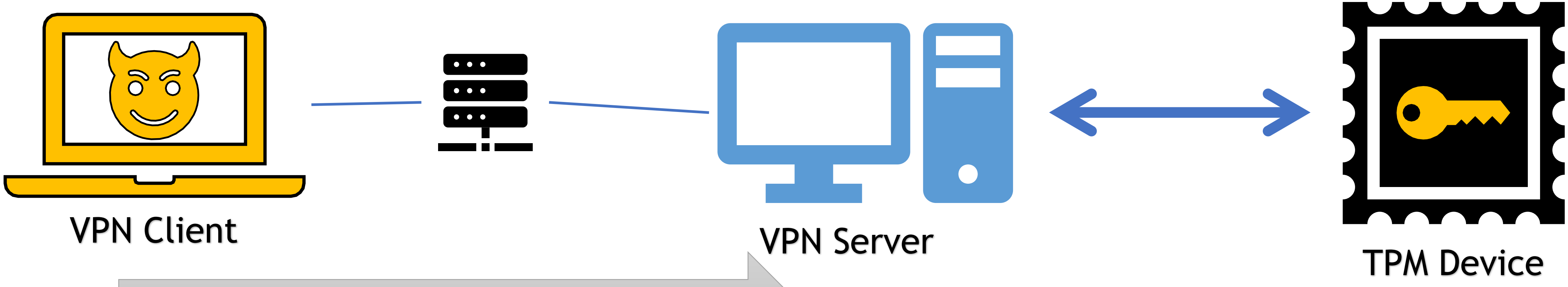
TPM-Fail Case Study: StrongSwan VPN



TPM-Fail Case Study: StrongSwan VPN



TPM-Fail Case Study: StrongSwan VPN

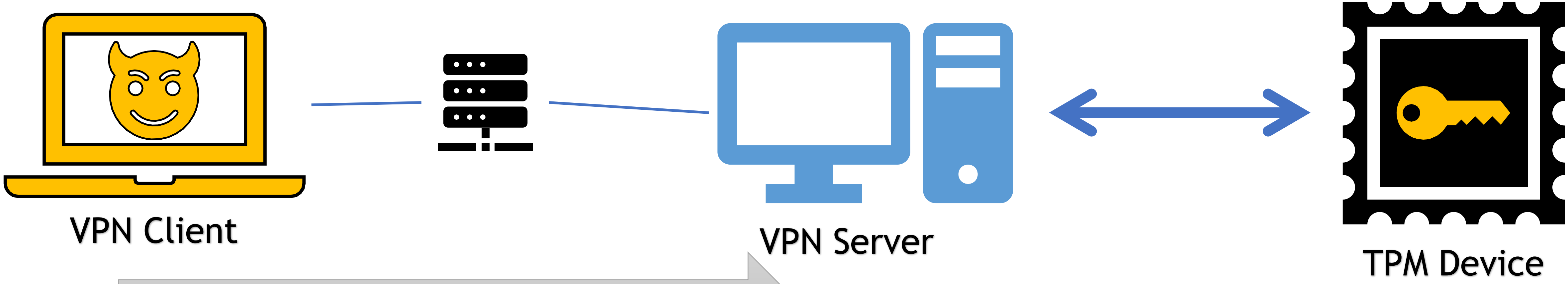


$IKE\_INIT[\,proposal, g^x, n_I, \ldots\,]$

$IKE\_INIT_{response}[\,proposal, g^x, n_R, \ldots\,]$

$$s_{shared-secret} = PRF_h(g^{xy})$$

TPM-Fail Case Study: StrongSwan VPN



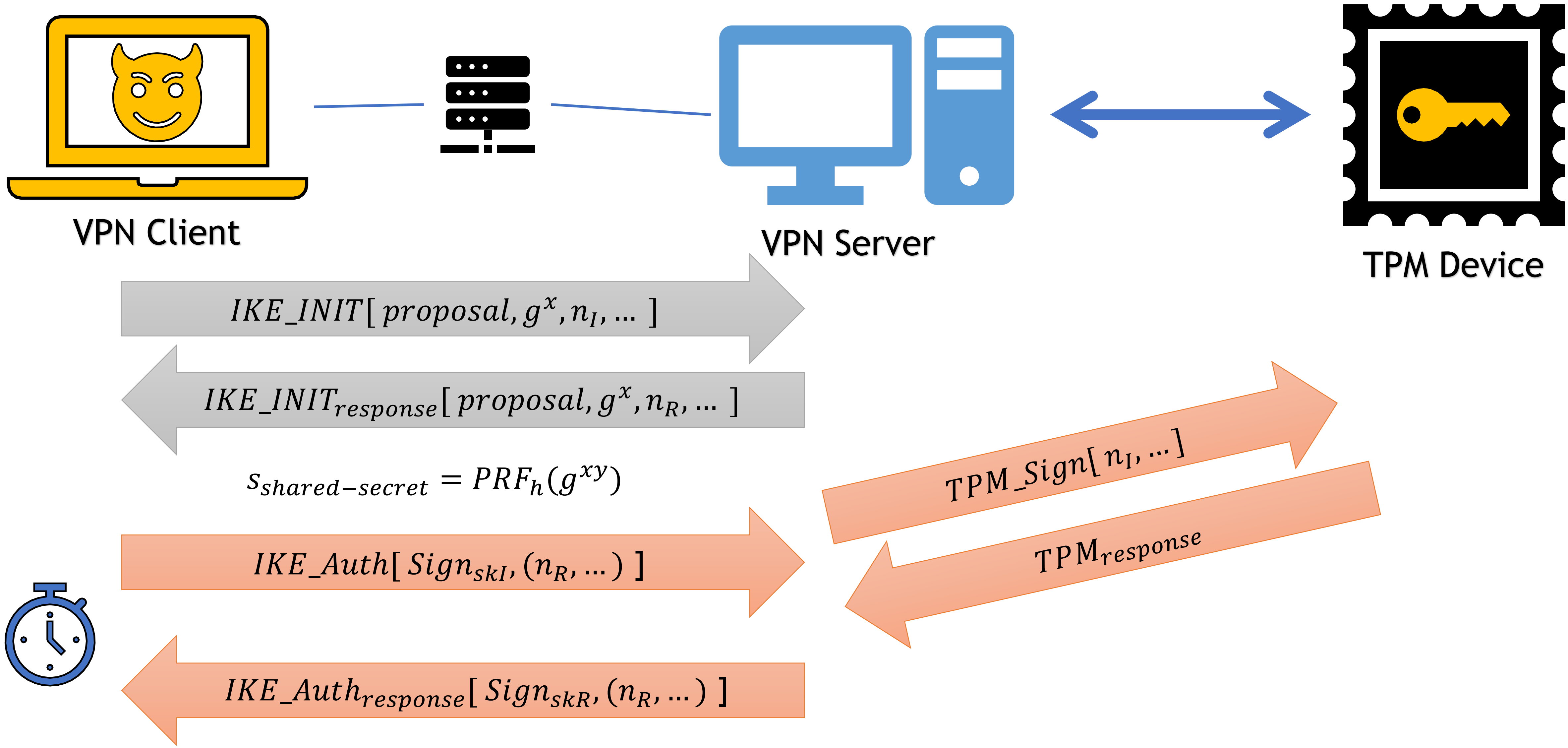
$IKE\_INIT[\, proposal, g^x, n_I, \dots\,]$

$IKE\_INIT_{response}[\, proposal, g^x, n_R, \dots\,]$

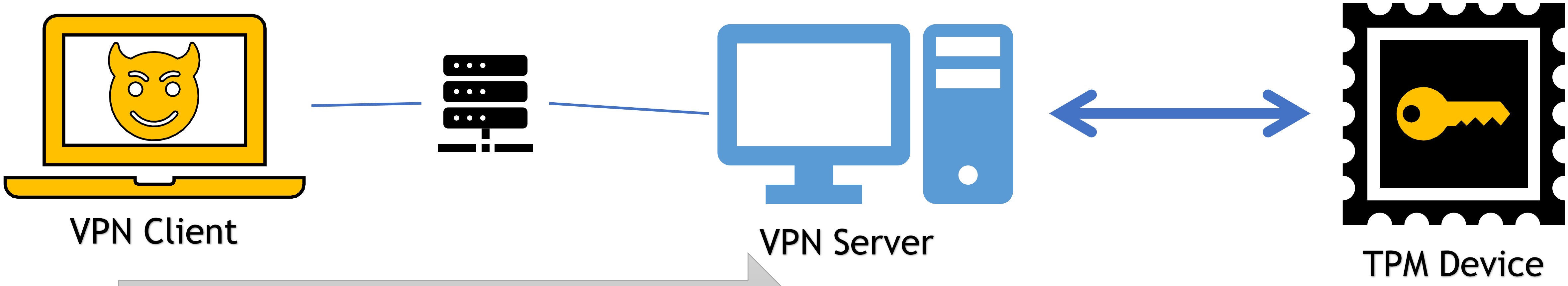
$s_{shared-secret} = PRF_h(g^{xy})$

$IKE\_Auth[\, Sign_{skI}, (n_R, \dots)\,]$

TPM-Fail Case Study: StrongSwan VPN



TPM-Fail Case Study: StrongSwan VPN



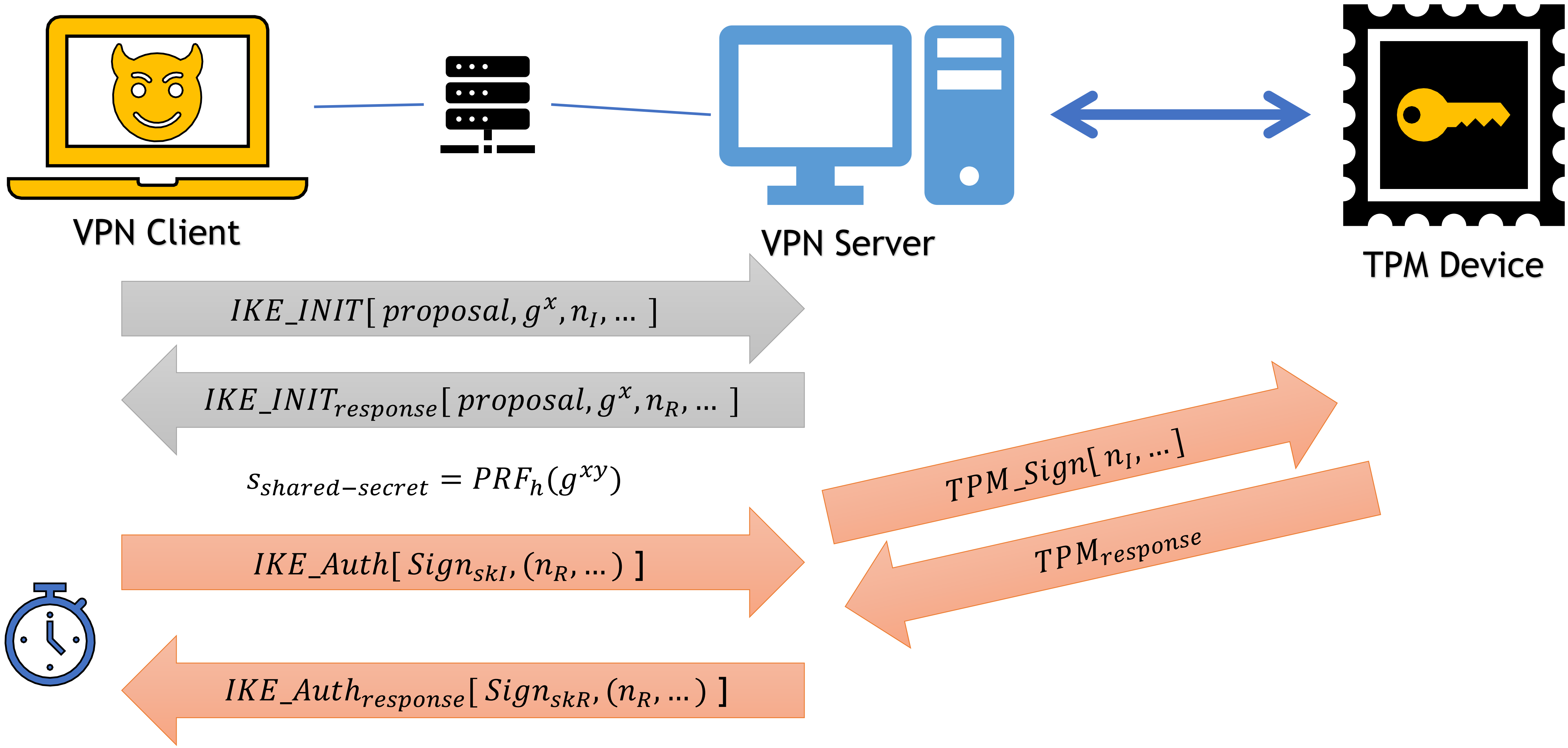
$$IKE\_INIT[\,proposal, g^x, n_I, \ldots\,]$$

$$IKE\_INIT_{response}[\,proposal, g^x, n_R, \ldots\,]$$

$$s_{shared-secret} = PRF_h(g^{xy})$$

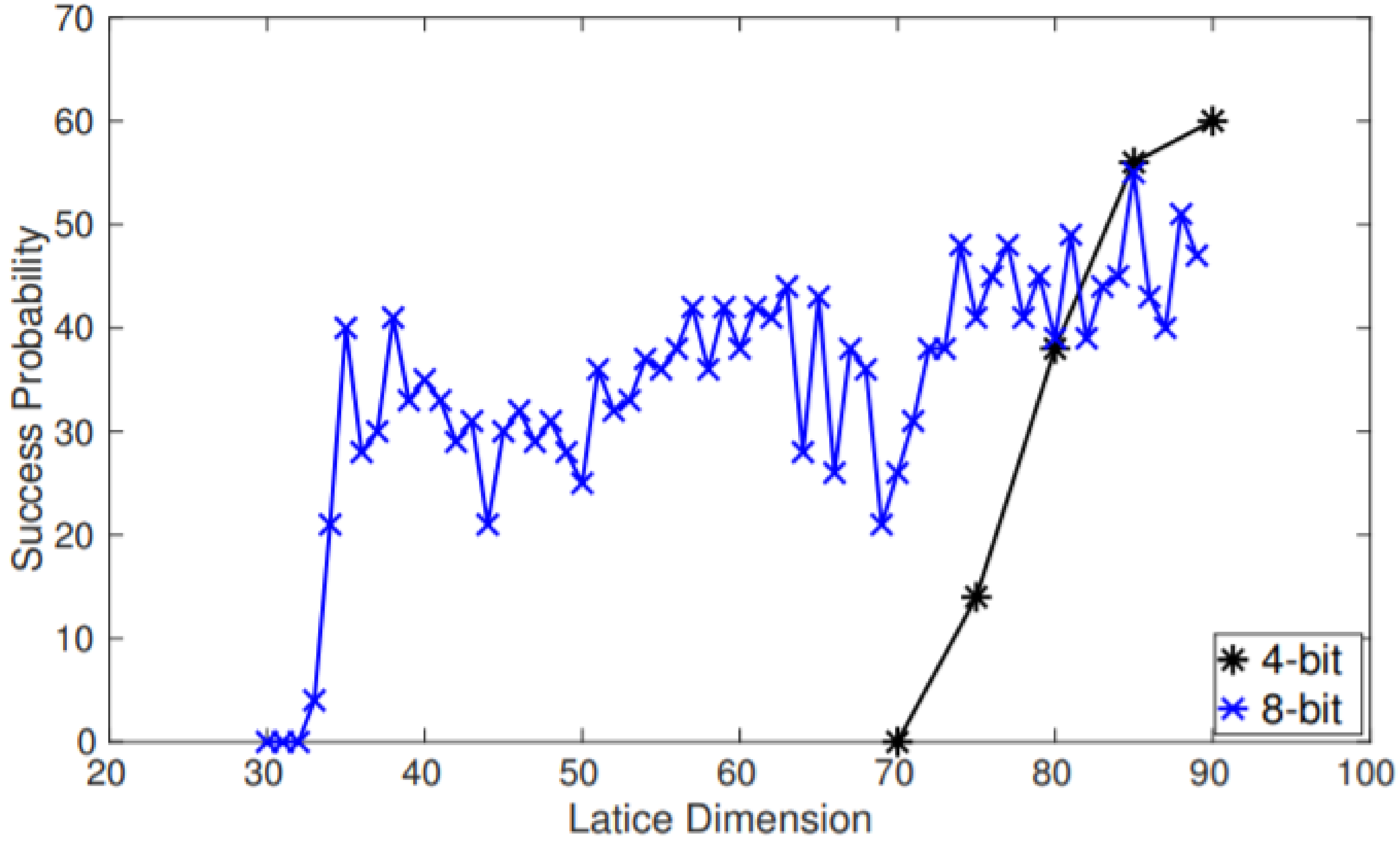
$$IKE\_Auth[\,Sign_{skI}, (n_R, \ldots)\,]$$

TPM-Fail Case Study: StrongSwan VPN

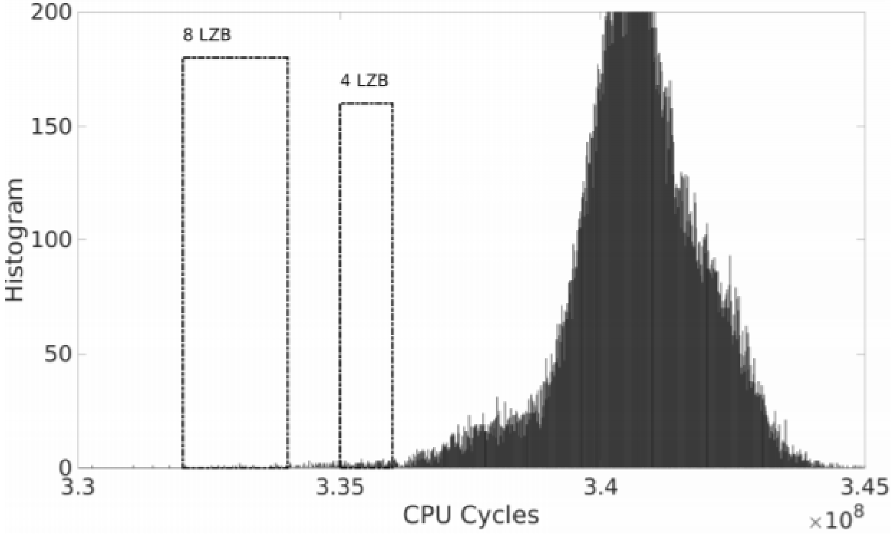


TPM-Fail Case Study: StrongSwan VPN

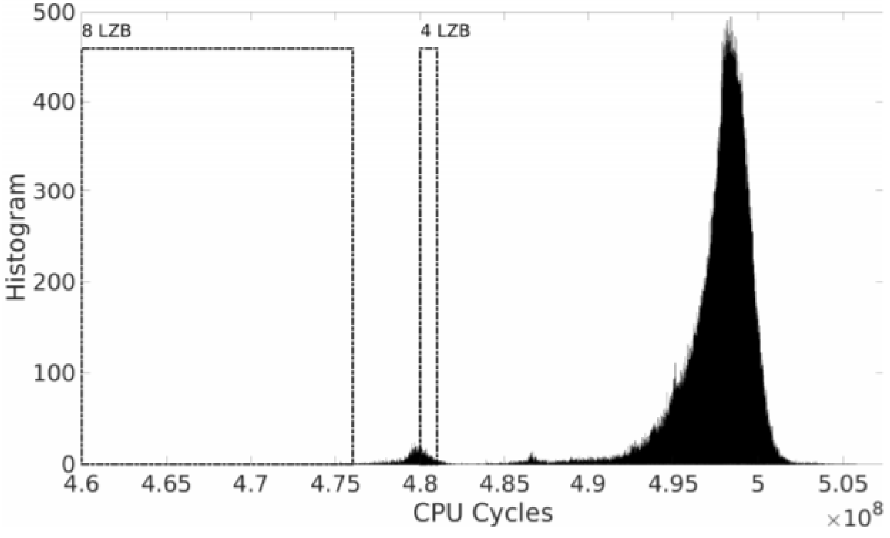
- Remote Key Recovery after about 44,000 handshake ~= 5 hours

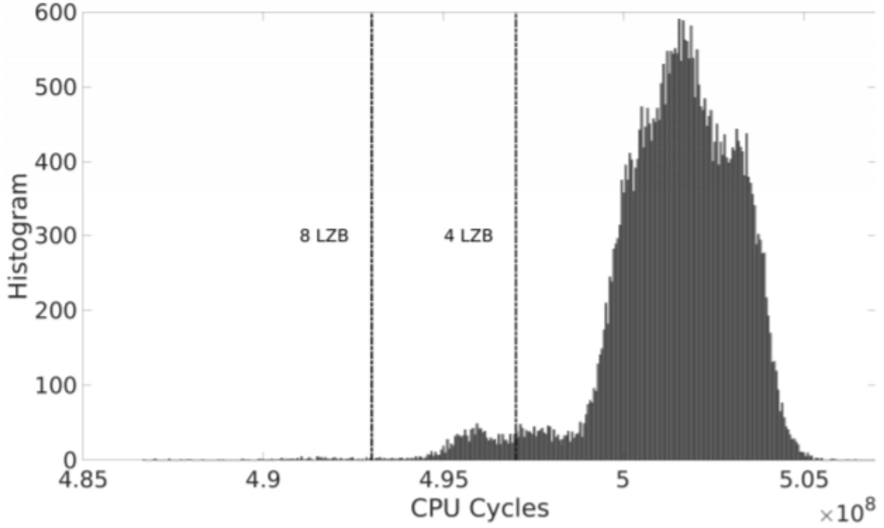


Remote StrongSwan VPN

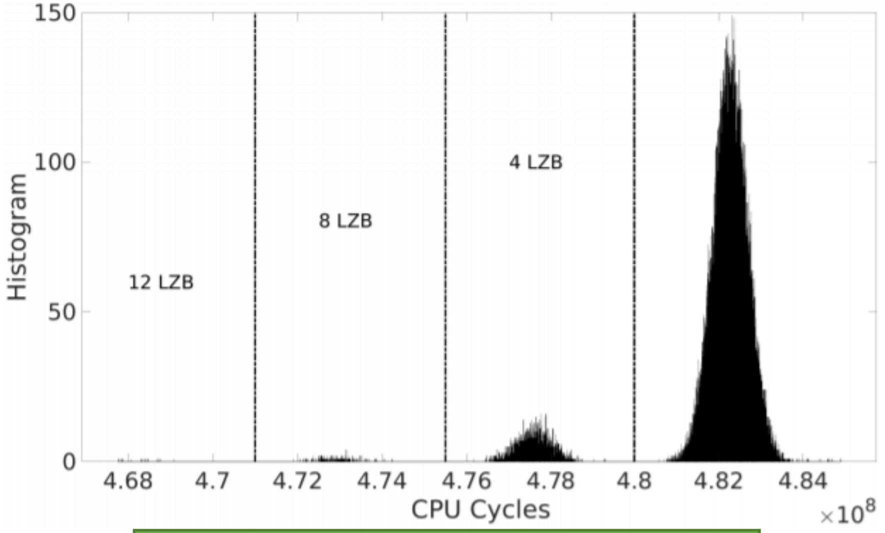


User Adversary





Remote Sample UDP App



System Adversary

Coordinated Disclosure - STMicroelectronics

- STMicroelectronics (CVE-2019-16863)
  - 05/15/2019: Reported to ST
  - 05/17/2019: Acknowledged
  - Lots of calls/emails to clarify the disclosure process
  - 09/12/2019: Verified new version of STM TPM firmware
  - After 11/12/2019:
    - HP and Lenovo have issued firmware updates.
    - ST released a list of affected devices.

- Intel (CVE-2019-11090)
  - 02/01/2019: Reported to IPSIRT
  - 02/12/2019: Acknowledged (Outdated Intel IPP Crypto library)
  - 11/12/2019: Firmware Update for Intel Management Engine

Questions?!

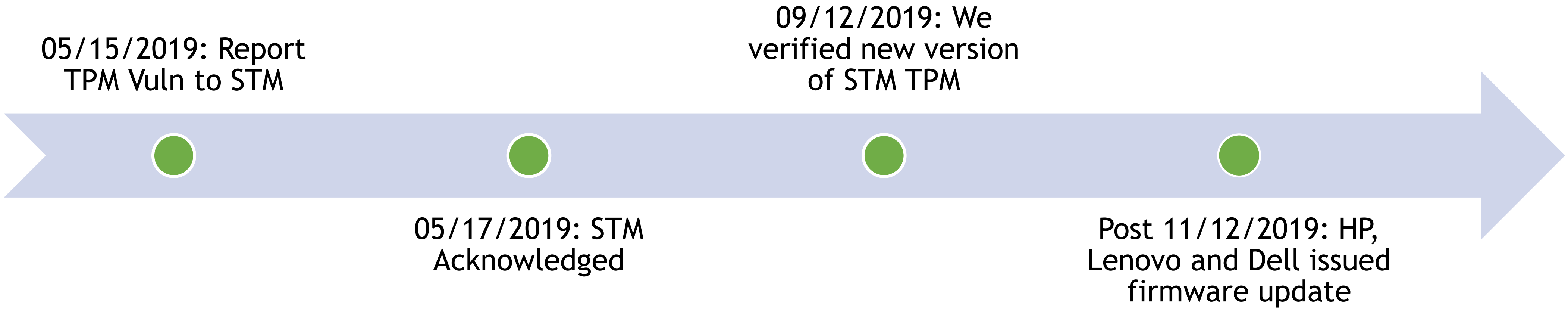
Daniel Moghimi
@danielmgmi

TPM-FAIL
https://tpm.fail/

https://github.com/VernamLab/TPM-Fail

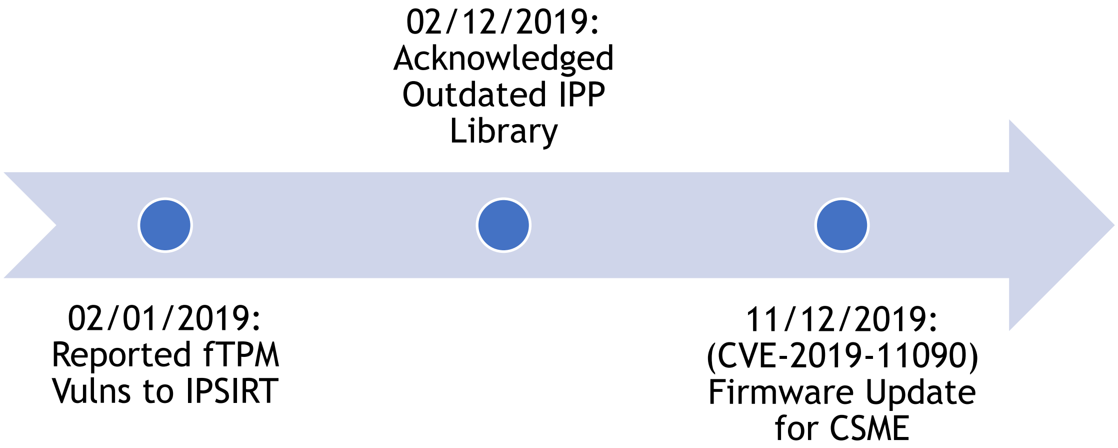
Coordinated Disclosure - STMicroelectronics

- STMicroelectronics (CVE-2019-16863)
  - 05/15/2019: Reported to ST
  - 05/17/2019: Acknowledged
  - Lots of calls/emails to clarify the disclosure process
  - 09/12/2019: Verified new version of STM TPM firmware
  - After 11/12/2019:
    - HP and Lenovo have issued firmware updates.
    - ST released a list of affected devices.



Coordinated Disclosure - Intel

- Intel (CVE-2019-11090)
  - 02/01/2019: Reported to IPSIRT
  - 02/12/2019: Acknowledged (Outdated Intel IPP Crypto library)
  - 11/12/2019: Firmware Update for Intel Management Engine



Coordinated Disclosure - Intel

- Intel (CVE-2019-11090)
  - 02/01/2019: Reported to IPSIRT
  - 02/12/2019: Acknowledged (Outdated Intel IPP Crypto library)
  - 11/12/2019: Firmware Update for Intel Management Engine

- Intel IPP CVEs (MicroWalk)
  - CVE-2018-12155
  - CVE-2018-12156