Comp 590-184: Hardware Security and Side-Channels

Lecture 15: Trusted Execution Environments

March 10, 2026

Andrew Kwong

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Slides adapted from Mengjia Yan

Outline

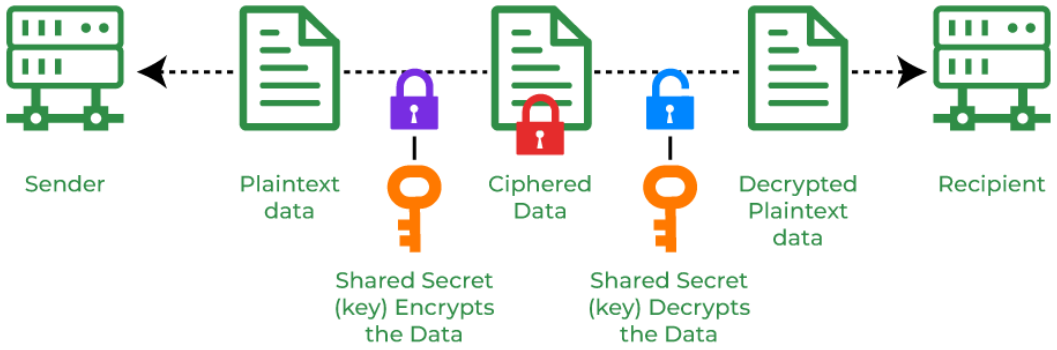
- Finish HSMs
- Trusted Execution Environments (TEE)

Hardware Security Modules

- Hardware can offer security primitives we cannot achieve with only software

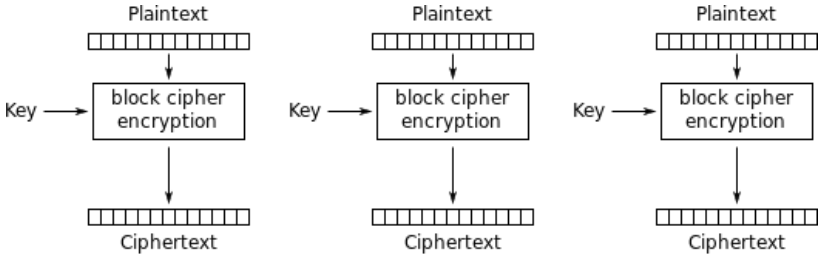
Symmetric Cryptography

- Encryption key and decryption key are the same

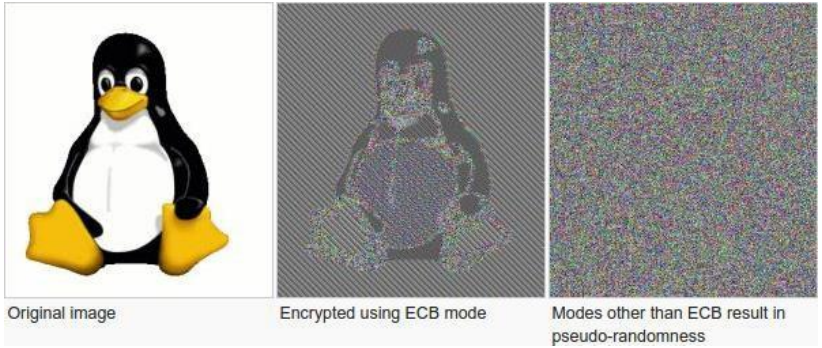


Block ciphers (e.g., DES, AES)

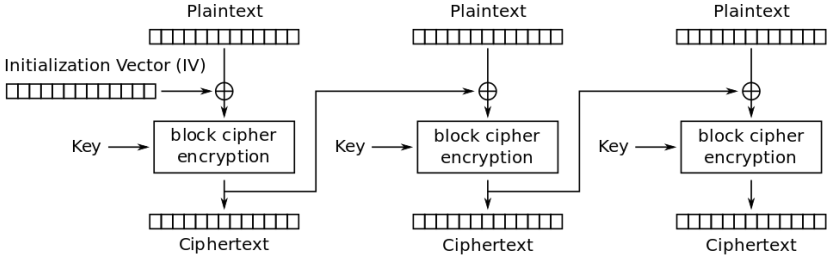
- Divide data in blocks and encrypt/decrypt each block
- Fixed length input/output (e.g. 256bit input and 256bit output
- **ECB IS NOT RECOMMENDED**



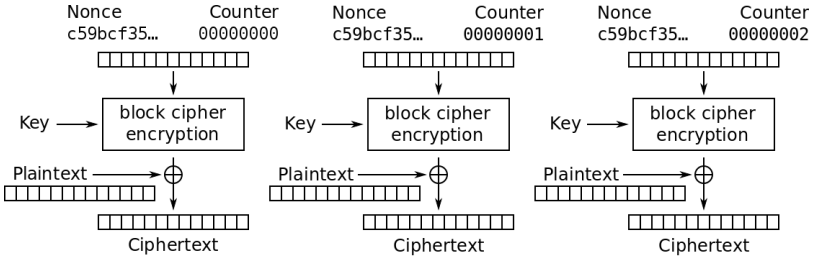
Electronic Codebook (ECB) mode encryption


Original image | Encrypted using ECB mode | Modes other than ECB result in pseudo-randomness

Other Block cipher Modes



Cipher Block Chaining (CBC) mode encryption



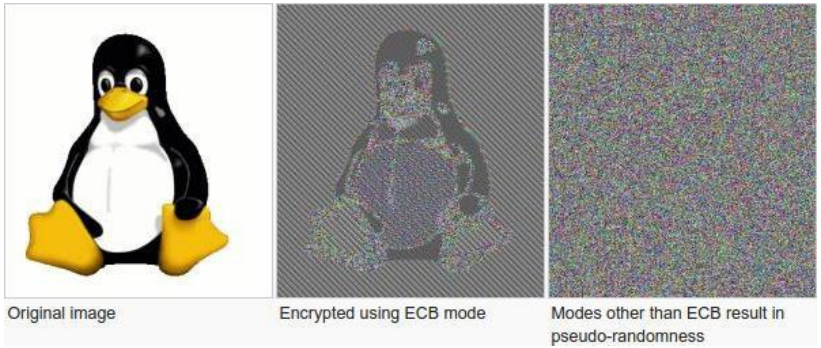
Counter (CTR) mode encryption

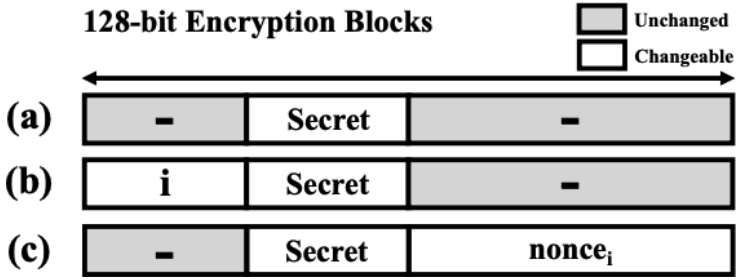
IV can be public, but need to ensure to not reuse IV for the same key.

Use cases: file/disk encryption and memory encryption.

Use Correct Crypto Primitives

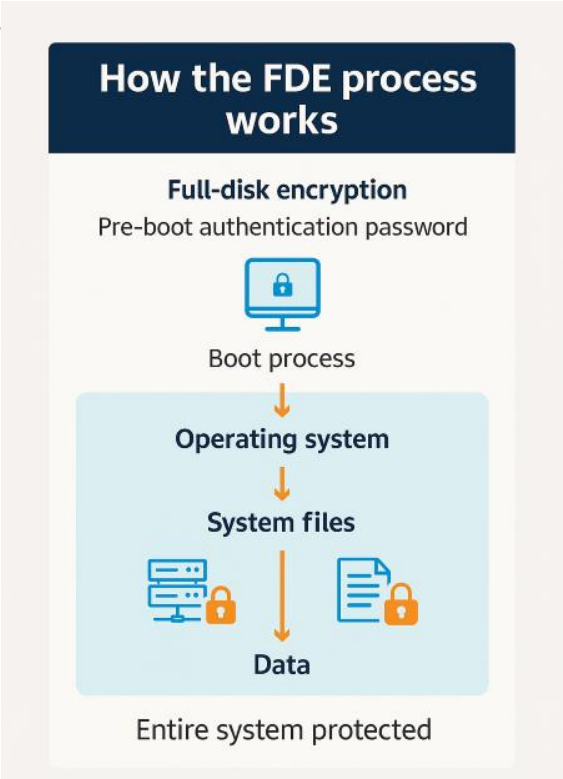
- Ciphertext Side Channels on AMD SEV
- SEV's memory encryption engine uses an XOR-Encrypt-XOR (XEX) mode -> deterministic encryption during the lifetime of a VM





Li et al, CIPHERLEAKS: Breaking Constant-time Cryptography on AMD SEV via the Ciphertext Side Channel, USENIX'21
Li et al, A Systematic Look at Ciphertext Side Channels on AMD SEV-SNP, S&P'22

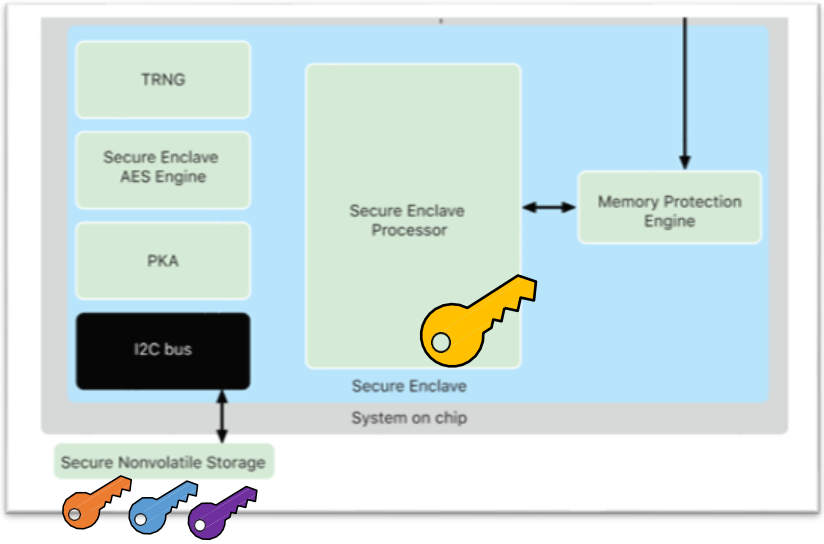
Disk Encryption



Encrypt using Short Passcode

- How many attempts do we need to brute-force 6-digit passcode?
- How to mitigate brute-force?
- How to deal with attacks who can copy the data across devices and brute-force in parallel?

Bind Crypto Keys to Device



User data encryption **keys**

A unique ID (**UID**) root cryptographic key.

- Unique to each device
- Randomly generated
- Fused into the SoC at manufacturing time
- Not visible outside the device
- BIG!

`Passcode + UID -> encryption entropy`

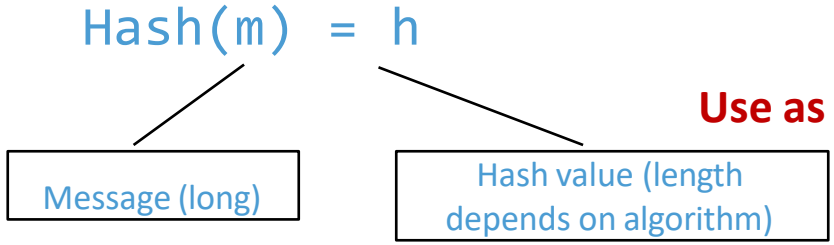
Brute-force has to be performed on the device under attack

Combine with other mitigations:

- Escalating time delays
- Erase data when exceeding attempt count

Real-world use case

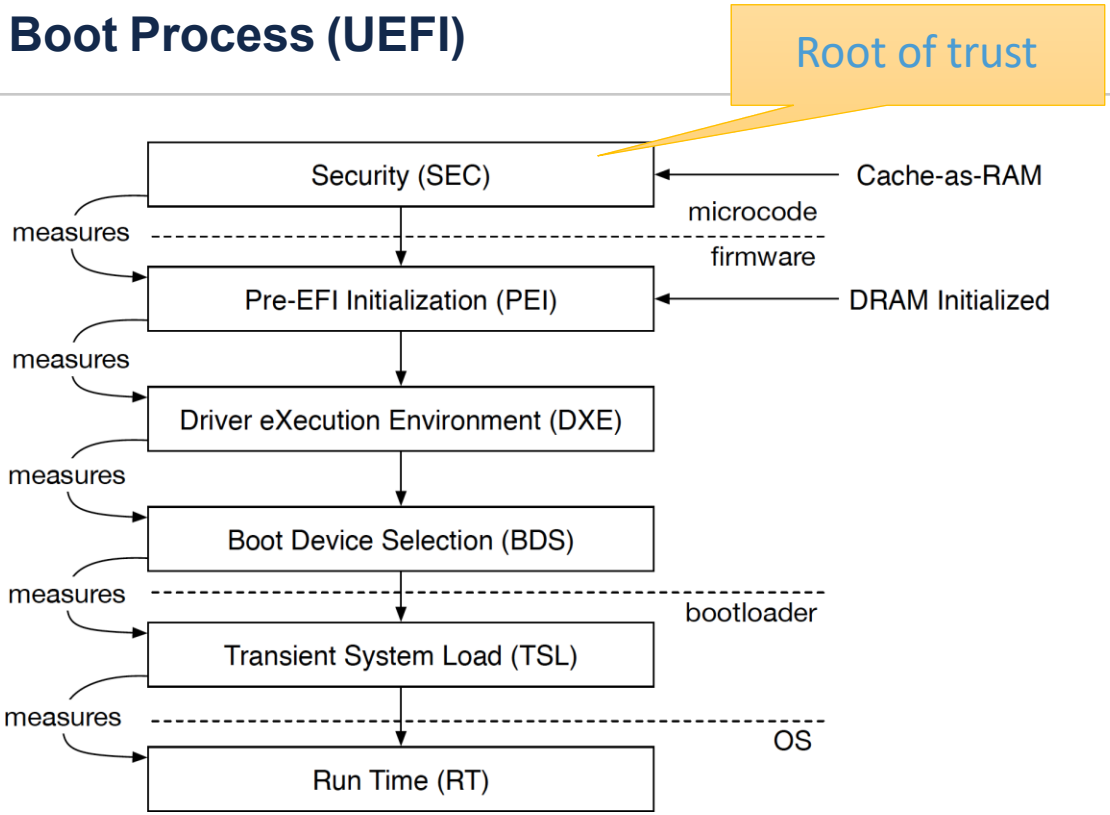
Integrity (Hashing)



Use as fingerprints

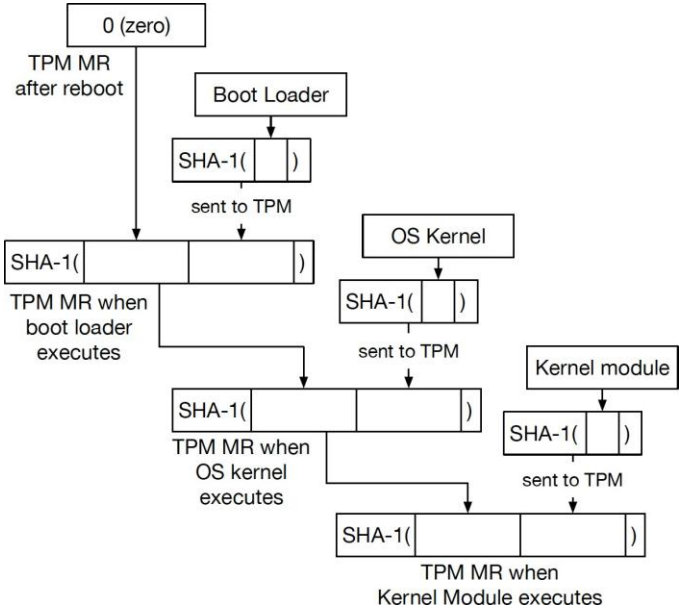
- One-way hash
  - Practically infeasible to invert, and difficult to find collision
- Avalanche effect
  - "Bob Smith got an A+ in ELE386 in Spring 2005"→ 01eace851b72386c46d
  - "Bob Smith got an B+ in ELE386 in Spring 2005"→ 936f8991c111f2cefaw
- When message is long
  - Divide message into blocks, and keep extending the hash by adding previous hash

Boot Process (UEFI)



Always measure before executing …

Secure Boot using TPM



Did you enable TPM 2.0 but keep getting prompted in *Call of Duty*? It's possible your motherboard requires a BIOS firmware update. **Learn more about updating BIOS firmware**.

The **Trusted Platform Module 2.0 (TPM 2.0)** is a technology that provides hardware-based security features on PCs operating Windows.

Secure Boot is another system-level feature that helps protect against low-level cheats by ensuring only trusted software loads during your PC's startup. Enabling **Secure Boot** alongside **TPM 2.0** provides an added layer of protection and is recommended for the best, most secure ***Call of Duty***® experience.

PC players without **TPM 2.0** enabled may receive an in-game notification indicating their system does not meet the new security requirements when launching applicable ***Call of Duty*** games.

Both security features were added to *Call of Duty* with the launch of Season 05, in August 2025.

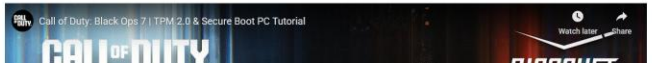
Both **TPM 2.0** and **Secure Boot** are required to play *Call of Duty: Black Ops 7* and *Call of Duty: Warzone*, however, neither are required to play other currently available *Call of Duty* titles. Keeping these settings enabled ensures a fair and fun experience for all players

Note: PCs using Windows 11 likely already have both security features enabled, as they are required for the operating system. Windows 10 systems will require updates if TPM is disabled or if a legacy TPM version (1.0 - 1.2) is enabled. While TPM 2.0 requires Windows 10 version 20H2 or later, Call of Duty *requires Windows 10 version 22H2 or later.*

Important! This article offers general guidance, and some steps may differ depending on your PC.

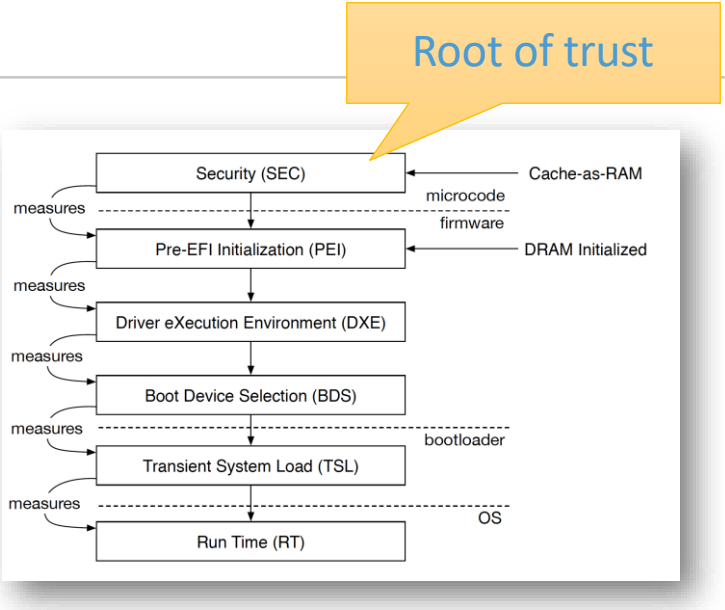
If you're not familiar with accessing and updating your UEFI/BIOS settings, you should reach out to your hardware manufacturer's customer support or a professional for assistance. Changing UEFI/BIOS settings improperly can cause system issues, including boot failures.

We also strongly recommend reviewing your PC and motherboard manufacturer's manuals and support resources before making any changes to your UEFI/BIOS settings. Activision is not responsible for changes made to your UEFI/BIOS settings.



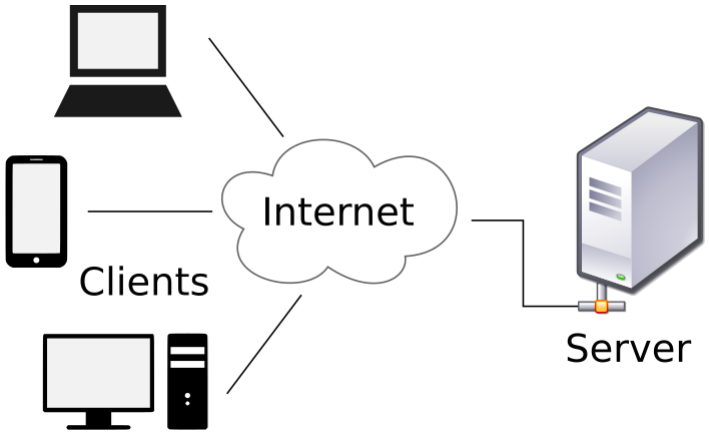
Security Problems of Using TPM

- Assume the first-stage bootloader is securely embedded in motherboard
- Not easy to use with frequent software/kernel update
- TPM Reset attacks
  - exploiting software vulnerabilities and using software to report false hash values



Han et al. A Bad Dream: Subverting Trusted Platform Module While You Are Sleeping. Usenix Security'18 Wojtczuk et al. Attacking Intel TXT® via SINIT code execution hijacking. 2011

Security Context #3



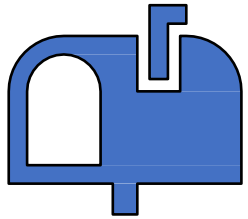
a) A remote server wants to trust an end-user, e.g., when joining a company's highly-secure network.

b) A device wants to update/install an new version of OS/software approved by the vendor

-> **Authentication and establishing trust**

Asymmetric Cryptography

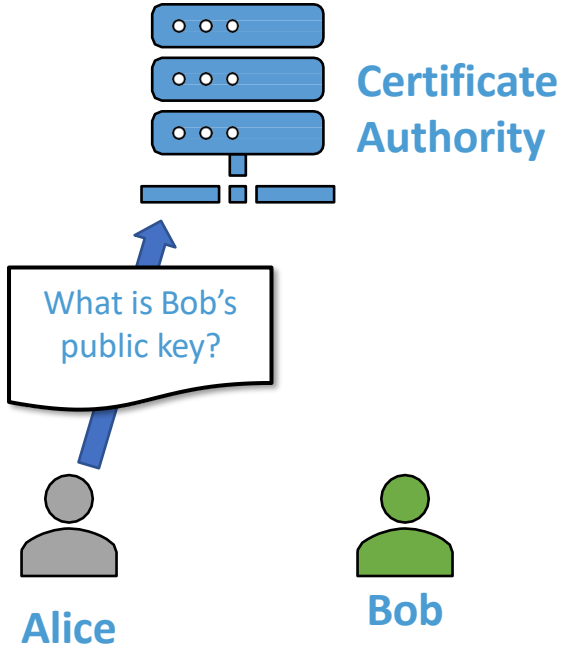
- A pair of keys:
  - Private key ($K_{private}$ – kept as secret)
  - Public key (K_{public} – safe to release publicly)
- Computation:
  - `Sign(plaintext,` $K_{private}$`) = signature`
  - `Verify(plaintext, signature,` K_{public}`) = T/F`
- How to announce and obtain the public key?



Mail box is public;
Box key is private

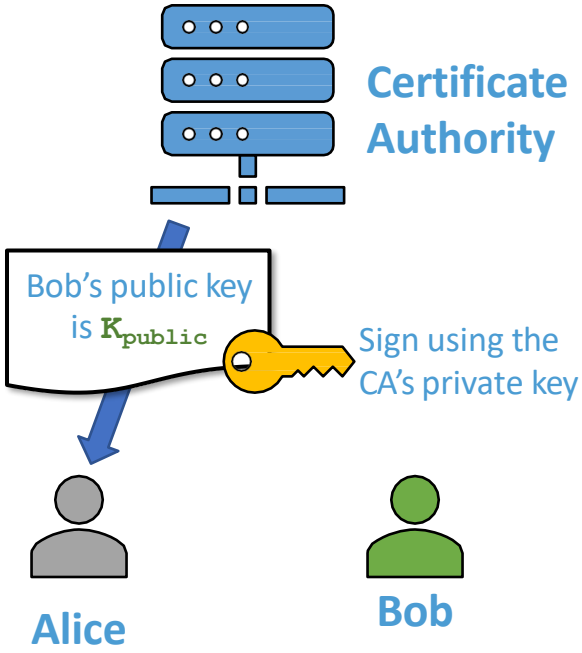
Public Key Infrastructures (PKIs)

- Analogy: public key is like a government-issued ID, need to be validated by an authority and tied to a private key.
- Bob has a private key $K_{private}$ and wants to claim he corresponds to a public key K_{public}

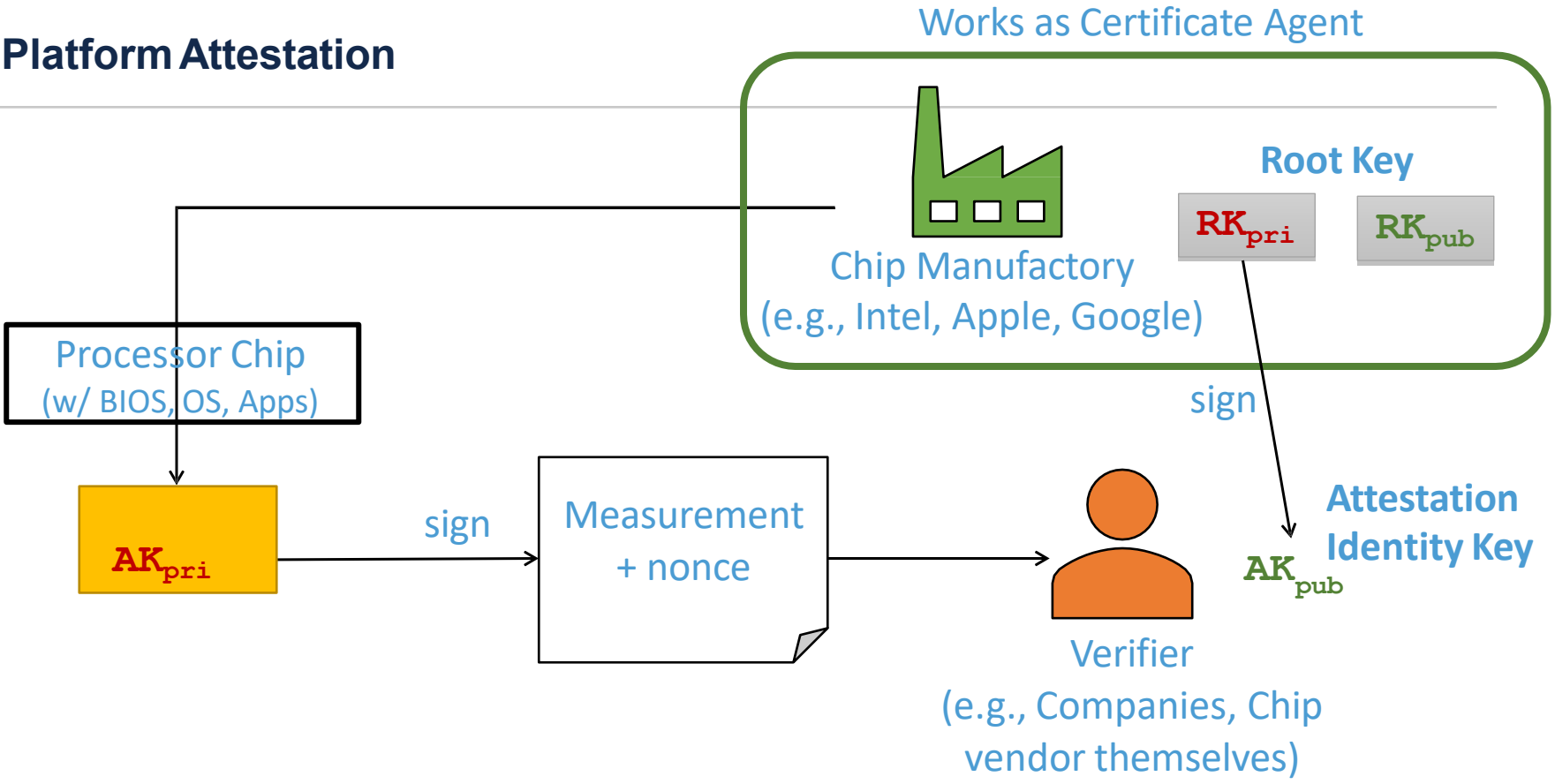


Public Key Infrastructures (PKIs)

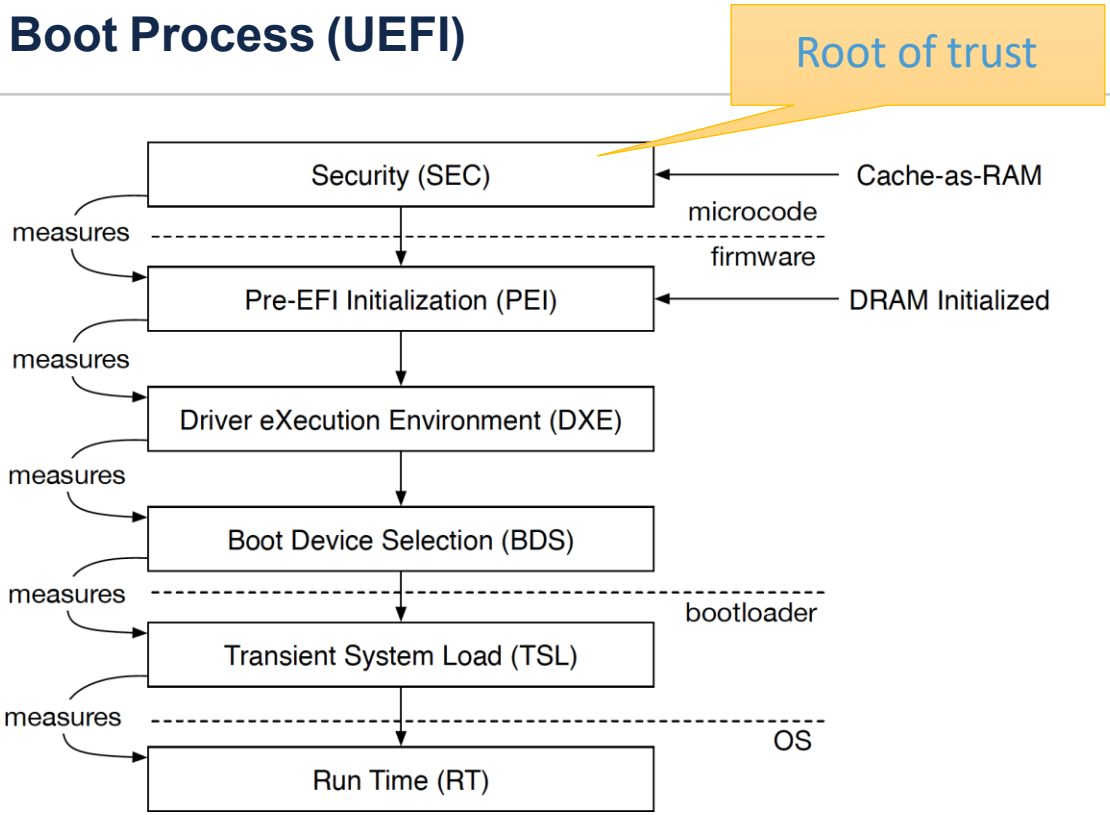
- Analogy: public key is like a government-issued ID, need to be validated by an authority.
- Bob has a private key $K_{private}$ and wants to claim he corresponds to a public key K_{public}
- Establish a chain of trust
- **Real-world use cases**: identify website, identify hardware chips/processors



Platform Attestation

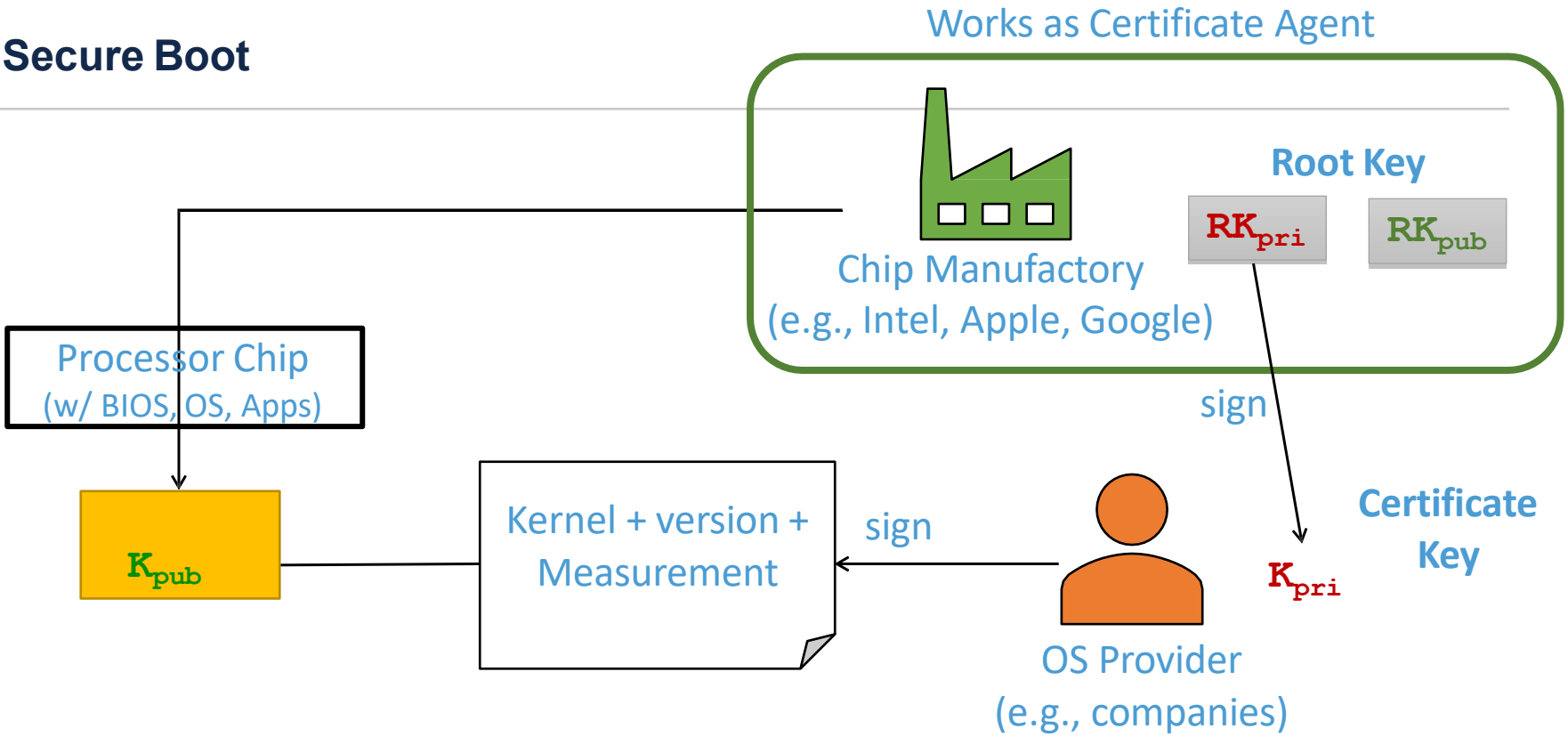


Boot Process (UEFI)



Always measure before executing …

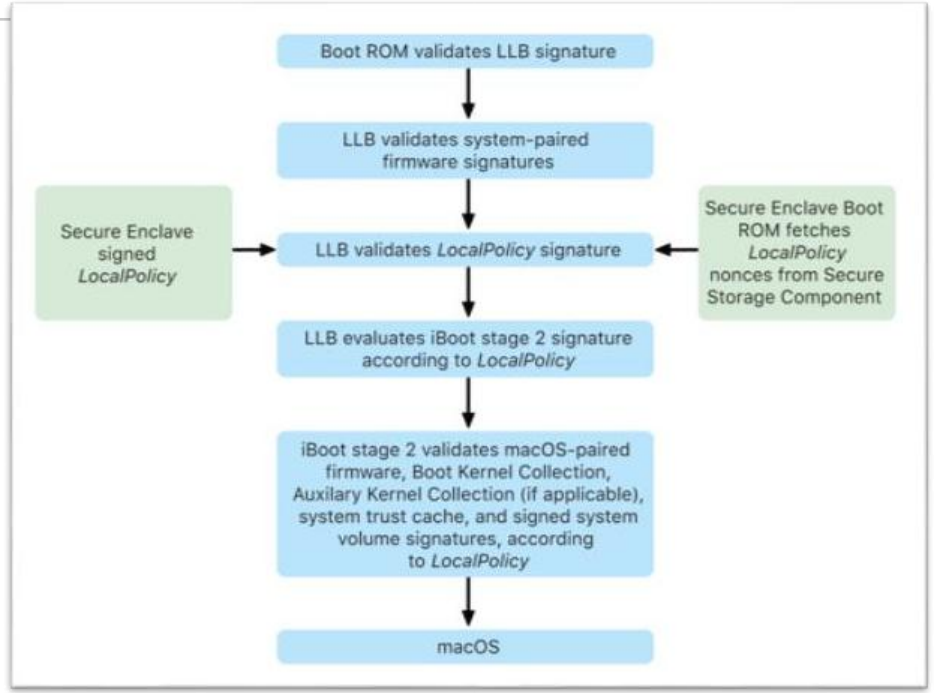
Secure Boot


Works as Certificate Agent
Root Key
RK_{pri}
RK_{pub}
Chip Manufactory
(e.g., Intel, Apple, Google)
Processor Chip
(w/ BIOS, OS, Apps)
sign
K_{pub}
Kernel + version +
Measurement
sign
Certificate
Key
K_{pri}
OS Provider
(e.g., companies)

Secure Boot on Apple

Similar to TPM but with more constraints

- Each step is signed by Apple to prevent loading non-Apple systems
- Verify more components, including operating system, kernel extensions, etc.
- Keep track of version number to prevent rolling back to older/vulnerable versions



Summary

What Can Hardware Security Modules Offer?

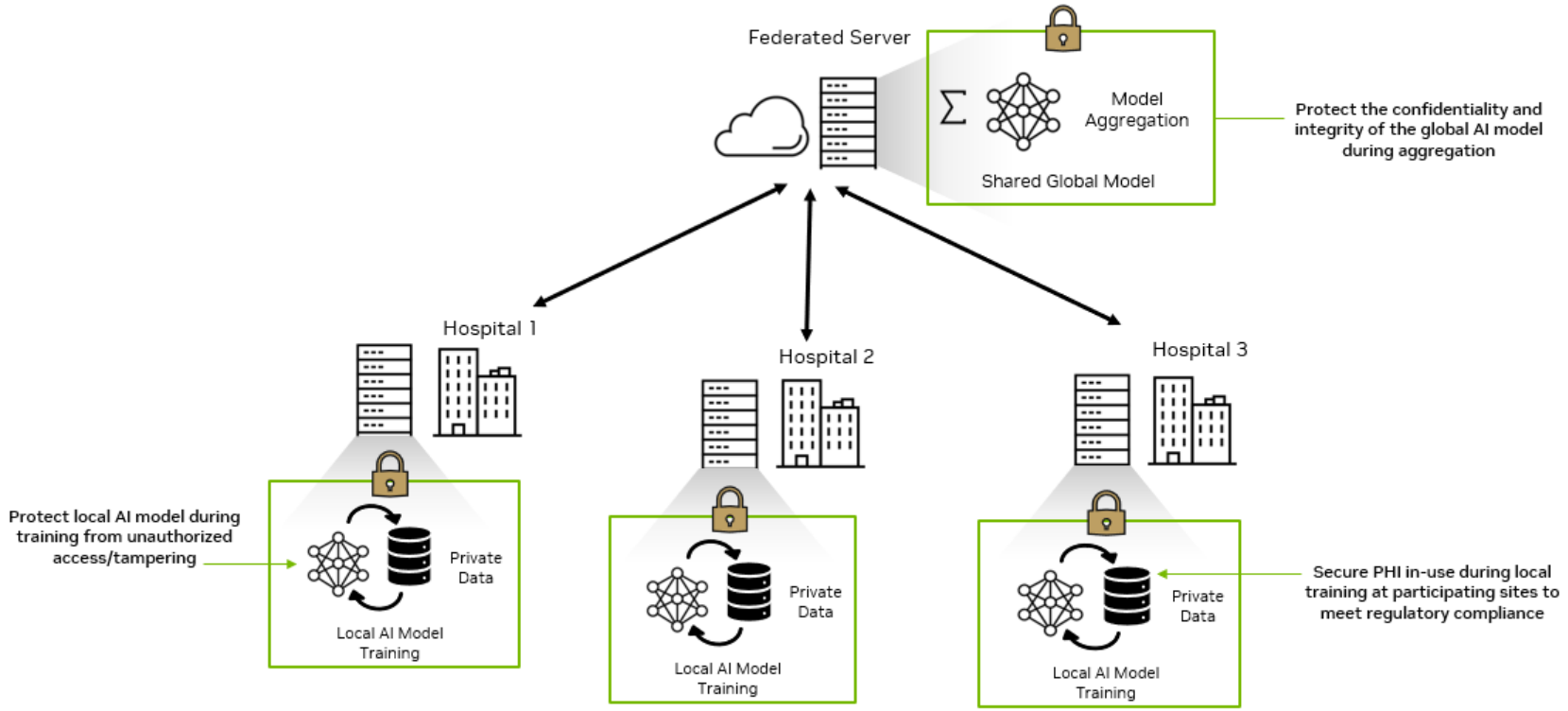
- Physical isolation
- Bind data and applications with the hardware device
- Establish root of trust

Challenges: software support. Programmability. Large TCB

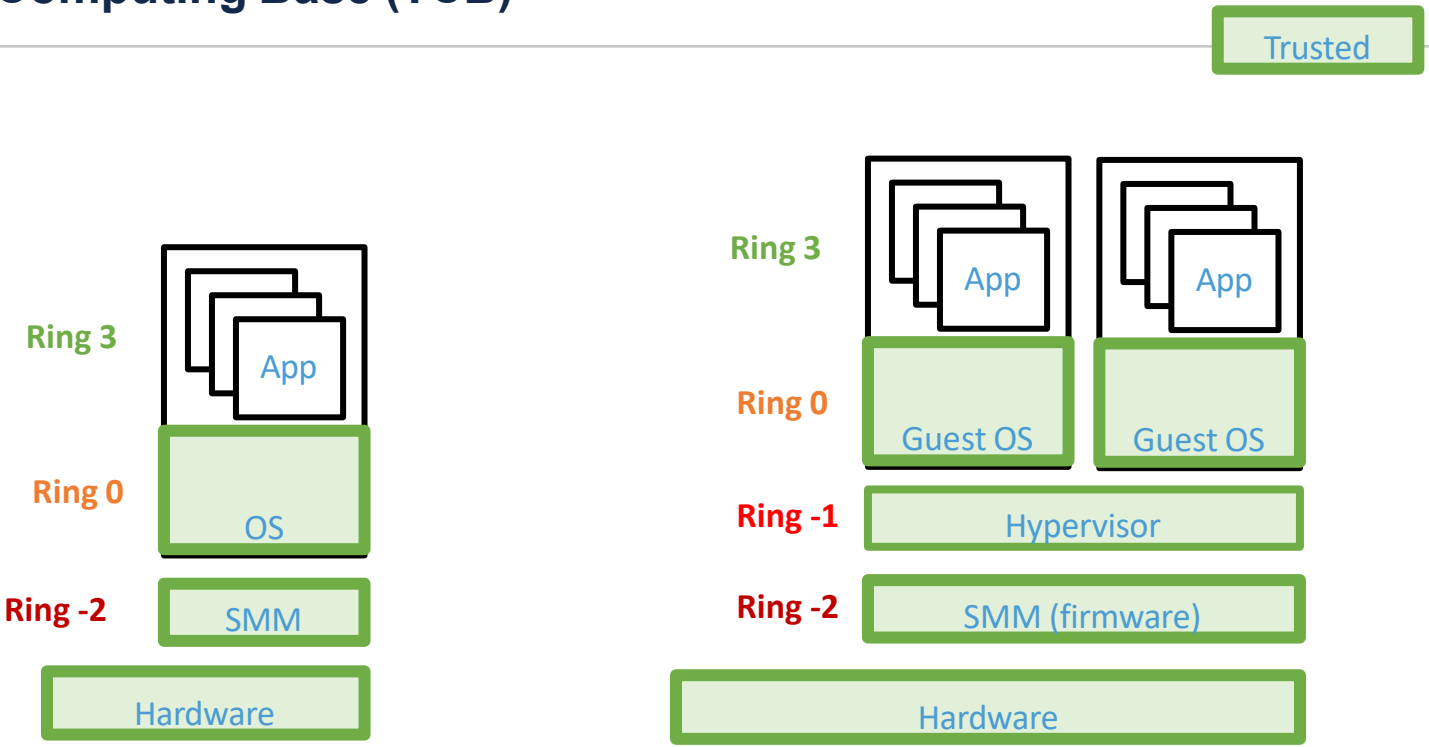
Motivation

- In 2018 hearing:
  - Can Mark view my private photos?
    - it's on Facebook's server and they need to do computations on it and process it
  - Facebook stores the data, therefore has access?

Secure AI Model Training



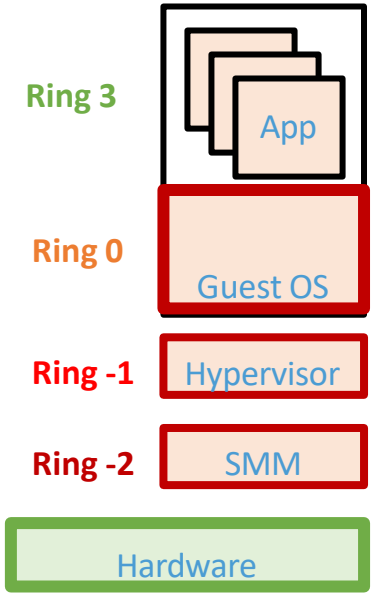
Trusted Computing Base (TCB)



Shrink TCB. Why?

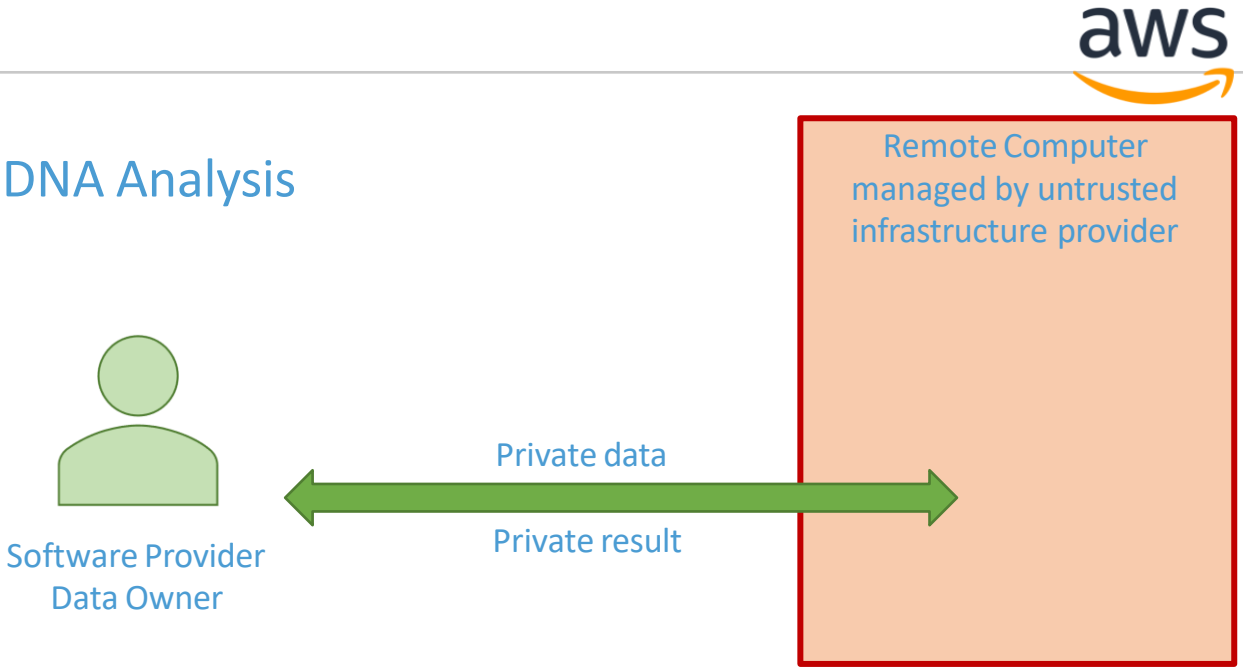
- Software bugs
  - Monolithic kernel, e.g., Linux, 30M LOC, 100+ vulnerabilities per year
  - Xen 150K LOC, 40+ vulnerabilities per year
  - SMM-based rootkits

- Remote Computing
  - Remote computer and software stack owned by an untrusted party



Secure Remote Computing

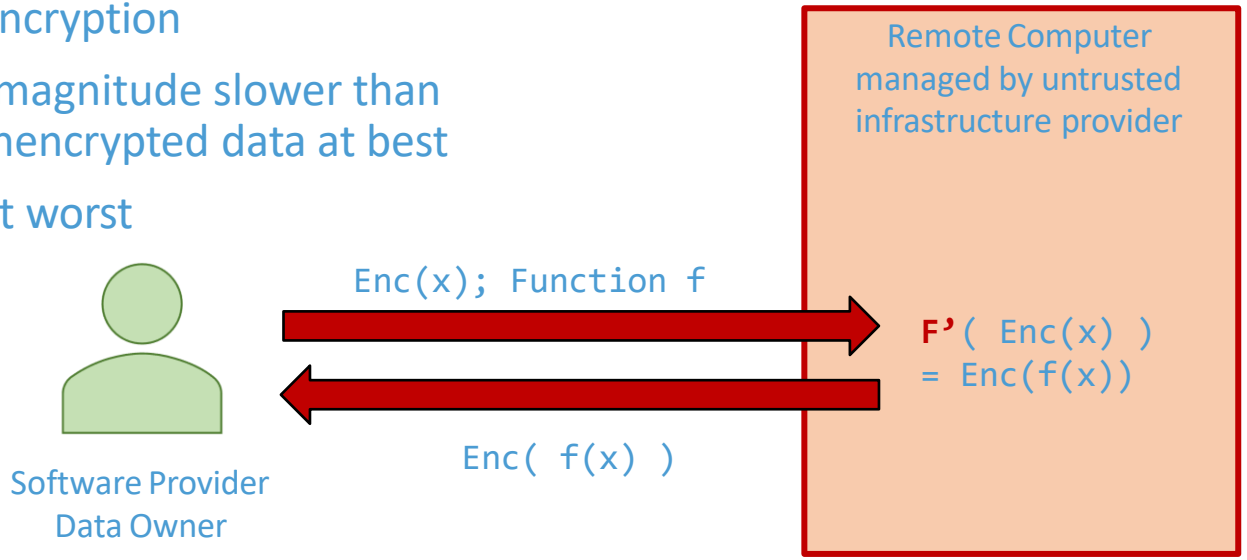
- Example: DNA Analysis



How can I keep my data private without trusting the host OS/hypervisor/SMM?

Software Solution

- Homomorphic Encryption
- **4 to 5 orders** of magnitude slower than computing on unencrypted data at best
  - Infeasible at worst

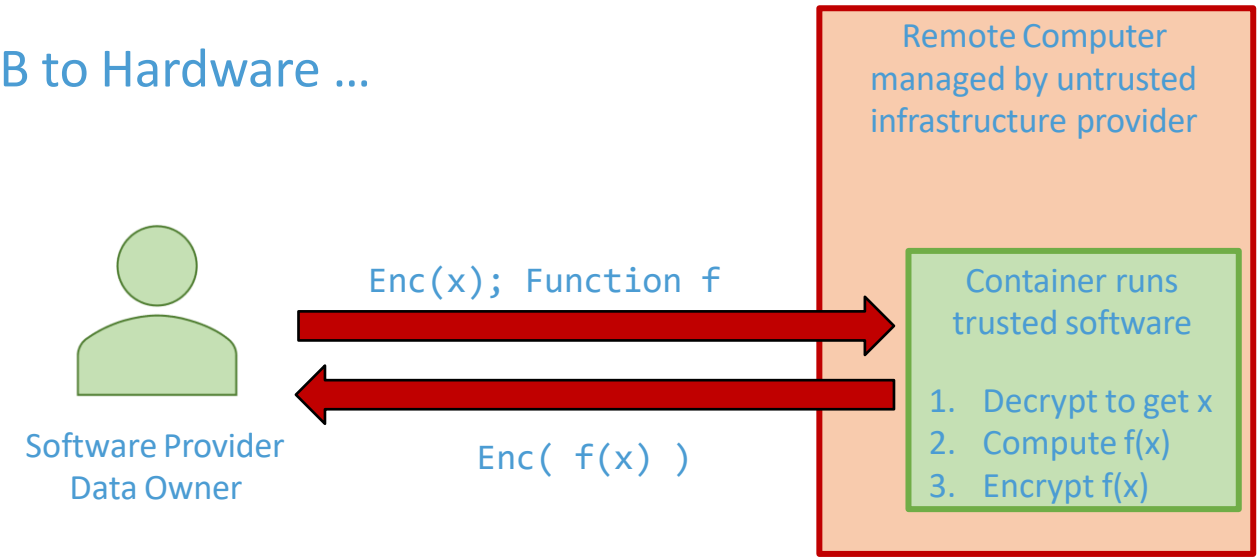


- Performance? Accelerators?

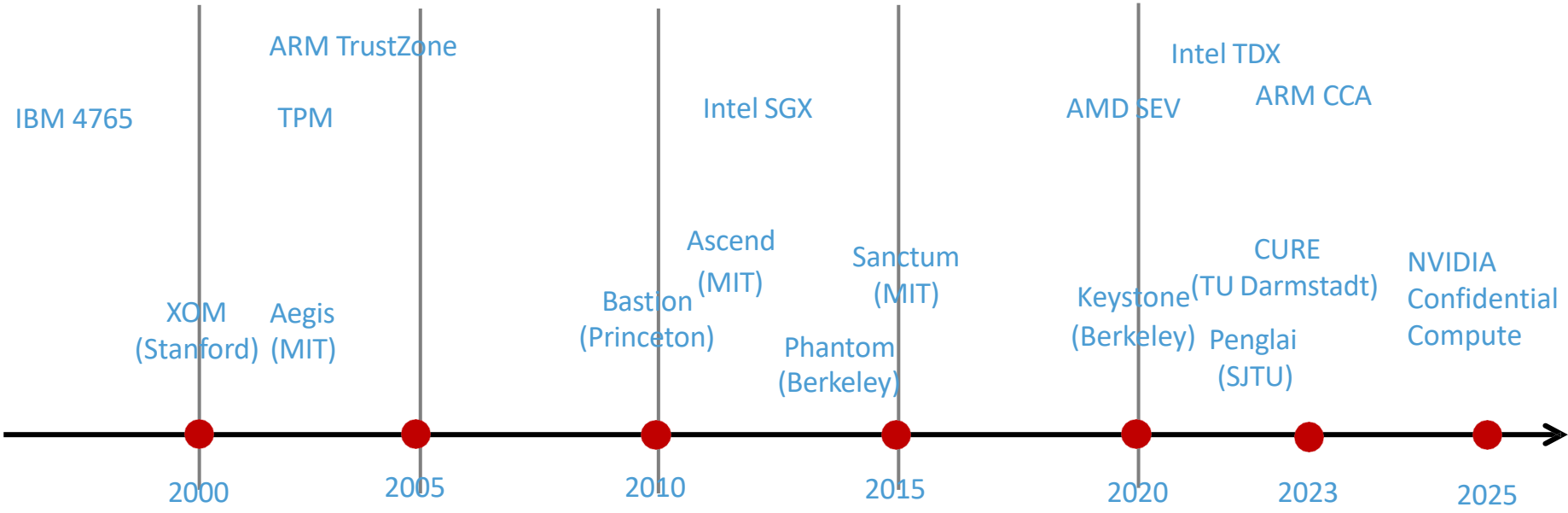
e.g., F1: A Fast and Programmable Accelerator for Fully Homomorphic Encryption; Axel Feldmann, Nikola Samardzic et al. MICRO'21

Hardware Solution

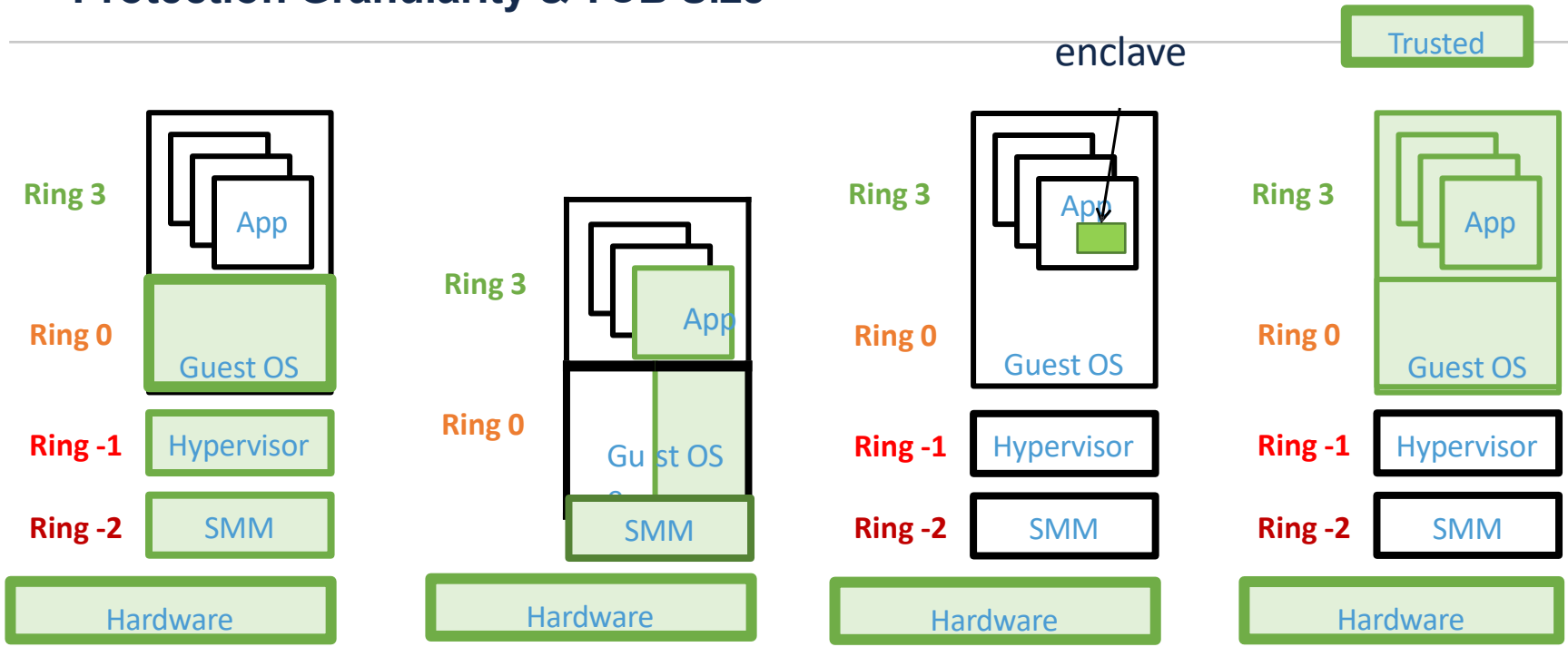
- Move TCB to Hardware …



TEE Examples



Protection Granularity & TCB Size



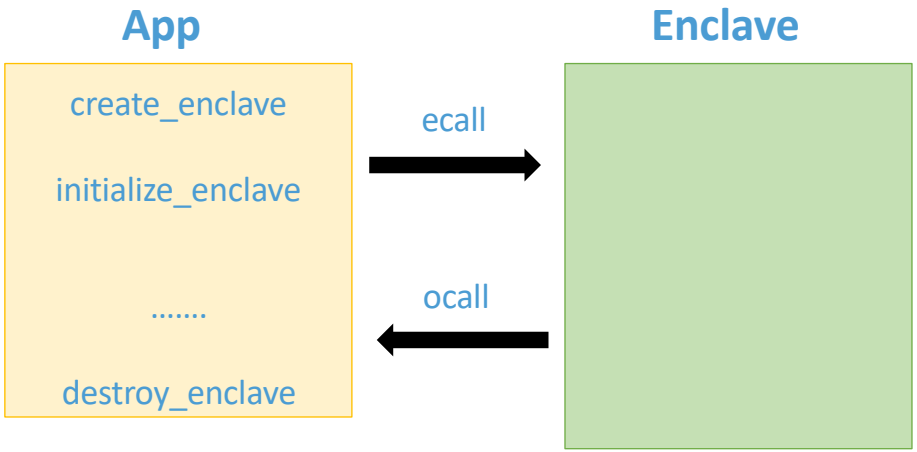
Arm TrustZone

Intel SGX

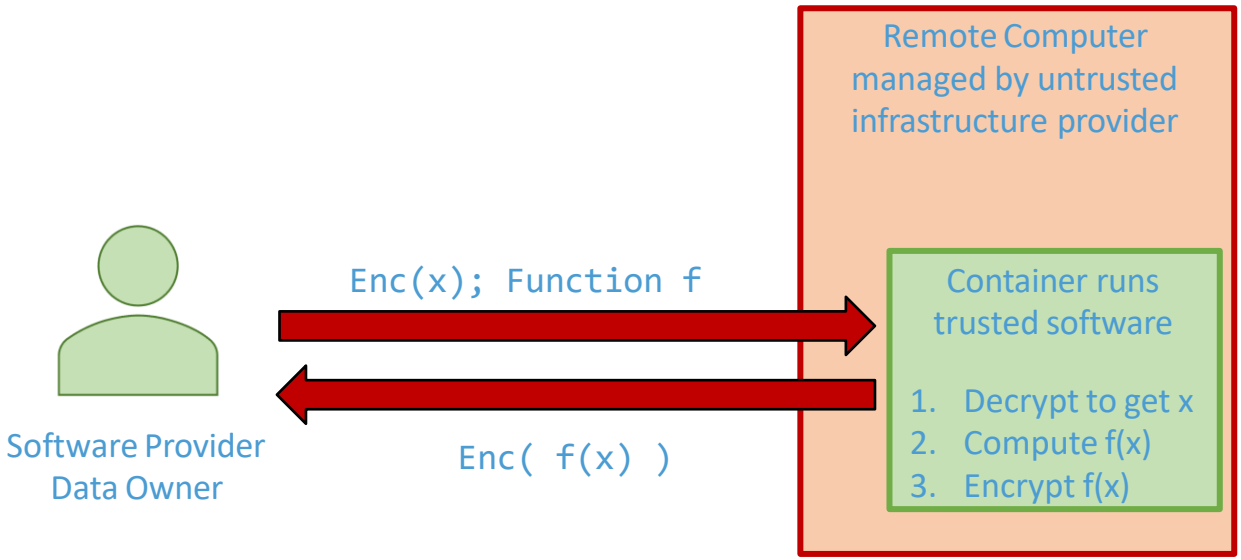
AMD SEV

SGX Enclave Programming Model

- Examples from: https://github.com/intel/linux-sgx

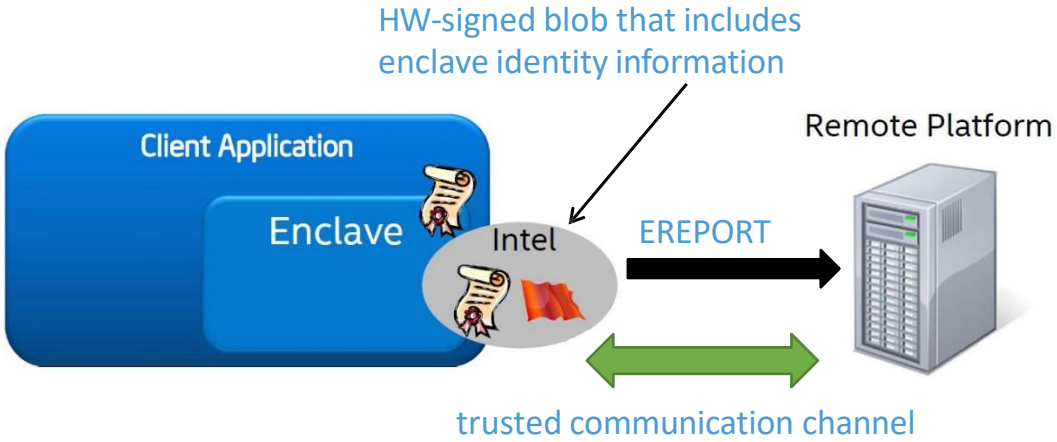


How SGX Achieves this



Remote Attestation

- HW based attestation provides proof that “this is the right application executing on an authentic platform” (approach similar to secure boot attestation)



Prove 2 things for remote attestation

- Specific piece of code is running in an SGX enclave
  - Ask hardware to use attestation key to sign a measurement that uniquely identifies the software inside the enclave
    - Confidentiality and integrity protected

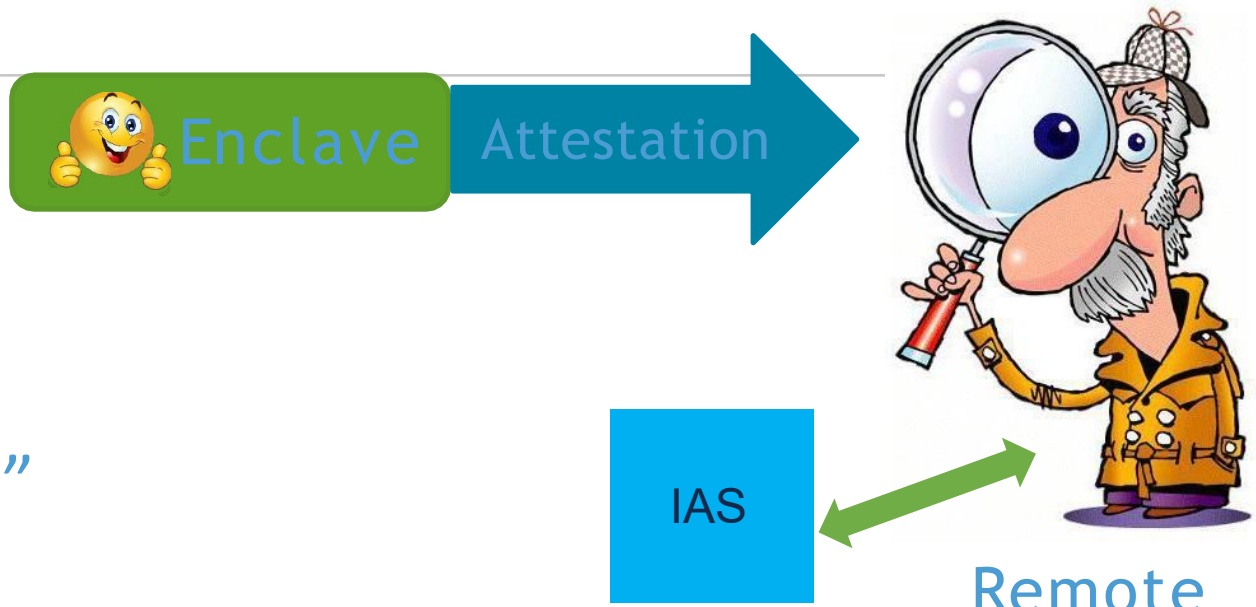
- convince a remote verifier that the attestation data was produced by that software
  - The enclave ask the hardware to sign a small piece of attestation data, producing an attestation signature.

Software Guard Extensions (SGX) Security Model

Attestation

Remote Client

Software Guard Extensions (SGX) Security Model



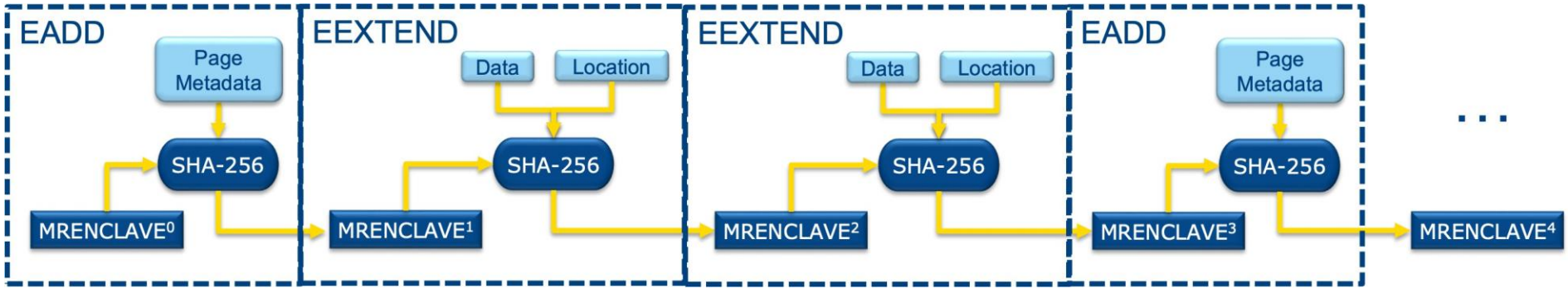
Quote includes:

- “measurement”
- data

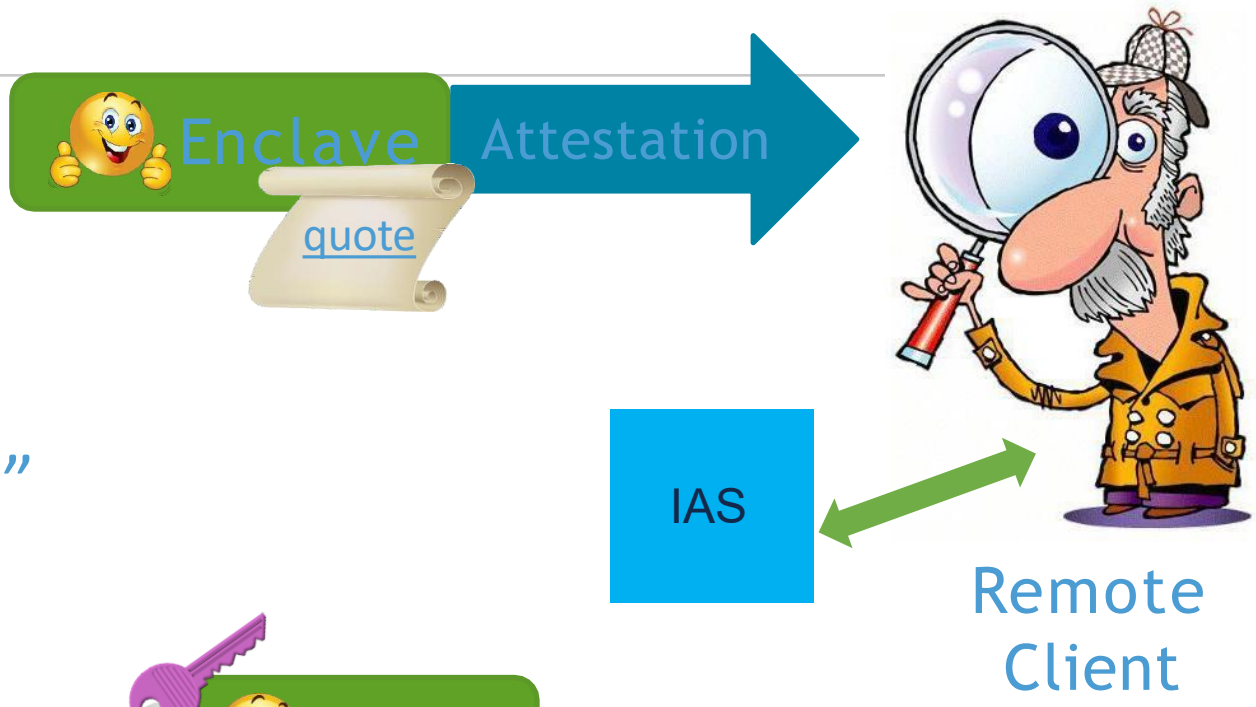
Enclave Measurement

- Hardware generates a cryptographic log of the build process
  - Code, data, stack, and heap contents
  - Location of each page within the enclave
  - Security attributes and enclave capabilities
  - Firmware version, hyperthreading

- Enclave identity (MRENCLAVE) is a 256-bit digest of the log that represents the enclave

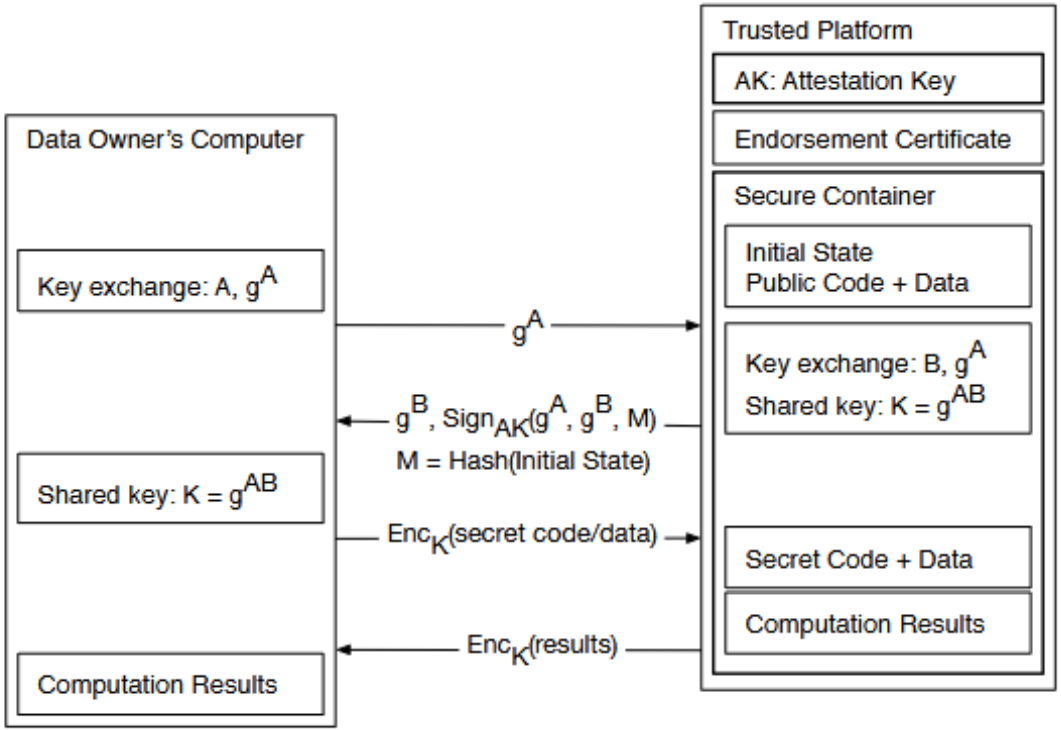


Software Guard Extensions (SGX) Security Model



Quote includes:

- "measurement"
- data


Trusted Platform
AK: Attestation Key
Endorsement Certificate
Secure Container
Initial State
Public Code + Data
Key exchange: B, g^A
Shared key: $K = g^{AB}$
Secret Code + Data
Computation Results
Data Owner's Computer
Key exchange: A, g^A
Shared key: $K = g^{AB}$
Computation Results
g^A
g^B, $Sign_{AK}(g^A, g^B, M)$
M = Hash(Initial State)
Enc_K(secret code/data)
Enc_K(results)

Secure Remote Computation Achieved!