Authenticate Everything

IoTSF Conference, 6-Dec-2016, London

The Internet of Things

21 Billion connected devices by the year 2020*, which all need to:

- Secure their own system, data and code
- Interact with each other autonomously
- Authenticate to devices, networks and services

*source: Gartner 2015

Internet of Threats


HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

2 more wireless baby monitors hacked: Hackers remotely spied on babies and parents

Major Android remote-access vulnerability is now being exploited [Updated]

Good luck getting this one patched quickly and effectively.

Medical Devices Vulnerable to Hack Attacks

ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

RISK ASSESSMENT —

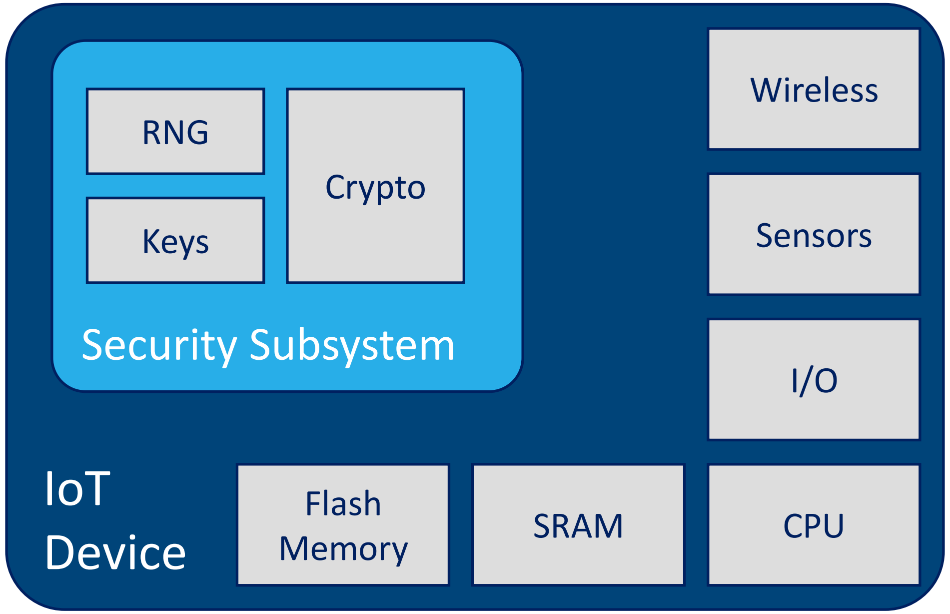
Record-breaking DDoS reportedly delivered by >145k hacked cameras

Security Subsystem

Every IoT device needs a security subsystem that:

- Protects cryptographic keys and credentials stored on the device
- Operates separately from user code / apps
- Provides authentication services
- Supports setting up secure communication channels

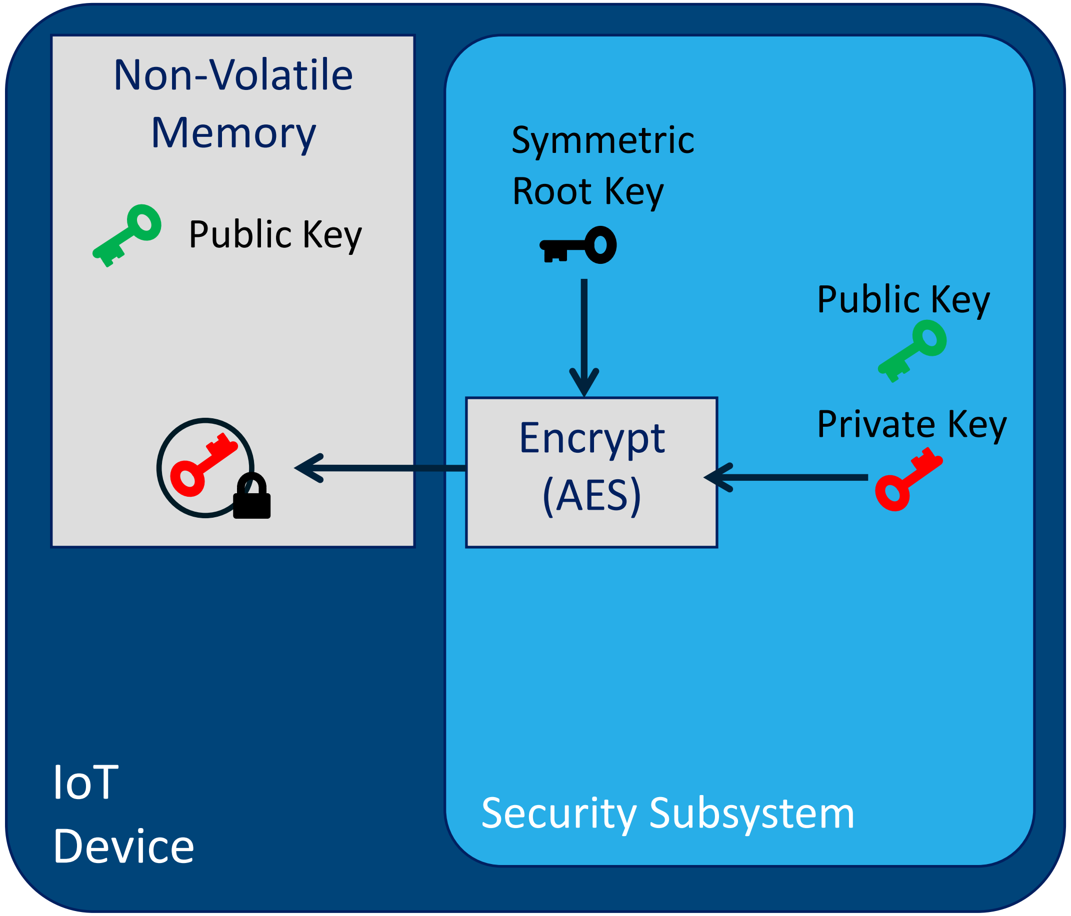
Cryptographic root key(s)



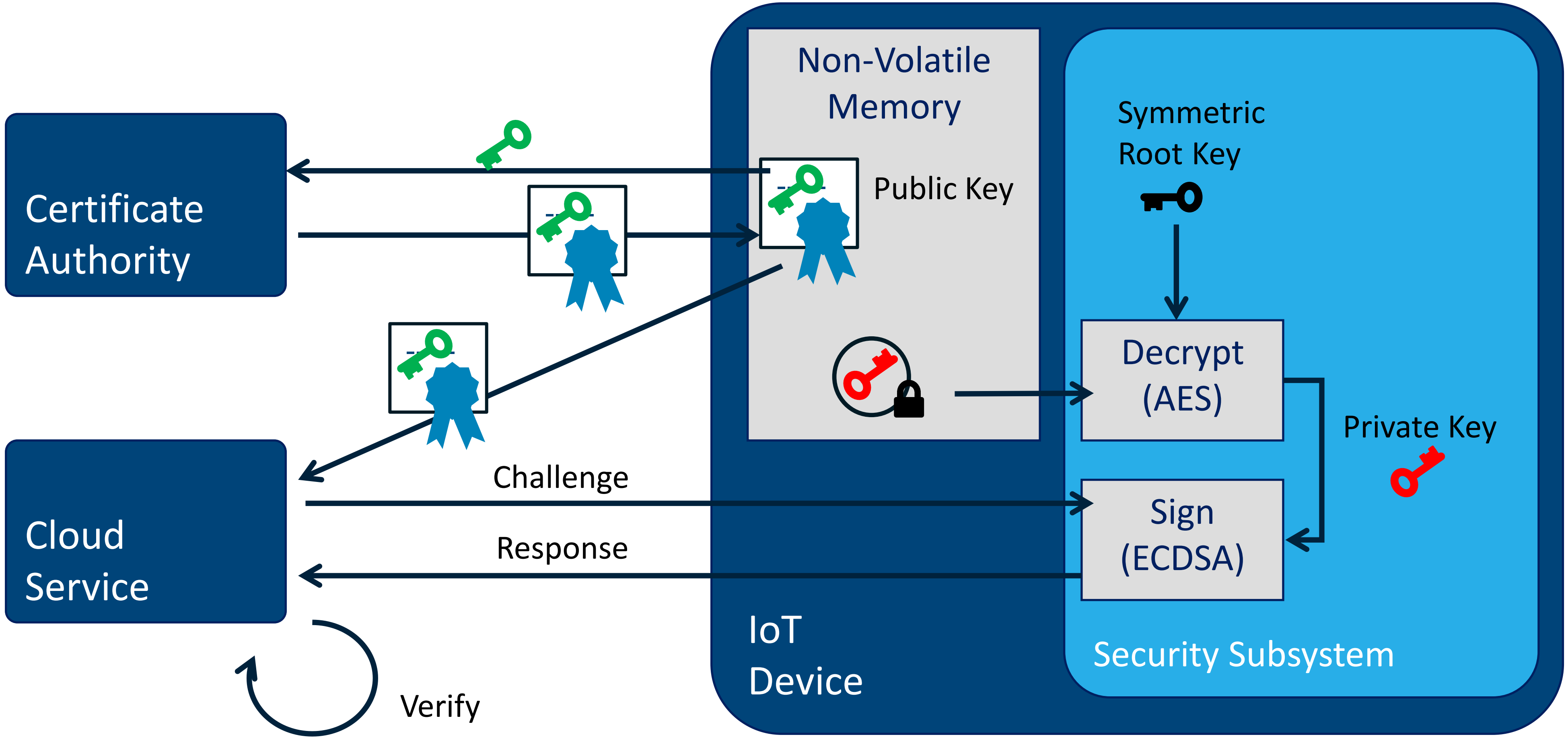
Cryptographic root key

Definition:

- Device-unique cryptographic key
- Never leaves the security perimeter of the device
- Used for encryption and authentication of secret values such as confidential data and private keys

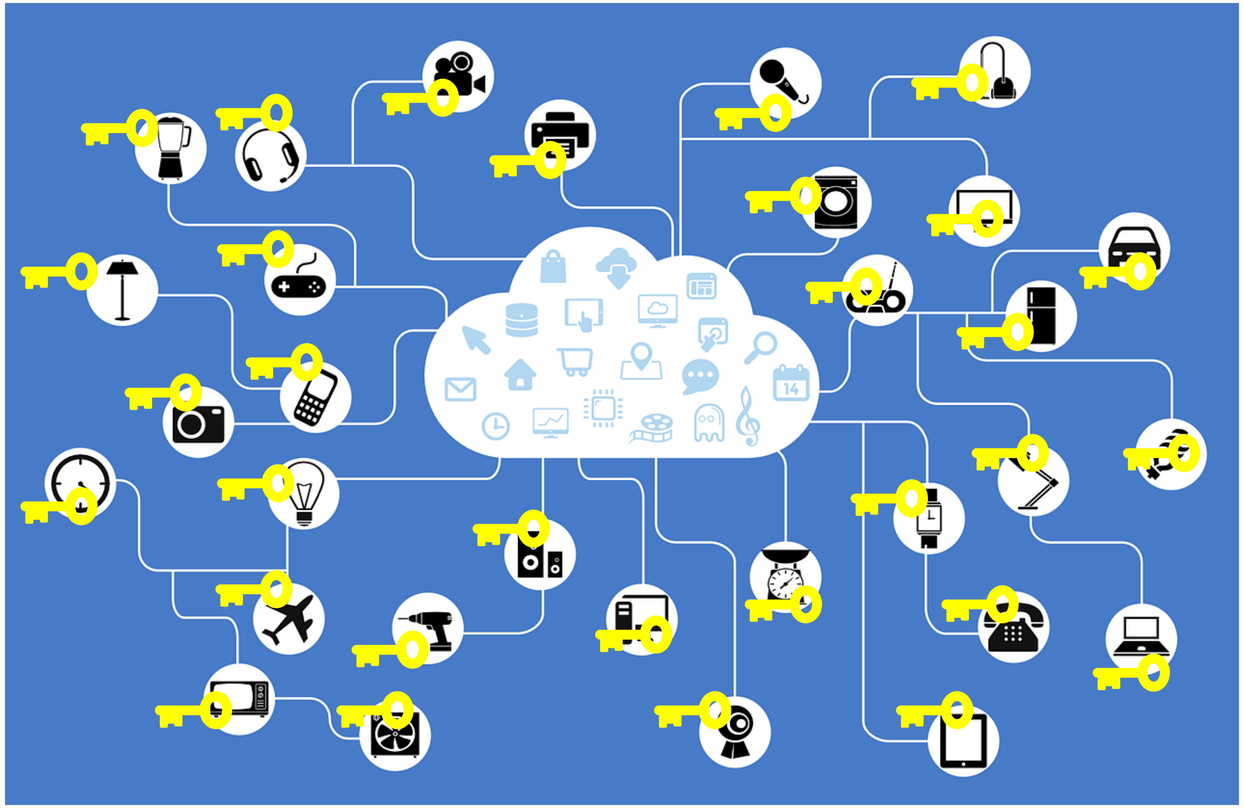


Authentication example



Problem

- Billions of IoT devices need to be provisioned with cryptographic root keys in order to bootstrap their security subsystem
- Traditional key storage provisioning methods are not capable of providing the required combination of flexibility, scalability and security



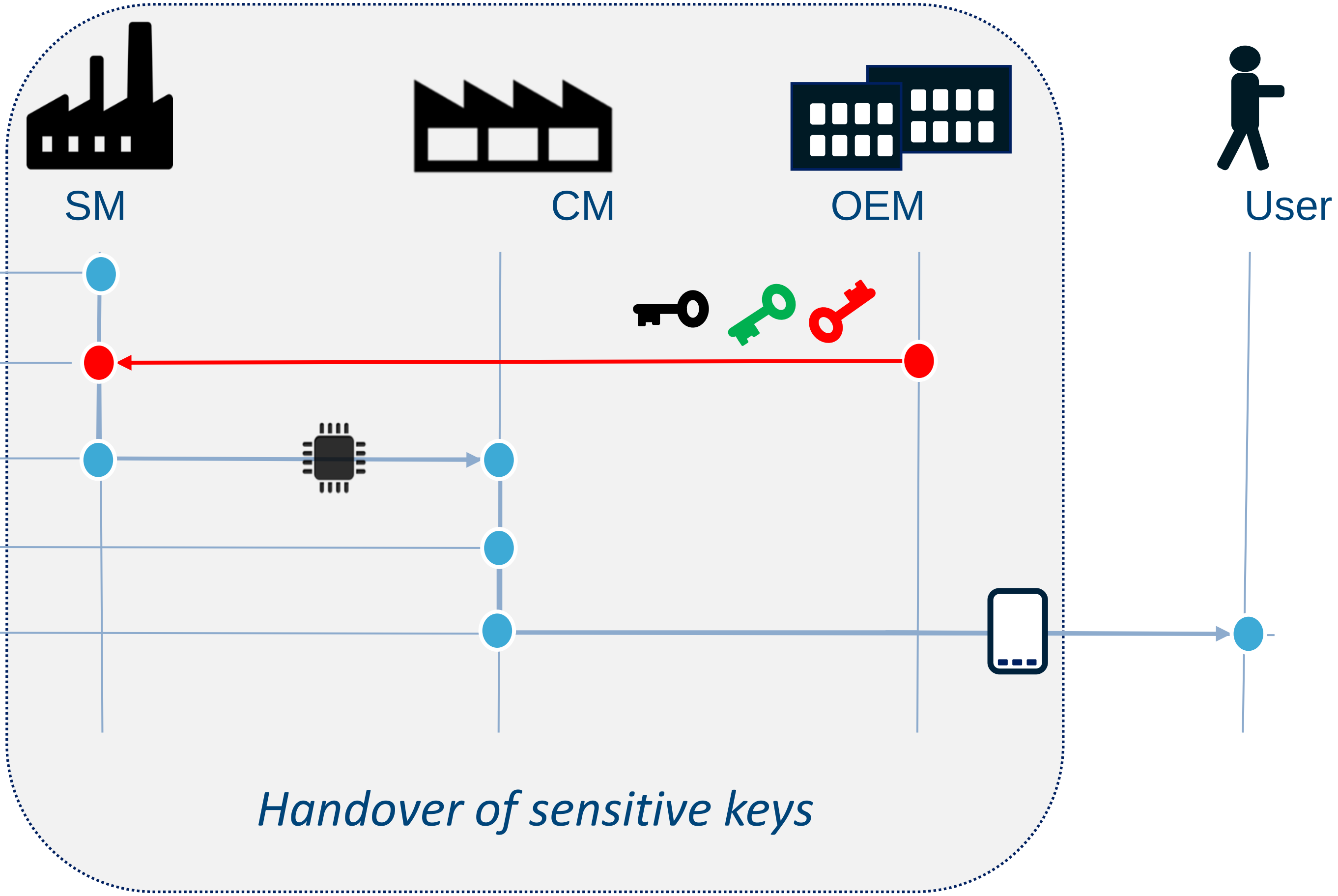
Traditional root key storage methods

- Fuses / anti-fuses
  - Need to be programmed early in production chain (e.g. by Silicon Manufacturer)
  - OEM needs to handover keys to Silicon Manufacturer → **undesired liabilities** and **increased handling costs**

- EEPROM / Flash
  - Can be programmed in later stages of the production chain
  - Typically intended for application code storage and therefore easy accessible by CPU → **low security**
  - Dedicated protected flash is prohibitively expensive for many types of devices

Example of traditional key provisioning flow

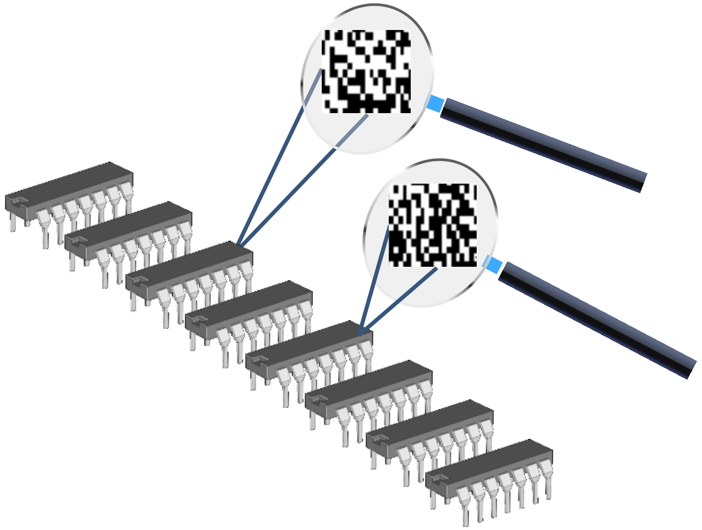
1. IC Production
2. Root key provisioning
3. IC Distribution
4. Contract Mfg.
5. Device Shipping



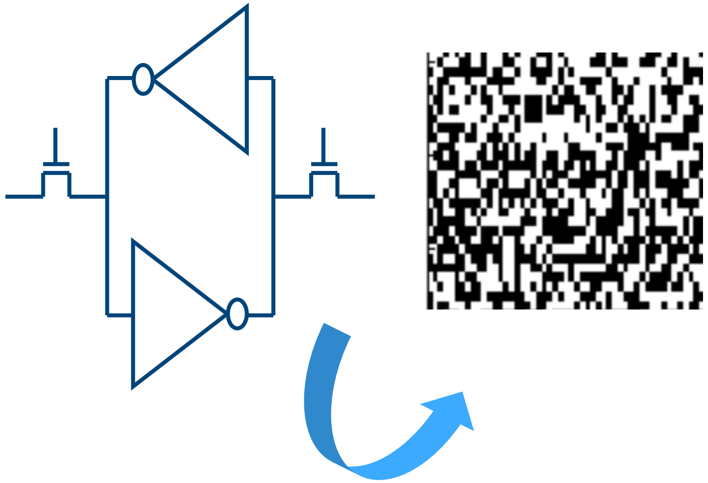
Solution requirements

- Secure storage of keys
- Flexible provisioning
- Low cost & small footprint
- Deployable from low-end to high-end devices

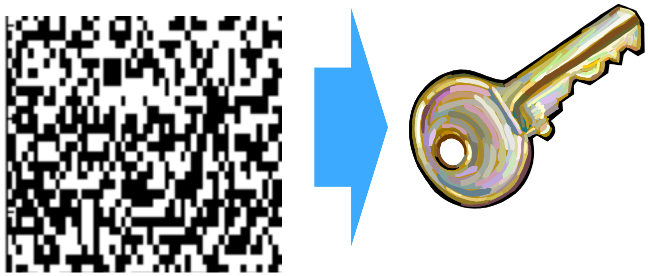
SRAM-PUF Technology



Uncontrollable process variations at manufacturing result in small variations of transistor properties, making every chip unique

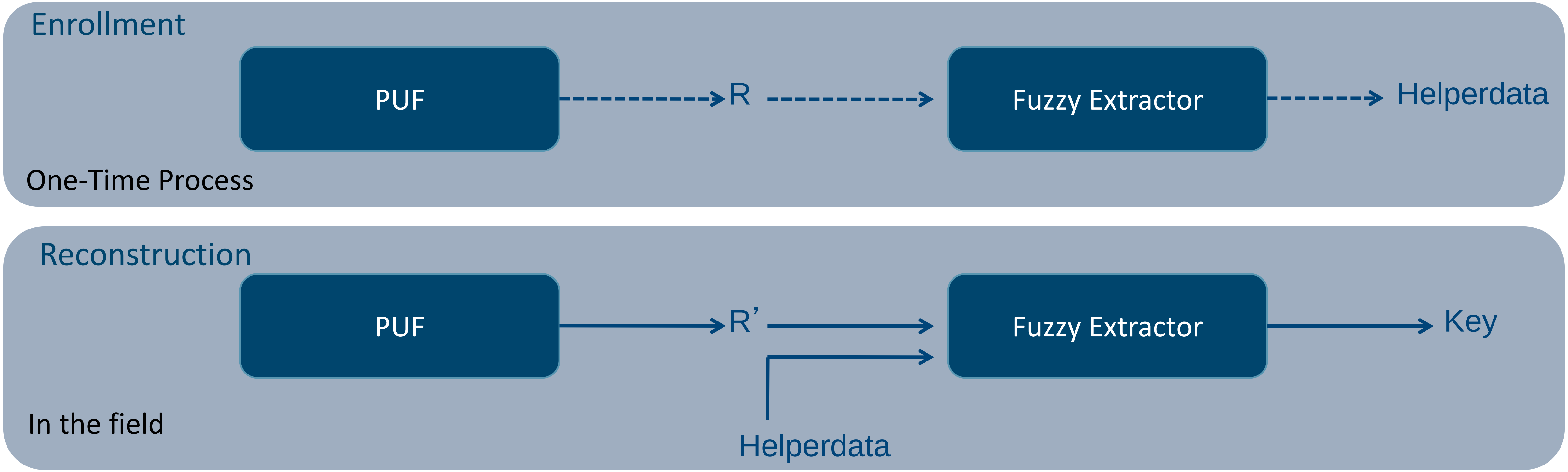


SRAM startup (PUF) values establish a unique but slightly noisy fingerprint



The fingerprint is turned into a chip-unique cryptographic root key

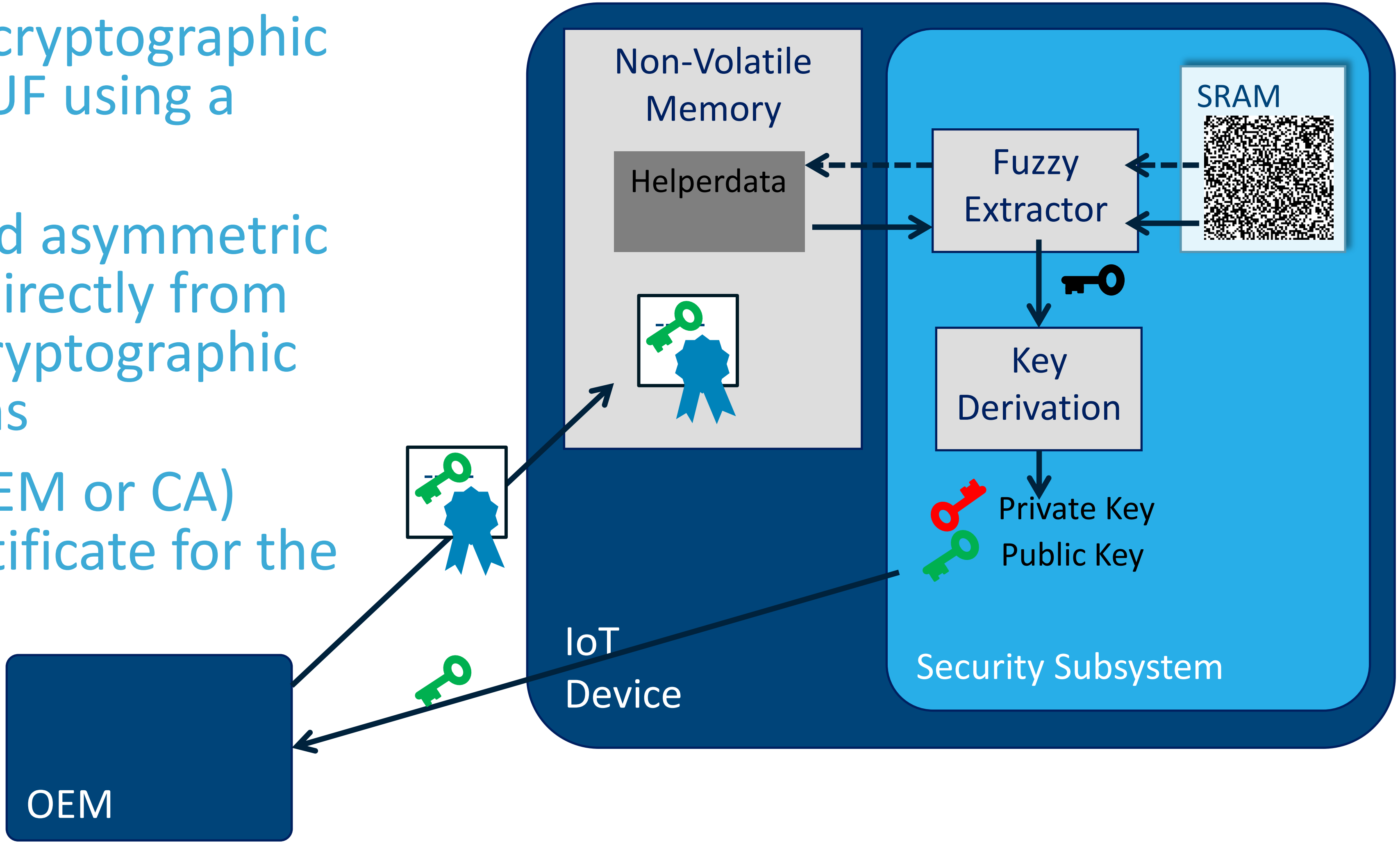
Fuzzy Extractor



$I(\text{Helperdata},\text{Key})<\varepsilon$ $\quad$ $P[\text{Key not Correct}]<\delta$

Root key storage solution based on PUF

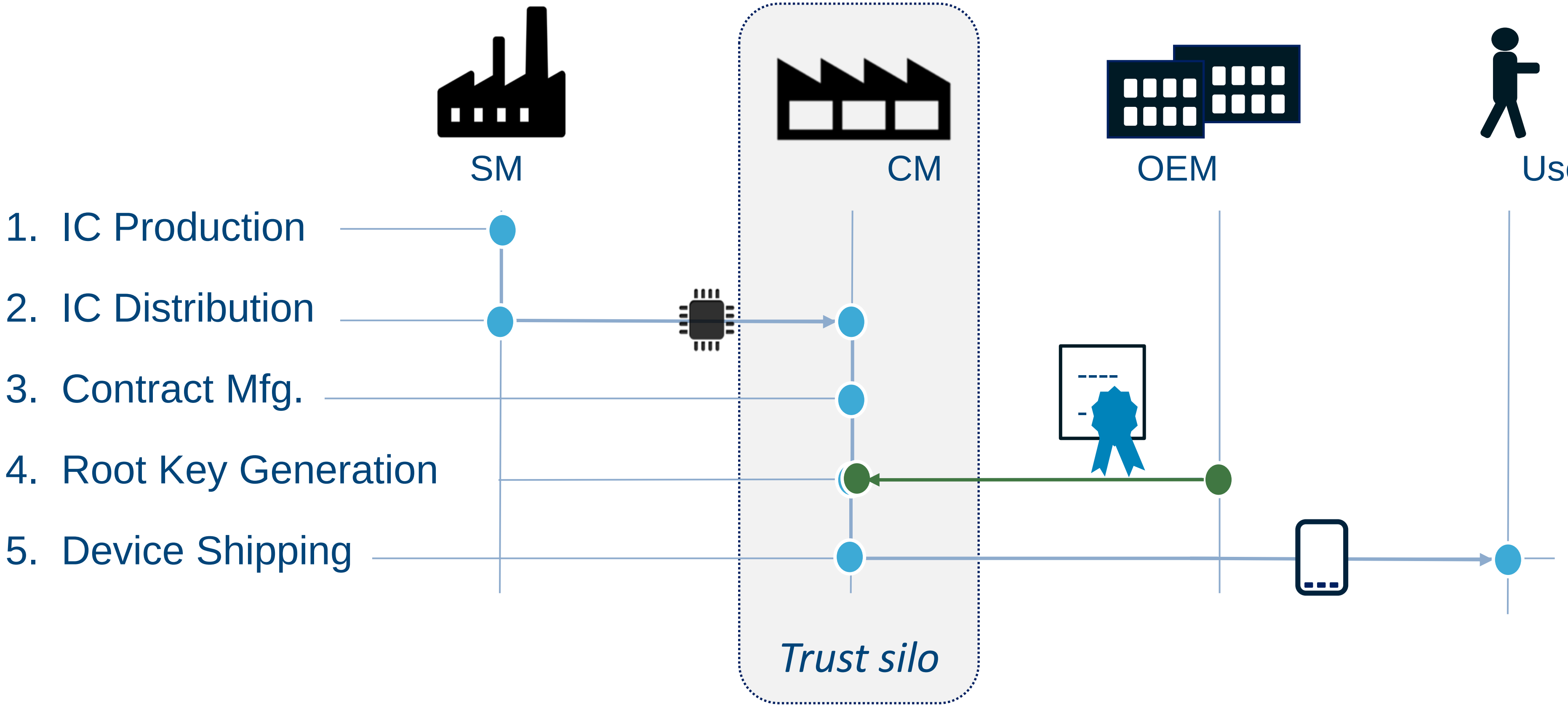
- Extract device-unique cryptographic root key from SRAM PUF using a Fuzzy Extractor
- Multiple symmetric and asymmetric root keys are derived directly from the PUF root key, via cryptographic key derivation functions
- A trusted party (e.g. OEM or CA) creates an identity certificate for the device public key



Advantages

- Root key programming (enrollment) can be done at any stage in the production chain → **Flexible**
- No OEM keys need to be handled by Silicon Manufacturer → **Reduced liabilities and costs**
- No sensitive data stored in any NVM → **Secure**
- Uses standard SRAM available in any device, algorithms can run in software on standard microcontroller → **Widely deployable**

Example of improved key provisioning flow



Comparison of root key storage mechanisms

Advantage	**Fuses / eFuses / anti-fuses** Programmed at SM	**Embedded Flash/EEPROM** Programmed with test/debug/app SW	**SRAM PUF** Programmed with test/debug/app SW
Flexible programming	-	√	√
Reduced liabilities and costs	-	√	√
Secure key storage	√	-	√
Guaranteed uniqueness of keys	-	-	√

Conclusions

- Billions of IoT devices need to be secured in the near future
- Devices need to be able to authenticate to each other and setup secure connections autonomously
- Traditional key storage methods do not provide the required combination of security, flexibility and scalability
- SRAM PUF technology provides a universal solution that is low cost, flexible and widely deployable
- PUF based key provisioning reduces key handling liabilities and costs in the supply chain

Thank You!

Geert-Jan Schrijen, CTO
geert.jan.schrijen@Intrinsic-ID.com