


**Secure non-volatile memory: is it a new trend for the security of embedded systems?**

Guido Bertoni

2007  
1987  
20 years dedicated to the future 

The slide features a dark blue background with light blue geometric shapes and lines. The ST logo is prominently displayed in the upper right. The text is in a clean, sans-serif font.

### Outline of the presentation

- ▣ Actual system of SoCs
- ▣ Practical Case of Mobile Phone
- ▣ Add Security to Flash Memory
- ▣ Centric vs Distributed



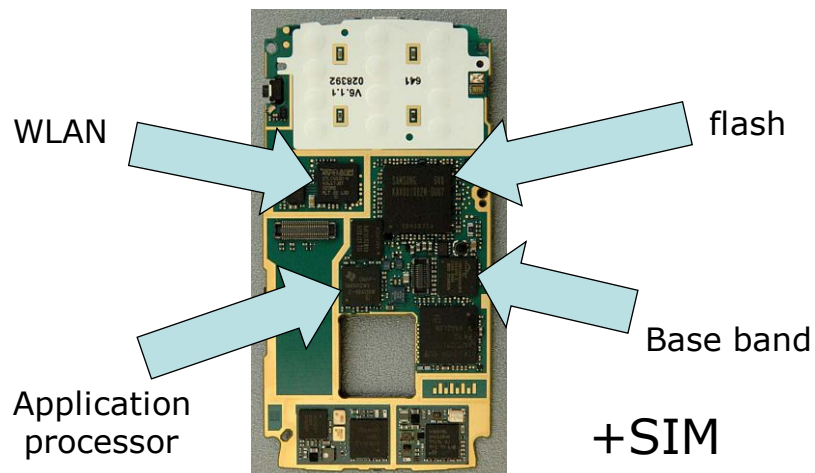
## Complex evolving devices

- ▣ Electronic devices are composed by less and less chips, but still single chip devices are not common particularly for devices with complex functionalities
- ▣ Take an example, mobile phone
  - ▣ Multiple Connectivity
    - ▣ Cellular, Bluetooth, Wlan ...
  - ▣ Multimedia
    - ▣ audio and video acquisition/reproduction
  - ▣ Storage

Secure non-volatile memory



### SoC?



Secure non-volatile memory



## Mobile phone application

- ▣ Mobile phone was initially conceived as phone
  - ▣ Just to place calls
- ▣ The security requirements were mainly two
  - ▣ Telecom operator: identify subscriber for billing
  - ▣ End user: privacy of the communication



Secure non-volatile memory



## System Level Security

- ▣ Base band encrypts GSM traffic
- ▣ SIM takes care of the subscriber identification
- ▣ Subscriber Identification most critical
  - ▣ SIM is a secure component
  - ▣ Not the base band



Secure non-volatile memory



## But...

- ▣ A new business model was developed
- ▣ Telecom operators provider of
  - ▣ service of placing phone call
  - ▣ renting of the device
- ▣ Cost of the mobile phone paid by the user partially as initial fee, and partially with phone call
  - ▣ Different phone -> different cost of the call!



Secure non-volatile memory



## Sustainability of the business model

- ▣ SIM and mobile phone needs to be "paired"
  - ▣ SIM locking is the technical term used
- ▣ Prevent a mobile phone to use any SIM card
- ▣ Firmware of the baseband developed for linking the device to a SIM card



Secure non-volatile memory



## SIM Unlocking

- ▣ SIM unlocking works on the mobile phone side
- ▣ Firmware of the mobile phone is hacked
- ▣ SIM unlocking estimated to be a 1B\$ annual cost
- ▣ What are the security requirements on the mobile phone?
  - ▣ Code integrity/authenticity
  - ▣ Control software upgrade



Secure non-volatile memory



## TREND

- ▣ Evolution of devices, like mobile phone, to portable multimedia devices
- ▣ Predict security needs for enabling new business models
- ▣ Target: broad open security model
  - ▣ Avoid ad-hoc solutions & obscurity
- ▣ Protecting data and applications



Secure non-volatile memory



## Security overview

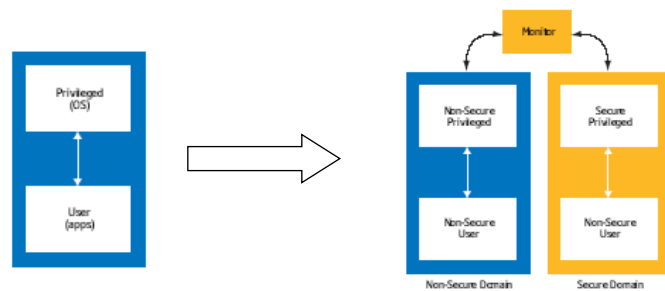
- ▣ Practically SIM (or Smart Card or TPM) will remain an important piece of the security model
- ▣ But mobile phone security should grow since it can not rely solely on the SIM security (or TPM)
- ▣ Where security pieces should be placed?

Secure non-volatile memory



## Application Processor Enforcement

- ▣ Efforts for enforcing application processor is clear
- ▣ ARM TrustZone



Secure non-volatile memory



## Application Processor Enforcement

- ▣ Crypto accelerators
- ▣ On chip ROM & RAM
  - ▣ Secure boot
  - ▣ Secure code
  - ▣ Secure memory
- ▣ ChipID and Fuses
  - ▣ Storing a unique key



Secure non-volatile memory



## Application Processor Enforcement

- ▣ It is needed!
- ▣ Increase the security level of the software stack and avoid a certain number of attacks
- ▣ Even the PC is evolving in this direction
  - ▣ Intel and AMD initiative
  - ▣ Micro kernels



Secure non-volatile memory



## The End?

- Is it the ultimate solution?
  - Yes if the system converge to a real single SoC
  - Application Processor should be equipped with non volatile memory capability



Secure non-volatile memory



## Code Integrity

- What the Application Processor can do for preserve security of external memory?
  - Thanks to HW accelerator & unique key
    - Protect privacy encrypting it
    - Verify authenticity via digital signature
    - Integrity through MAC-based approach
  - AP behaves as a master
  
- But if the code is buggy...
- Exploiting a bug could be used for loading external applications



Secure non-volatile memory



## Software Upgrade

- ▣ A patch is released and a software upgrade is executed
- ▣ The device should be capable of discriminate the good code from the bad code
  - ▣ Not run old versions!
  - ▣ prevent downgrade
- ▣ Ideally the SoC should have a secure non volatile memory
  - ▣ Secure in term of authenticated reading/writing operations



Secure non-volatile memory



## On chip NVM

- ▣ Integration of CMOS-logic and FLASH is already available but it is a compromise:
  - ▣ Memory size is reduced (< 1 MB)
  - ▣ Reduced frequency (order of 100MHz)
  - ▣ The two technology can not be combined for high end performances



Secure non-volatile memory



## On Chip OTP

- ▣ It is possible to have One Time Programmable memory cells in a general ASIC
- ▣ These are very expensive and thus limited in the amount of bits available
- ▣ It is an ad-hoc solutions not available for a general platform with many different software pieces
- ▣ They are used for
  - ▣ Storing chip ID
  - ▣ Device secret key
  - ▣ Root public key



Secure non-volatile memory



## secure Flash

- ▣ If NVM capability are difficult to add to the SoC, add security to the FLASH!
- ▣ Flash incorporates already processing elements
  - ▣ It is not a plain memory array
- ▣ Adding crypto is not a big cost
- ▣ Storing a key is not a problem at all
- ▣ Flash can be enriched with security features



Secure non-volatile memory



## Authenticated Command

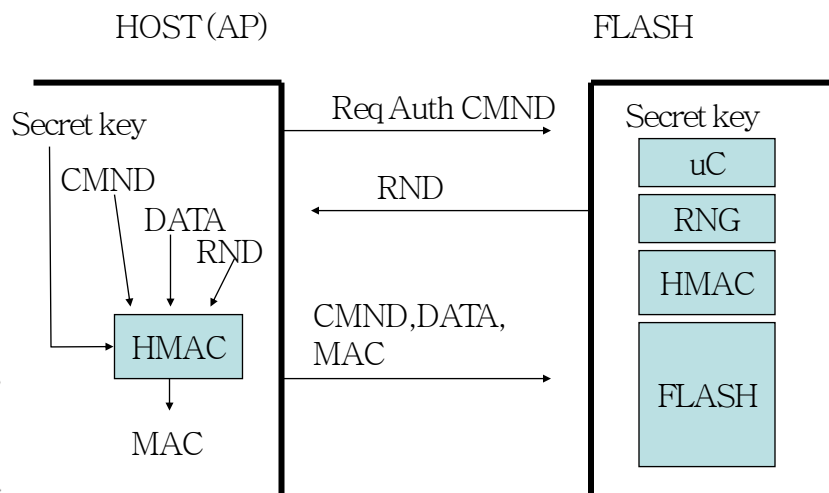
- ▣ Use case: local Authenticated Command
- ▣ Threat: Flash is read/written through standard commands
  - ▣ Flash is not capable of distinguish who is executing a write operation
  - ▣ Add this functionality!



Secure non-volatile memory



## Authenticated Write



Secure non-volatile memory



## Authenticated Command

- ▣ Command authentication can be done
  - ▣ With MAC for local authentication using secret key
  - ▣ With public key digital signature for remote flash commands
- ▣ Memory can be partitioned
  - ▣ different keys for different memory segments
- ▣ Different actors can manage their own memory segment
- ▣ A RNG can be used for preventing replay attack



Secure non-volatile memory



## Trend

- ▣ Terminal is evolving to an open platform where different actor would like to have their own "secure sand box"
  - ▣ Mobile phone firmware
  - ▣ Telecom operator software
  - ▣ Operating System
  - ▣ Third party services
  - ▣ User????



Secure non-volatile memory



## Device evolution: Centric vs Distributed

- ▣ There are two alternatives
  - ▣ Convergence to a single SoC
  - ▣ Keep a system of SoCs
- ▣ The SIM-only approach can not satisfy all the security needs



Secure non-volatile memory



## Centric Approach

- ▣ From a set of SoCs convergence to a single SoC
- ▣ In order of manage security in the SoC NVM capabilities are needed
  - ▣ A minimal amount could be dimensioned, and a "trusted process" manage access to main external memory
  - ▣ Like a driver in the secure part of the trusted platform



Secure non-volatile memory



## Distributed approach

- ▣ Different SoCs is the actual state
- ▣ Technology compatibility
  - ▣ RF vs CMOS vs Memory
- ▣ Cost of integration
- ▣ Security of distributed system can be addressed and could be an open model



Secure non-volatile memory



**THANKS!**



Secure non-volatile memory

