Secure Key Management on General-purpose Processors

L. GASPAR, V. FISCHER, F. BERNARD

Secure Embedded Systems group
Laboratoire Hubert Curien, UMR 5516 CNRS
Université Jean Monnet, Saint-Etienne, France
Member of Université de Lyon

Introduction (1/2)

- Current problems in communication security

1) Many customer services, frequent changes in communication standards

=> Processors are often used

+ flexibility, fast adaptation to new standards

- slow execution of cryptographic algorithms
- low security (software attacks)

2) Need for a crypto-coprocessor

- Higher performance
- But how about security?

Introduction (2/2)

- Current problems in communication security (cont.):

  3) Higher security levels in microprocessor systems are required

  - Growing market: from military to telecommunications, avionics, automotive, banking, multimedia, …
  - FIPS-140-2 levels 3 and 4: when transferred in clear, secret keys have to be exchanged using a separate channel, otherwise keys have to be encrypted when being exchanged via data channel
  - Counter-measure techniques against side channel attacks and fault-injection are not 100% efficient
  - Because of side channel attacks threat, the keys must be generated and exchanged regularly
  - **Secure key management => A serious problem nowadays !!!**

Current cryptographic processors (1/3)

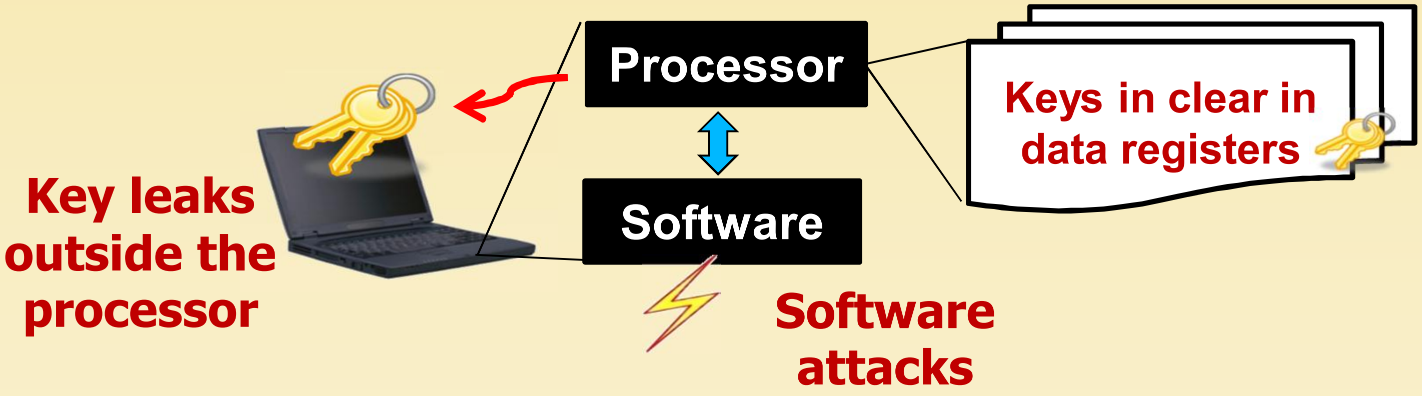
1. Software implementation – processors, controllers (e. g. [1])

Advantages

- High flexibility, low price

Disadvantages

- Vulnerability to software attacks => software manipulation can lead to total key disclosure! (e.g. cache attack [2])
- Slow


Processor
Software
Keys in clear in data registers
Key leaks outside the processor
Software attacks

[1] J. Steiner, "Kerberos: An Authentication Service for Open Network Systems," Proc. Winter USENIX Conference, pp. 191-201, 1988

[2] E. Bangerter, D. Gullasch, and S. Krenn, "Cache games–Bringing access-based cache attacks on AES to practice," Workshop COSADE, pp. 215–221, 2011.

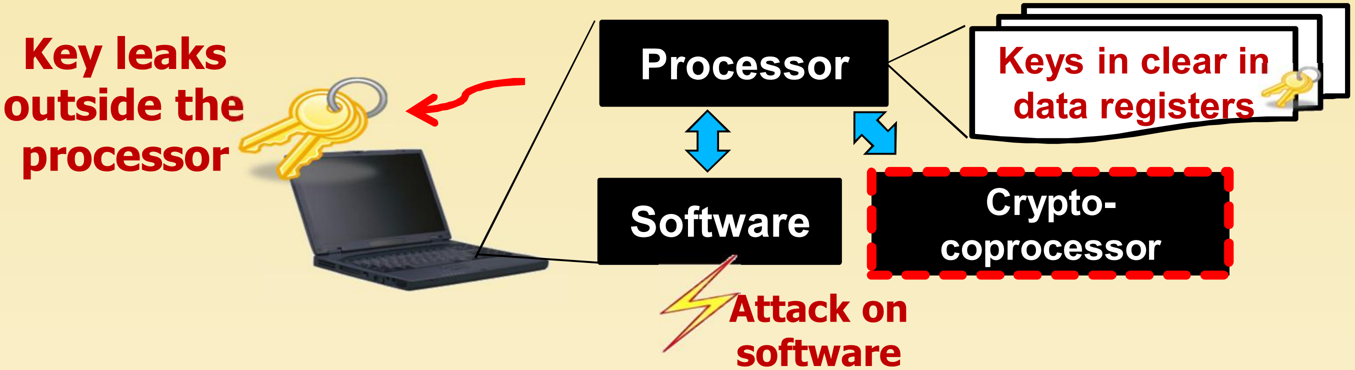
Current cryptographic processors (2/3)

2. Hardware implementation – Cryptographic coprocessor

a) Processor + Cryptographic coprocessor [3], [4], connected to the processor via dedicated coprocessor bus

Advantage – Higher speed

Disadvantage – Vulnerability to software attacks remain the same


Key leaks outside the processor
Processor
Keys in clear in data registers
Software
Crypto-coprocessor
Attack on software

[3] K. Sakiyama, "Multicore curve-based cryptoprocessor with reconfigurable modular arithmetic logic units over GF (2n)," IEEE Transactions on Computers, pp. 1269–1282, 2007

[4] F. Crowe, "Single-chip FPGA implementation of a cryptographic co-processor," IEEE International Conference on Field-Programmable Technology, pp. 279–285, 2004

Current cryptographic processors (3/3)

2. Hardware implementation – Cryptographic coprocessor (cont.)

b) Processor + Cryptographic coprocessor + virtual key memory [5]

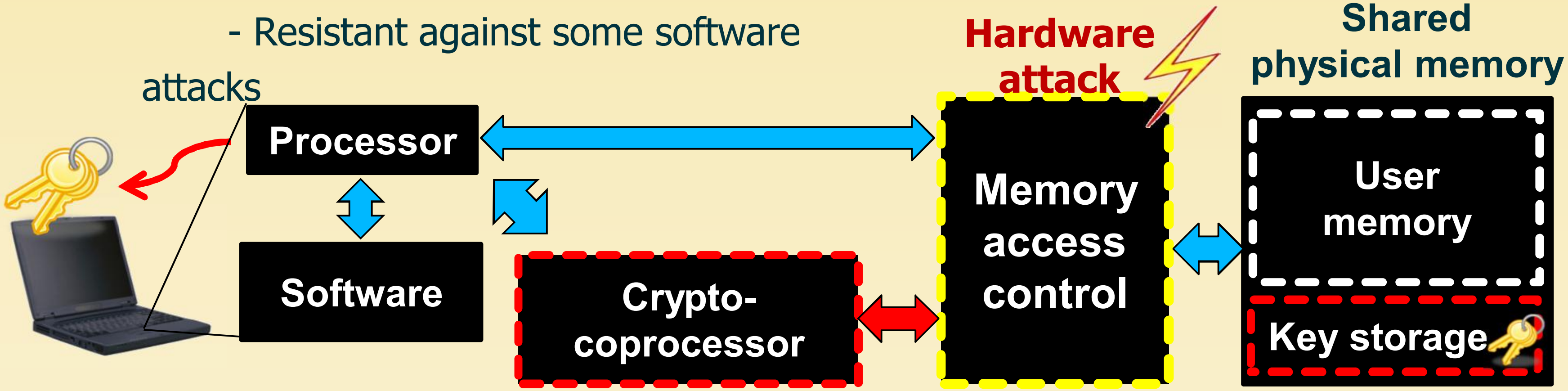
- Physical memory separated to virtual user memory and virtual key memory
- Processor can access only virtual user memory

Advantages

- Hardware acceleration => high speed
- Resistant against some software attacks

Disadvantage

- Physical isolation is impossible (hw Trojans)



[5] A. Ashkenazi and D. Akselrod, “Platform independent overall security architecture in multi-processor system-on-chip integrated circuits for use in mobile phones and handheld devices”, Computers & Electrical Engineering, vol. 33, no. 5-6, pp. 407–424, 2007.

Scientific gap

The gap

- Solution enabling manipulation with keys by a processor and guaranteeing physical isolation of the key memory does not exist
- General-purpose processor has to be able to manage secret keys across the isolation zone without having access to them in clear

Two solutions are possible by using

- Dedicated processor with ability to manage keys in a secure way (CryptArchi 2010)
- General-purpose processor + special security module (this talk)

The proposed solutions have to fulfill stringent requirements of

Principles of security zones separation and physical isolation

Outline

- Principle - separation of security zones
  - Protocol level
  - Architectural level
  - Physical level
- Interfacing GPP and security module
  - Internal processor bus
  - Coprocessor dedicated internal bus
  - Peripheral bus
- Implementation and results
  - NIOS II security extension
  - Microblaze security extension
  - Cortex 1 security extension
- Conclusions and perspectives

Separation principle (1/5)

- In order to guarantee the security, security zones must be created

- We propose to create three security zones
  - Processor
  - Cipher block (for reconfiguration purposes)
  - Key memory (for security)

- Security zones must be separated on several levels
  - Protocol level
  - System level
  - Architectural level
  - Physical level

Separation principle (2/5)

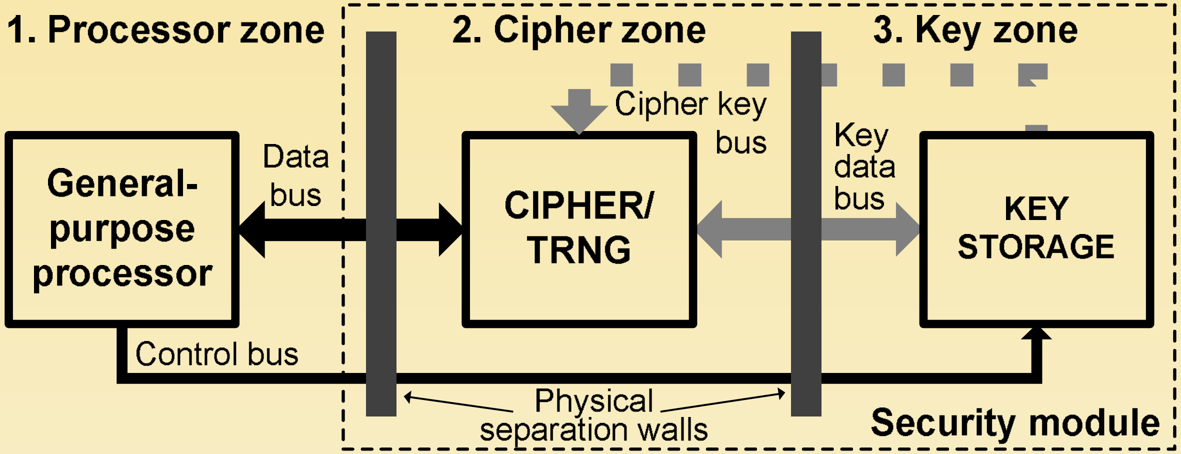
Separation on the protocol level

- Separation of key management and data processing tasks
  - Data processing (execution of cryptographic modes) performed directly by the GPP
  - Key management performed by the GPP only indirectly (hidden keys)
- Two-level key hierarchy
  - Session keys: data enciphering, can enter GPP only when enciphered
  - Master keys: enciphering of session keys, must never enter GPP
- Communication protocol is defined
  - Data are transported in packets, while being enciphered with session keys
  - Session keys are enciphered using master key and included in packets
  - To prevent side channel attacks, session keys have to be changed frequently

Separation principle (3/5)

Separation on the system level

- Creation of three zones: **Processor**, **Cipher** and **Key storage** zone



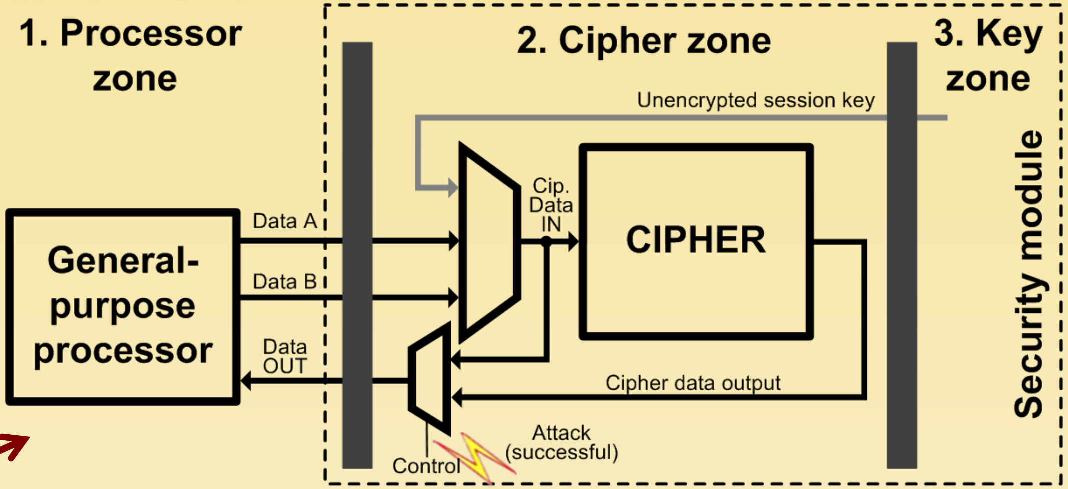
- Key cannot pass to the processor zone directly – has to pass through the cipher

- Processor can manage only key adresses (via dedicated control bus)

Separation principle (4/5)

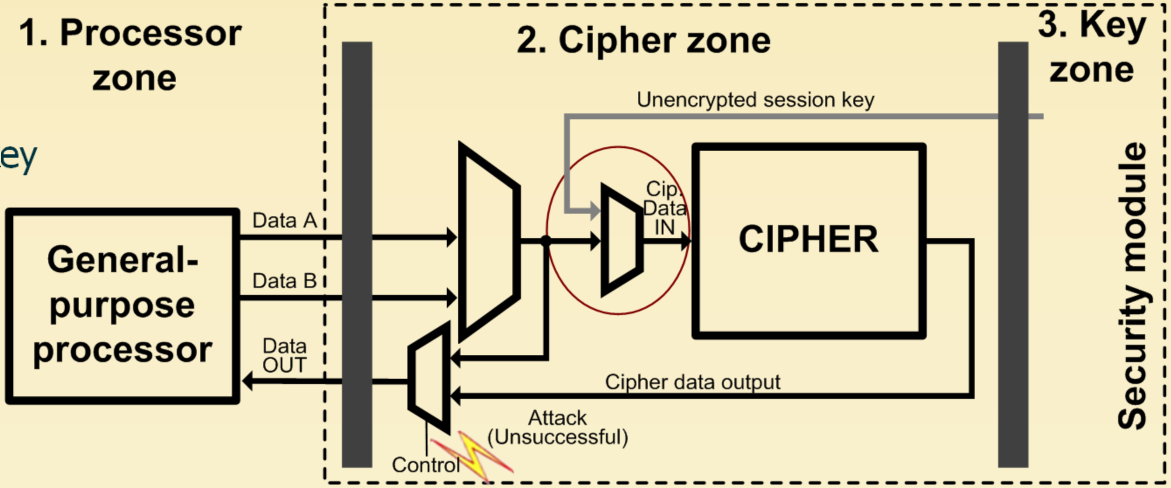
Separation on the architectural level

- Organization of **buses** and **multiplexers** have to be considered
  - Order of multiplexers must avoid key redirection even if the control is violated



- Example:

⇒ BAD SOLUTION: attack on multiplexer control bus can retrieve unencrypted session key

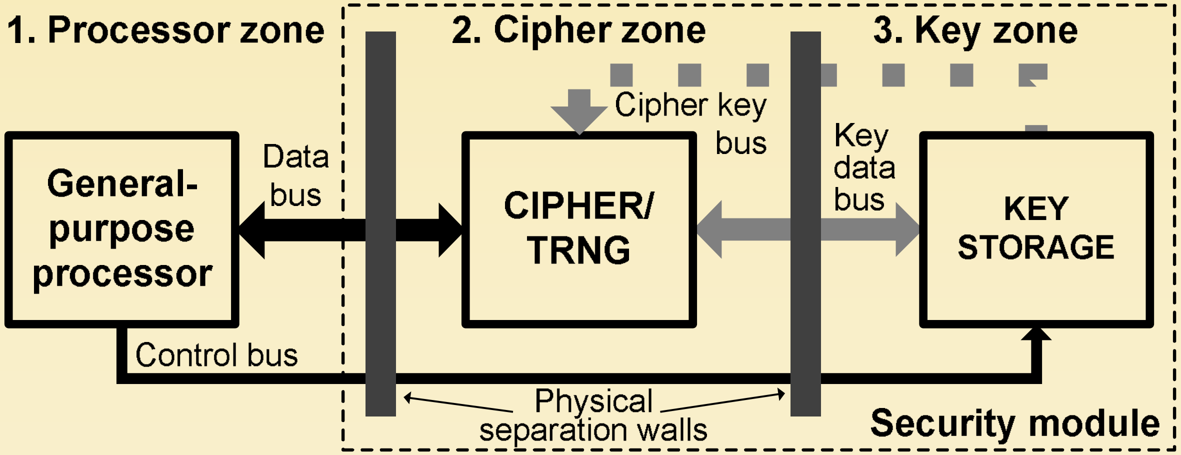


⇒ SECURE SOLUTION: no key retrieval is possible

Separation principle (5/5)

Separation on the physical level

- Special logic area dedicated to each zone – additional separation
  - Zones are isolated from each other by “insulation walls ” – empty zones
  - Only selected signals are allowed to pass between isolated zones using “isolated bridges”



Interfacing GPP and security module (1/2)

- Criteria for selecting interconnections
  - Speed
  - Security
  - Available GPP interfaces
- Speed
  - Latency – depends on interface, protocol, bus width
  - Throughput – depends on cipher and clock frequency
- Security
  - Point-to-point communication is considerably more secure than
  - Point-to-multipoint communication is less secure (potential of eavesdropping)
- Available topologies
  - Internal processor bus
  - Coprocessor-dedicated bus
  - Peripheral bus

Interfacing GPP and sec. module (2/2)

Interconnections between GPP and dedicated security module

- **Internal processor bus** – module is a part of the GPP datapath (custom instructions)

  + Point-to-point connection (higher security)
  + Low latency, high performance
  - Lower clock frequency because of long critical path
  - Not always available

- **Coprocessor dedicated bus** – module is connected to GPP registers, but it is not a part of the GPP's datapath (e.g. FSL bus)

  + Point-to-point connection (higher security)
  + Higher clock frequency, two clock domains can be separated by FIFOs
  - Higher latency (= low performance when exchanging small blocks)
  - Not always available

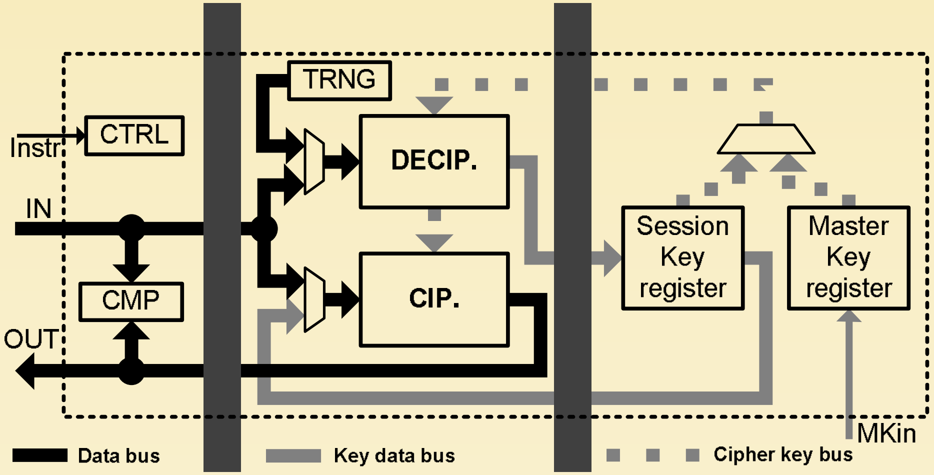
- **Peripheral bus** – connected to GPP via hierarchy of buses, access to peripherals is multiplexed

  + The most flexible solution, available in all processor systems
  - Point-to-multipoint (lower security)
  - Very high latency/very low performance

Security module implementation

Security module extension

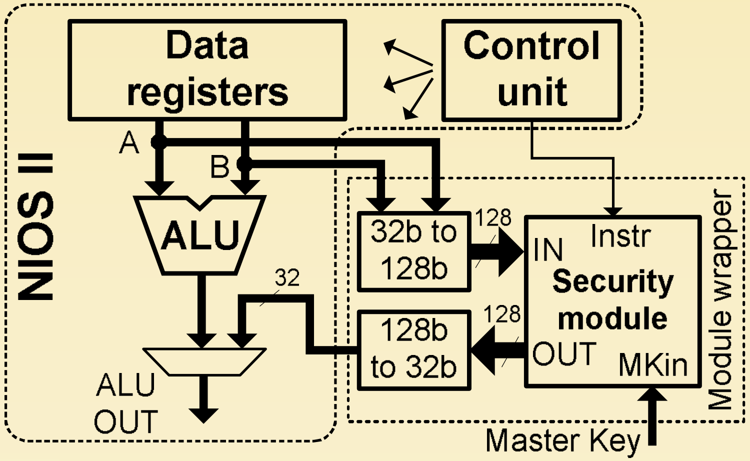
- Separation on selected levels into three zones
  - Unsecured (processor) zone
  - Keys stored in key storage registers: Master key register, Session key register
  - Zone including cipher and decipher cores - AES with a 128-bit datapath is used
- Three independent buses
  - Data bus – carries data and enciphered session keys
  - Key data bus – transport of keys to/from cipher data I/O
  - Cipher key bus – transport of keys to the cipher key input

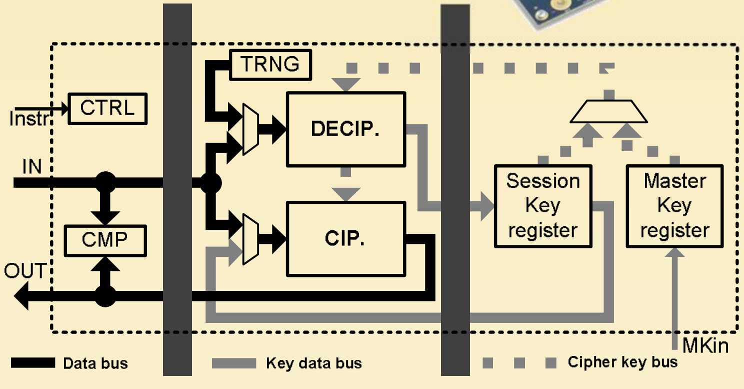


Security extension of NIOS II

Implementation constraints

- Security module wrapper allows implementation of custom instruction set
  - High security, high speed, a good trade-off between the size and speed (32-bit bus remains a bottleneck)
  - Easy control via dedicated control bus passing directly from NIOS II control unit
  - Lower frequency of the NIOS II influenced by the cipher critical path
- Implemented in Stratix II – NIOS II evaluation board



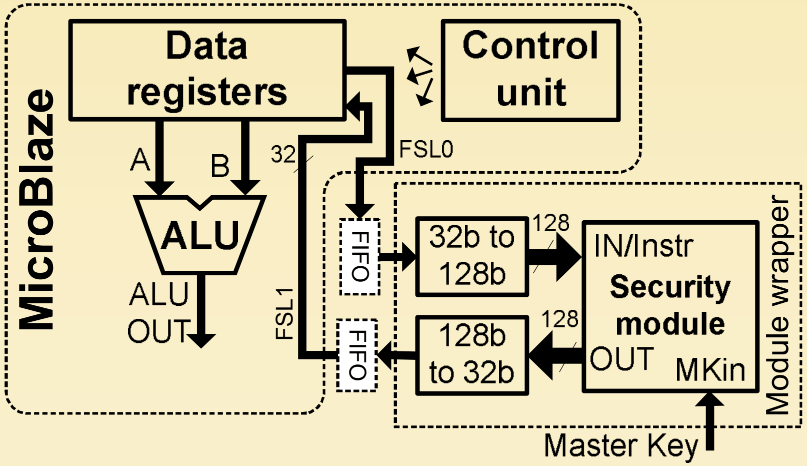


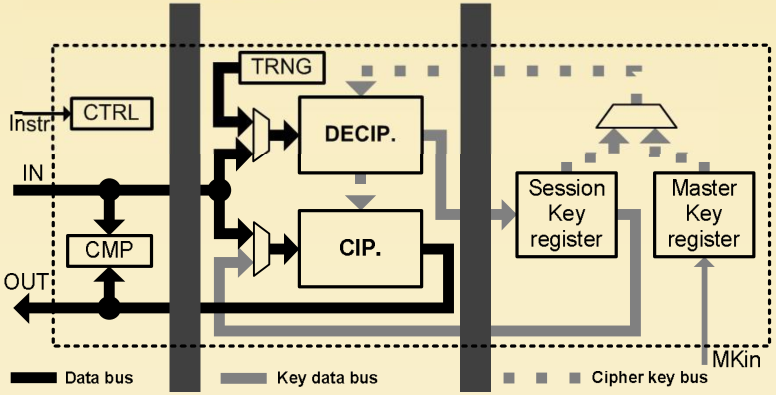
Security extension of MicroBlaze

Implementation constraints

- Security module wrapper is interconnected via FSL buses
  - High security, high latency
  - Difficult to control, because instructions have to pass via FIFOs
  - Higher frequency, because MicroBlaze is not limited by cipher critical path

- Tested in Virtex 6 – ML605 evaluation kit



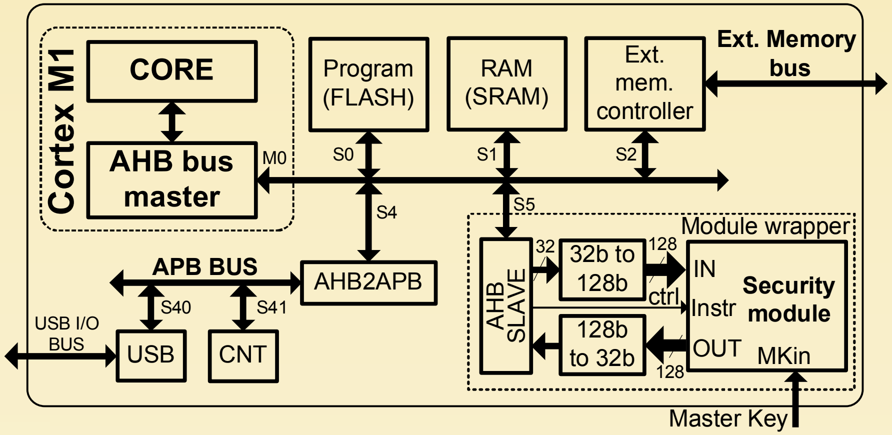


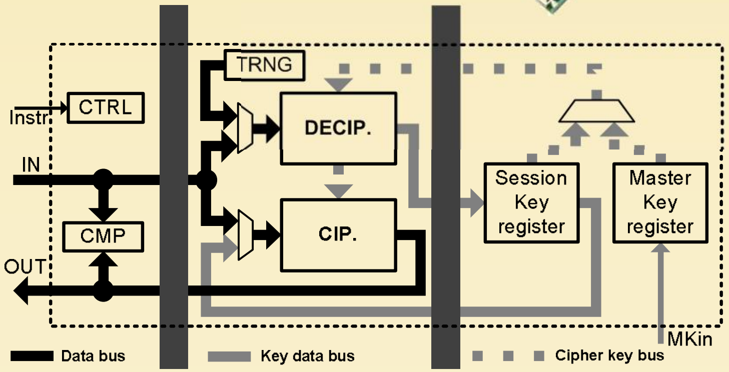
Security extension of Cortex M1

Implementation constraints

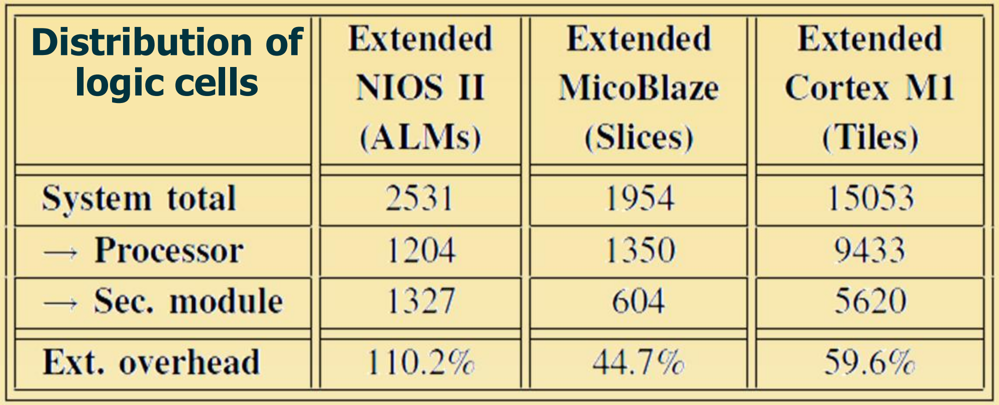
- Security module wrapper is interconnected via AHB system buses
  - Low security – point-to-multipoint bus => other units can eavesdrop
  - High latency – bus is multiplexed: instruction fetching, RAM access, etc.
  - Easy to control
  - Higher frequency, because Cortex M1 is not limited by cipher critical path

- Tested in Actel Fusion M1AFS600 – embedded kit

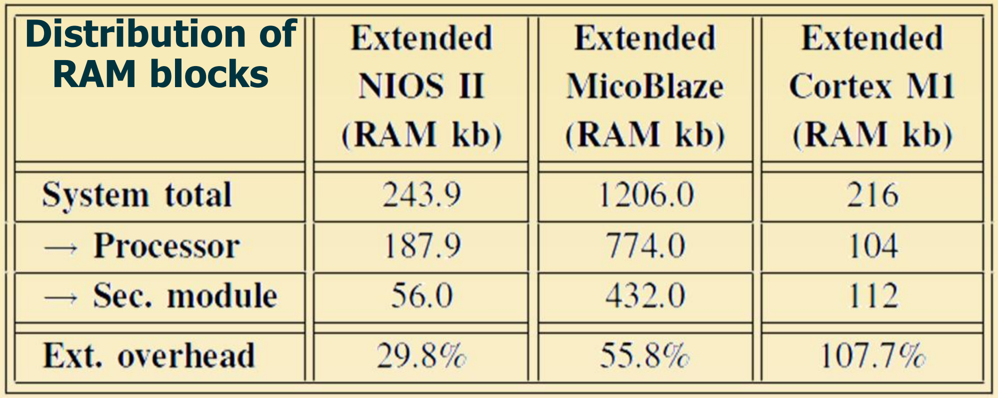




Distribution of FPGA resources



Distribution of logic cells	Extended NIOS II (ALMs)	Extended MicoBlaze (Slices)	Extended Cortex M1 (Tiles)
System total	2531	1954	15053
→ Processor	1204	1350	9433
→ Sec. module	1327	604	5620
Ext. overhead	110.2%	44.7%	59.6%

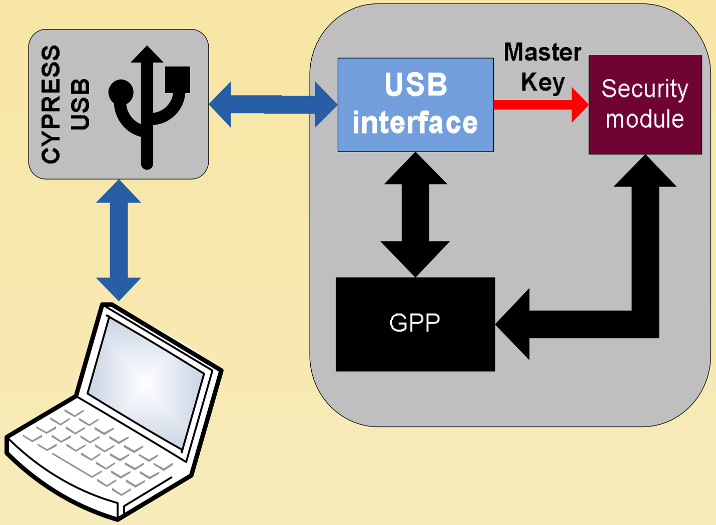


Distribution of RAM blocks	Extended NIOS II (RAM kb)	Extended MicoBlaze (RAM kb)	Extended Cortex M1 (RAM kb)
System total	243.9	1206.0	216
→ Processor	187.9	774.0	104
→ Sec. module	56.0	432.0	112
Ext. overhead	29.8%	55.8%	107.7%

- NIOS II: Security extension overhead is 110% of the processor size
- MicroBlaze: Security extension overhead is 44.7% of the processor size
- Cortex M1: Security extension overhead is 59.6% of the processor size

Tests in hardware

- Initialization
  Master key is transferred from the PC to the key register
- Data exchange
  1. Packet is generated in the PC
  2. Packet is sent to GPP via USB interface
  3. GPP reads the header and recognizes control word and session key
  4. Session key is decrypted inside the security module and authenticated
  5. CFB block cipher mode is executed on data
     a) XOR is performed by the GPP
     b) Ciphering/authentication is performed by the security module
  6. Packet carrying results is created by the GPP and send back to the PC



- Testing system frequency: 50 MHz
- Resulting throughput:
  - NIOS II: 25,1 Mb/s
  - MicroBlaze: 18,4 Mb/s
  - Cortex M1: 12,2 Mb/2

Conclusions & perspectives

Conclusions

- Separation principle was generalized to GPPs (not limited to dedicated cryptoprocessors)
- Interfacing GPP and security module was discussed concerning speed and security
- Special security module for general-purpose processors allowing secure key management secured on four levels has been proposed
- The principle was validated on NIOS II, MicroBlaze and Cortex M1 soft-core processors
- Tested successfully in hardware

Future plans

- Formal verification of the security protocol
- Security module extension for Cortex M3 and Intel Atom