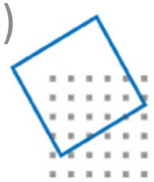
Countermeasures against EM Analysis for a Secured FPGA-based AES Implementation

Paolo Maistri[1], Sebastien Tiran[2], Philippe Maurine[2], Israel Koren[3], Régis Leveugle[1]

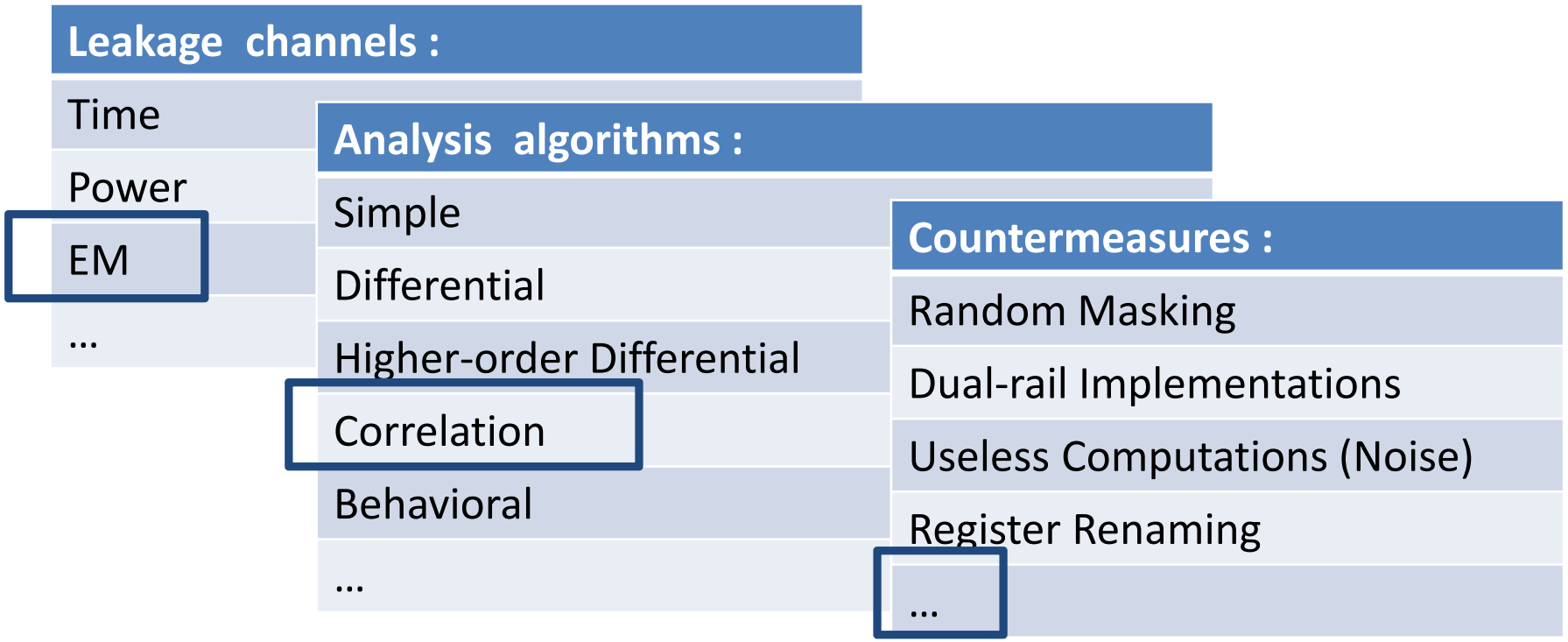
(1) 

(2) 
Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier

(3) 

Background

- Side channel analysis is a major threat against cryptographic implementations

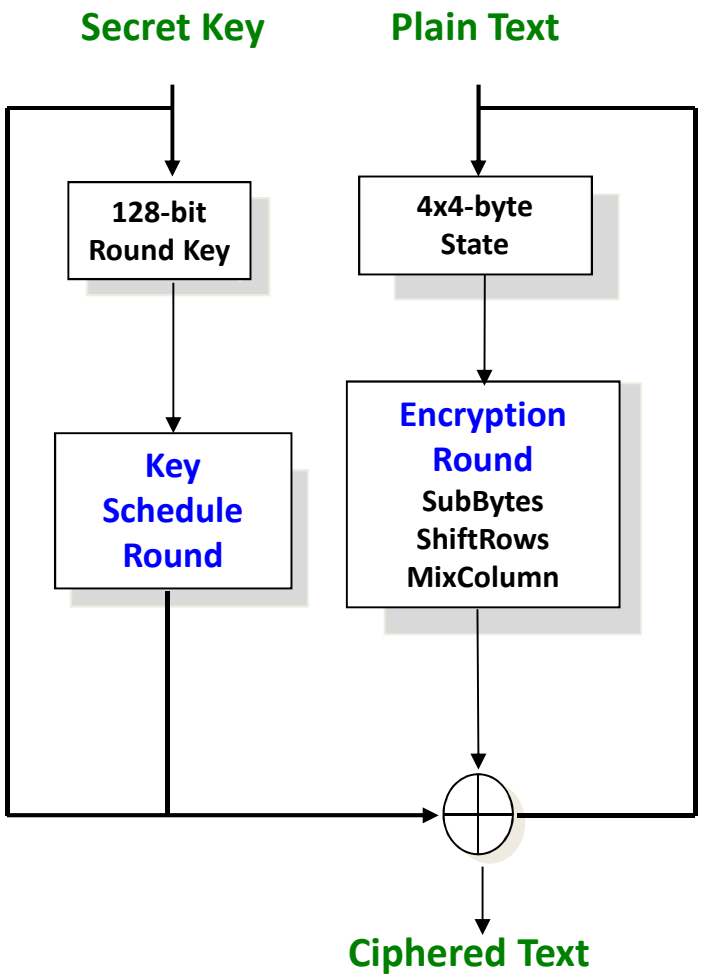


Outline

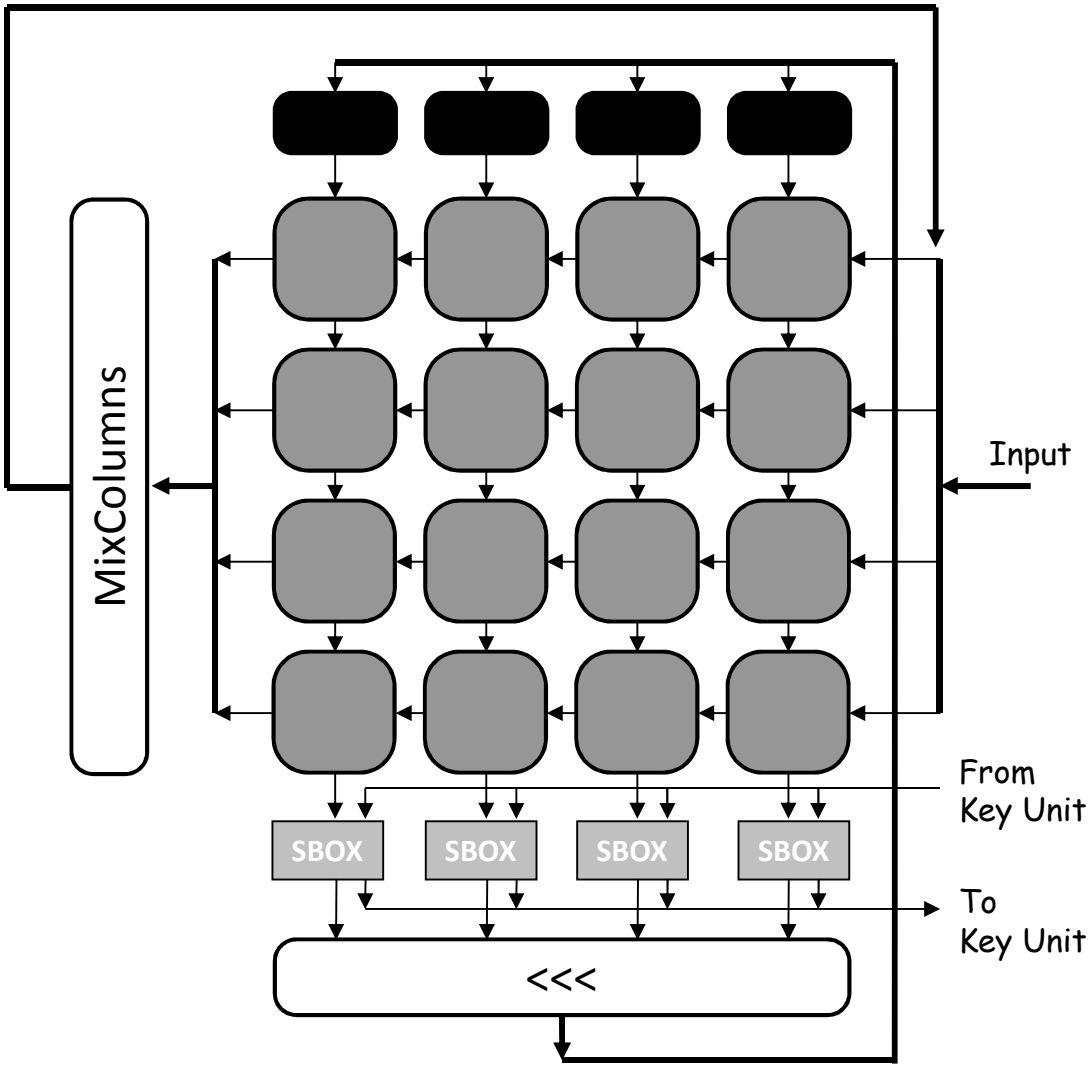
- The Advanced Encryption Standard
  - The algorithm
  - Base implementation
- Proposed Countermeasures
  - Mapping based
  - Relocation based
- Overheads
- Results
- Conclusions

AES algorithm

- Symmetric block cipher
  - Plain Text 128b,
  - Secret Key 128/192/256b
- SPN cipher
  - 10/12/14 rounds
- Round operations
  - SubBytes: nonlinear byte substitution
  - ShiftRows: row-wise word rotation
  - MixColumns: column-wise linear multiplication
  - Key Addition: XOR with round subkey



Starting from a given AES design…



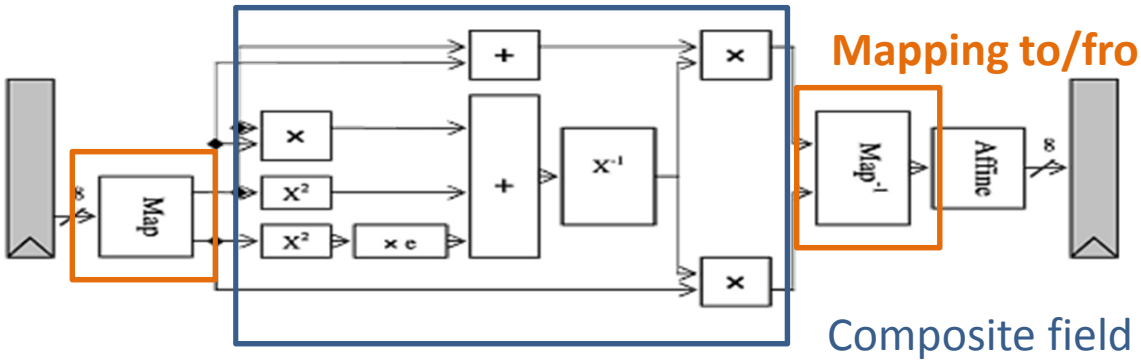
- Quite small
- 32-bit data-path
- 4 Substitution Boxes
- 4 GF Multipliers for *MixColumns*
- 10 clock cycles per round
- On-the-fly key unrolling (using shared *S-Boxes*)

AddRoundKey and State
2-stage S-Box
Register layer
Combinatorial logic
8-bit signal
32-bit signal

Countermeasures against EM Analysis for a Secured FPGA-based AES Implementation
Maistri, Tiran, Maurine, Koren, Leveugle – ReConfig 2013

A Word on S-Box Implementation

- SubBytes is a nonlinear byte substitution
  - Computed as **inverse in GF** + affine transformation
- Implemented as
  - S-BOX Look-Up Table
  - Inverse Look-Up Table + [inverse] affine transformation
  - Inversion in **composite** fields GF(2^4)^2



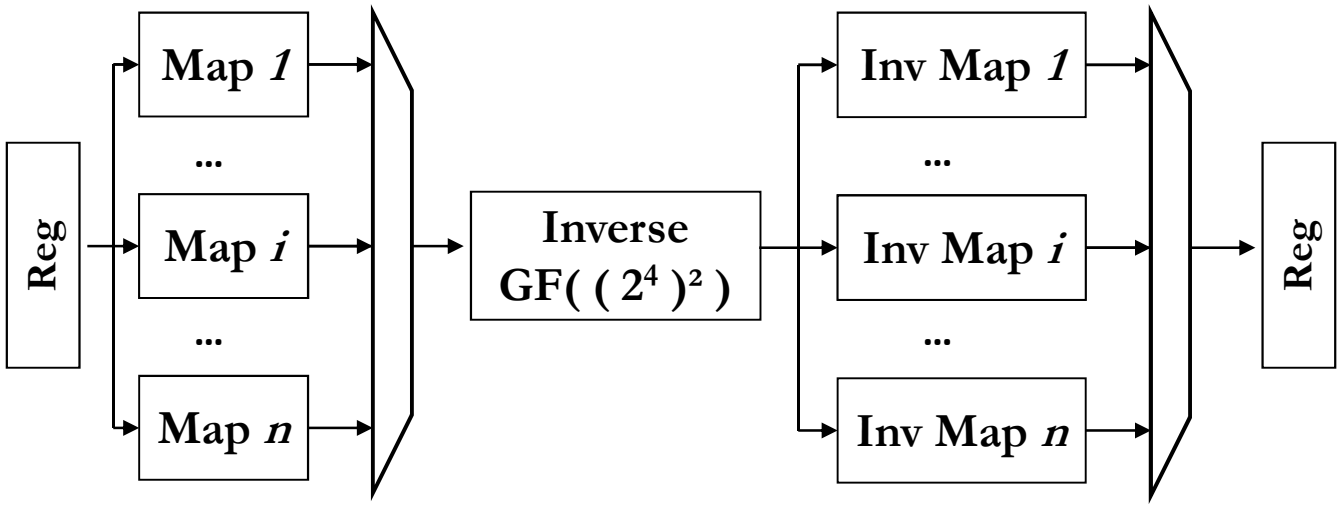
Mapping to/from composite field

Composite field inversion

Countermeasures against EM Analysis for a Secured FPGA-based AES Implementation
Maistri, Tiran, Maurine, Koren, Leveugle – ReConfig 2013

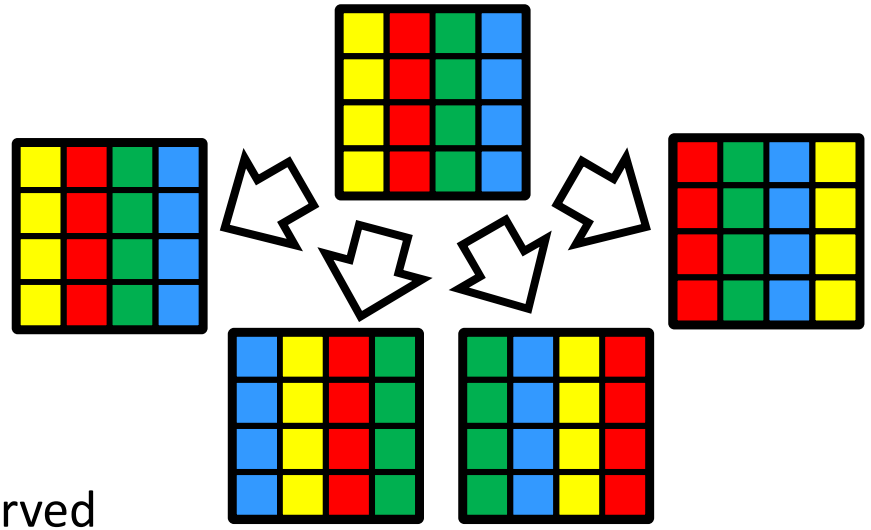
1st scheme: Dynamic Mapping (*DynMap*)

- For each S-Box, implement **several** parallel **composite field mappings**
  - From 1 to 8 possible dynamic mappings for any composite field
  - Choose randomly at runtime (several granularities)
  - At output, choose the correct inverse mapping to get back the result
- Limited to S-Box data path
- Independence of mappings?
- Vulnerable to Zero- and One-value attacks (as multiplicative masking)



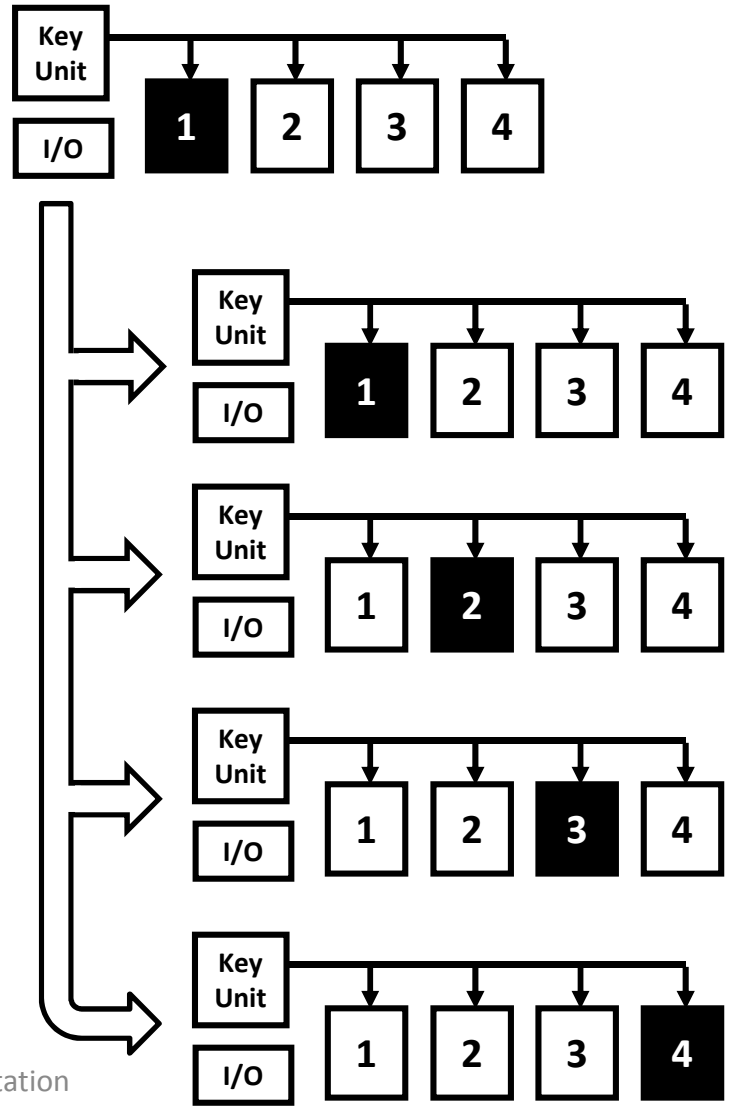
2nd: Dynamic Column Relocation (*DCR*)

- Dynamic resource allocation done **intra-round**
  - Column relocation
- Columns are shuffled at each round
  - Several external constraints (*MixColumns*, *ShiftRows*, …)
  - Only 4 different configurations
  - *ShiftRows* is NOT preserved
  - Column relative ordering IS preserved
- Correct order is restored before output



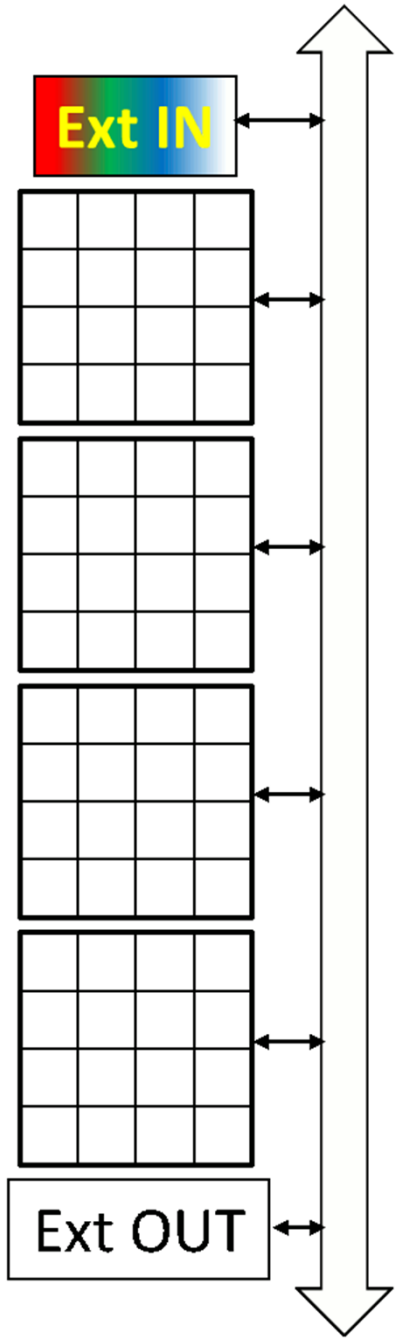
3rd: Dynamic Block Relocation (*DBR*)

- Dynamic resource allocation done **inter-round**
  - Several round components instantiated
  - Higher throughput modes
- At each round, **blocks are shuffled**
  - Temporary cipher text can go to <u>any</u> round instance
  - Auxiliary data is transferred with cipher text
- Variable number of configurations
  - 4 to 24 in the example
  - Depending on number of PTXs



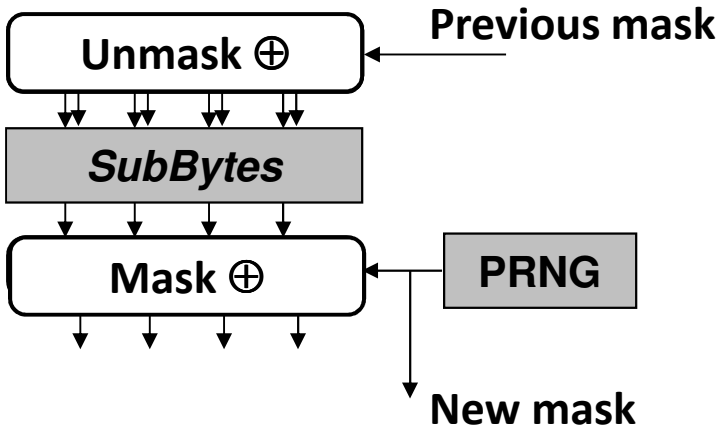
How does relocation work?

- Input text is sent to any round instance
- At each round, columns are scrambled
- At each round, intermediate cipher can be sent to any instance
- Column order is restored before sending cipher to output



4th: Linear Masking (*LinMask*)

- *DynMap* masks only SubBytes data path
  - Rest of design is unprotected and leaking
- Linear masking is a common low-cost masking scheme for linear data path (P network)
  - Masking at S-Box output
  - Unmasking at S-Box input
  - Bus transfers are masked
  - Mask must be transferred with data (due to DBR)



Full Implementation