Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs

Roel Maes[1], Pim Tuyls[1,2], Ingrid Verbauwhede[1]

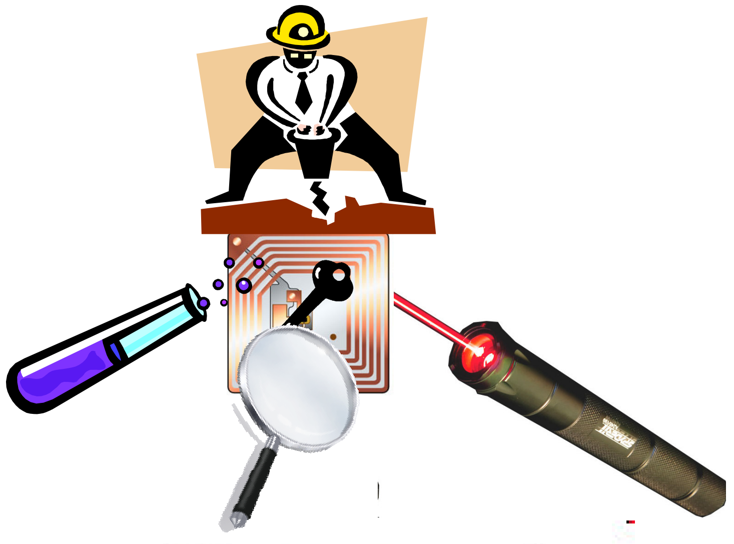
1. COSIC, K.U.Leuven and IBBT

2. Intrinsic-ID, Eindhoven

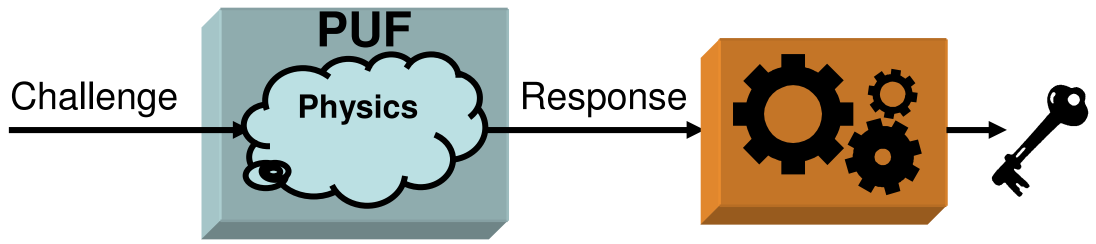
Workshop on Cryptographic Hardware and Embedded Systems, CHES 2009, Lausanne

Overview

Introduction



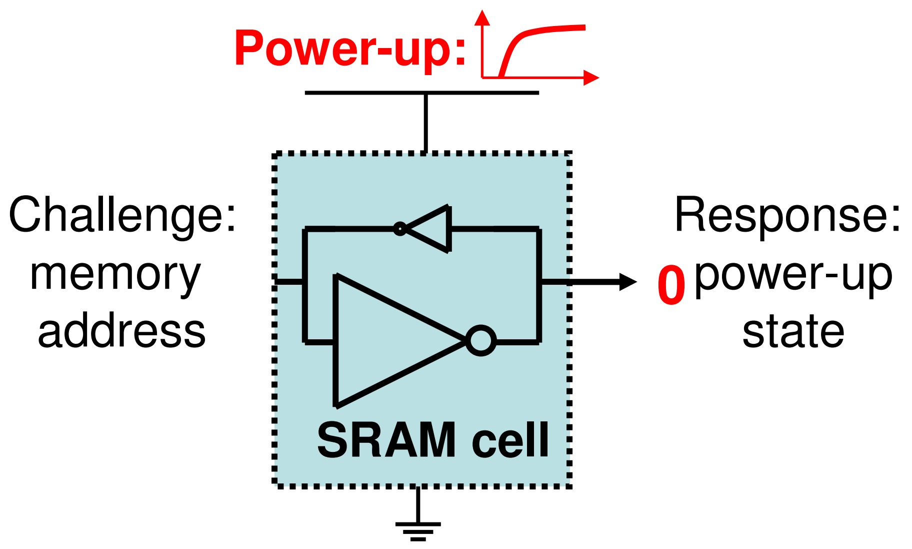
- Tampering attacks threaten secure key storage
- Traditional tampering countermeasures induce large overhead (cost/size/power/…)
- Need for low-overhead physical protection of sensitive data → **PUFs**

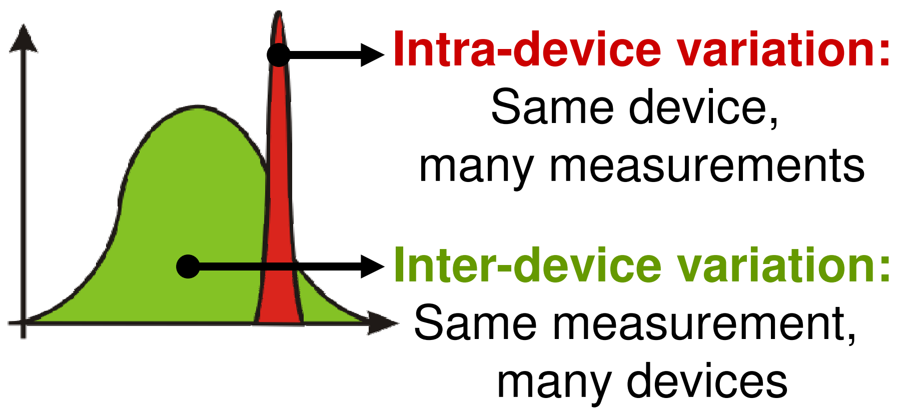


Overview

Related work: The SRAM PUF

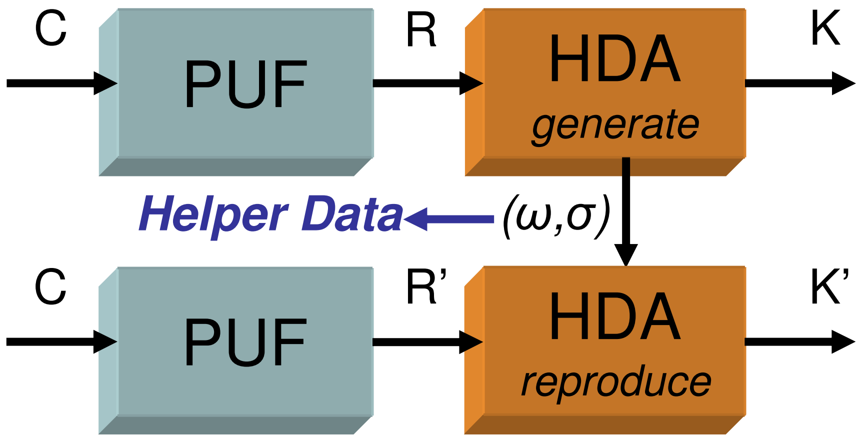
- Random manufacturing variability in ICs is a fact
- Power-up state of SRAM cells efficiently measures intrinsic device variability
  - ➢ **SRAM PUF** [GKST-CHES07]
- (SRAM) PUF responses are *noisy* and *non-uniform*
  - ➢ fuzzy secret

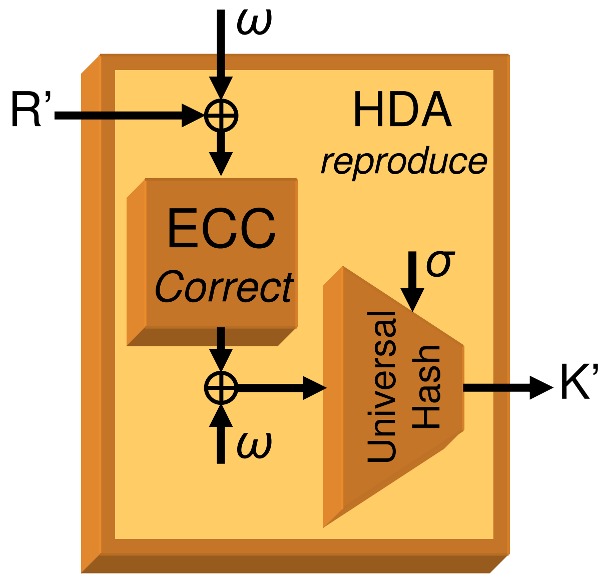




Related work: Helper Data Algorithms

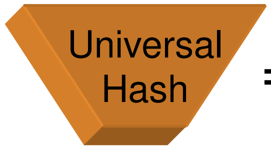
- HDA or **Fuzzy Extractor** extracts a secure key from a fuzzy secret





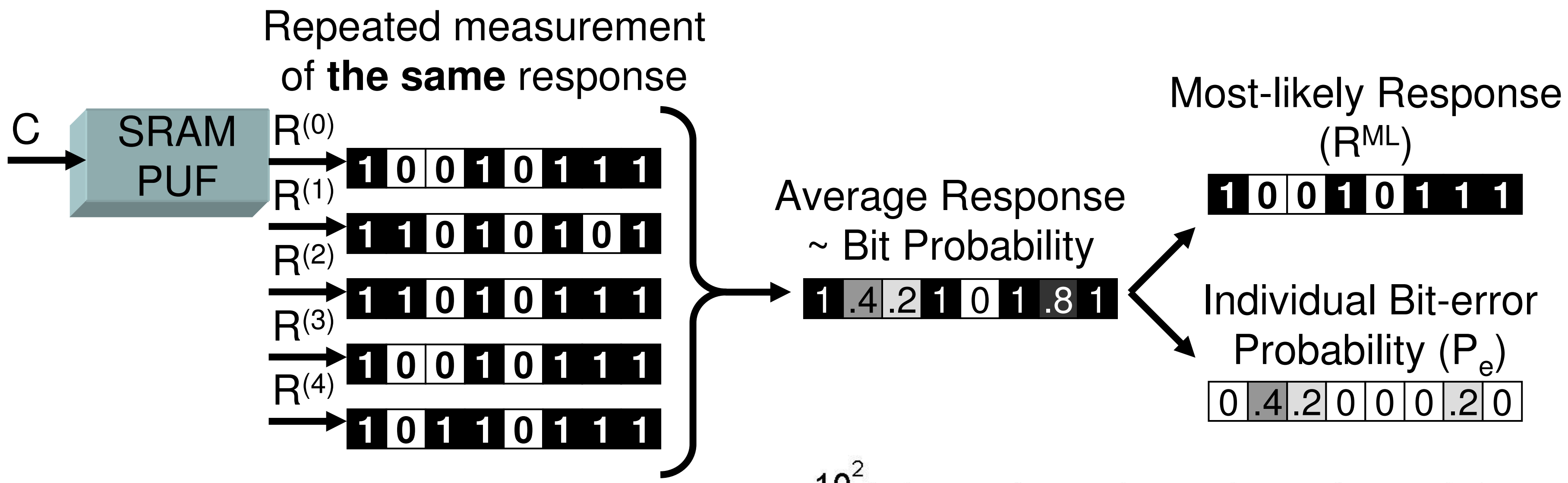
- Efficient implementation possible [BGSST-CHES08]

= **Linear Block Codes:** Golay ○ Repetition or Reed-Muller ○ Repetition

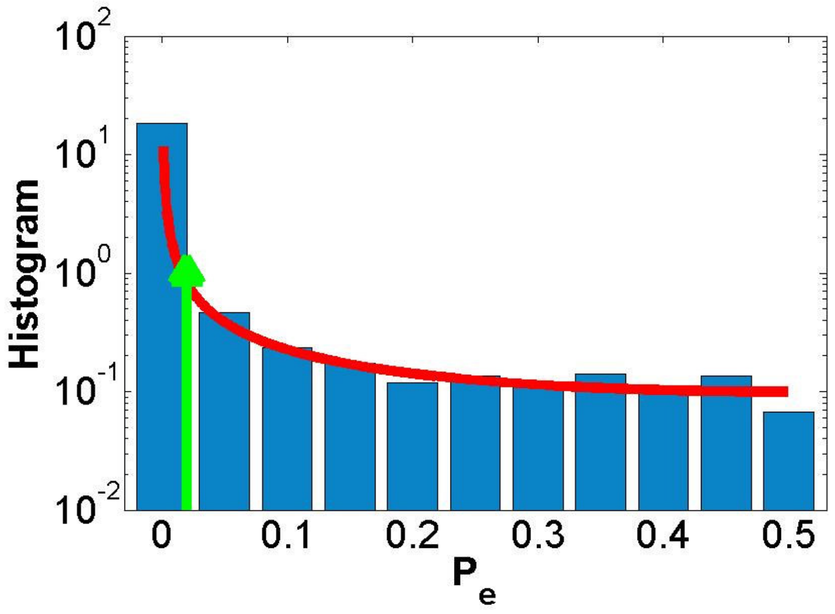


= LFSR-based Toeplitz Universal Hash [Krawczyk-Crypto94]

SRAM PUF Response Characteristics



Repeated measurement of **the same** response
C
SRAM PUF
$R^{(0)}$ 1 0 0 1 0 1 1 1
$R^{(1)}$ 1 1 0 1 0 1 0 1
$R^{(2)}$ 1 1 0 1 0 1 1 1
$R^{(3)}$ 1 0 0 1 0 1 1 1
$R^{(4)}$ 1 0 1 1 0 1 1 1
Average Response ~ Bit Probability
1 .4 .2 1 0 1 .8 1
Most-likely Response (R^{ML})
1 0 0 1 0 1 1 1
Individual Bit-error Probability (P_e)
0 .4 .2 0 0 0 .2 0

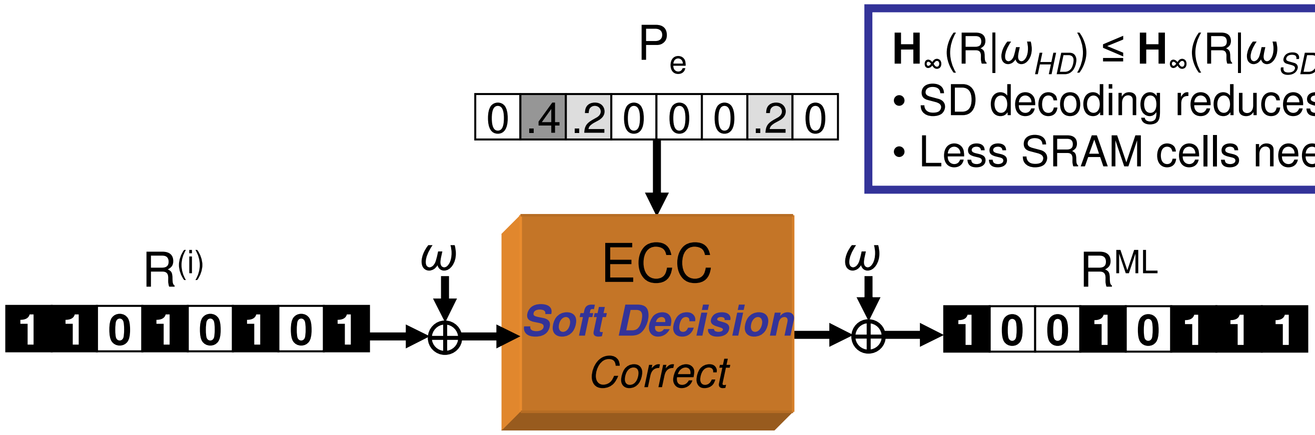


- Histogram of P_e from 6592 bits estimated from 350 measurements
- ⬆ = Distribution according to previous work [GKST-CHES07;BGSST-CHES08]
- ◟ = Distribution according to our new model [MTV-ISIT09]

Soft Decision HDA

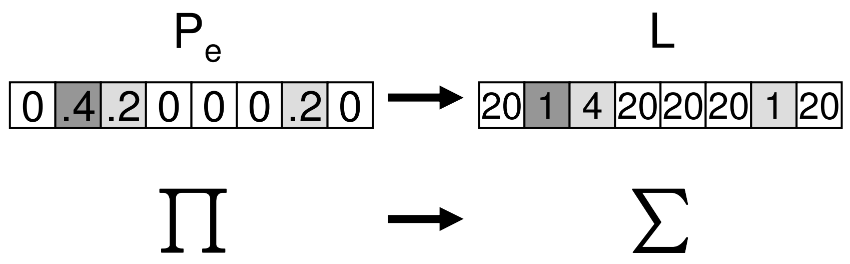
- Regular error-correcting algorithms assume fixed bit error probability for every bit
- Additional *reliability information* of every response bit is available
  - – enables Soft-Decision error correction (SD)
  - – improves efficiency of ECC algorithm

$\mathbf{H}_\infty(R|\omega_{HD}) \leq \mathbf{H}_\infty(R|\omega_{SD}, P_e) \leq \mathbf{H}_\infty(R)$
- SD decoding reduces min-entropy loss
- Less SRAM cells needed for key extraction



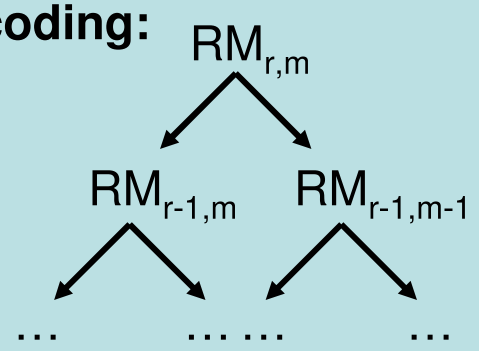
Soft Decision Error Correction

- Reliability information
  → Log-likelihood ratio
- Soft-Decision Maximum Likelihood Decoding (SDML)
  - exponential complexity in code dimension
  - ok for repetition codes
- Generalized Multiple Concatenated decoding (GMC) of Reed-Muller Codes



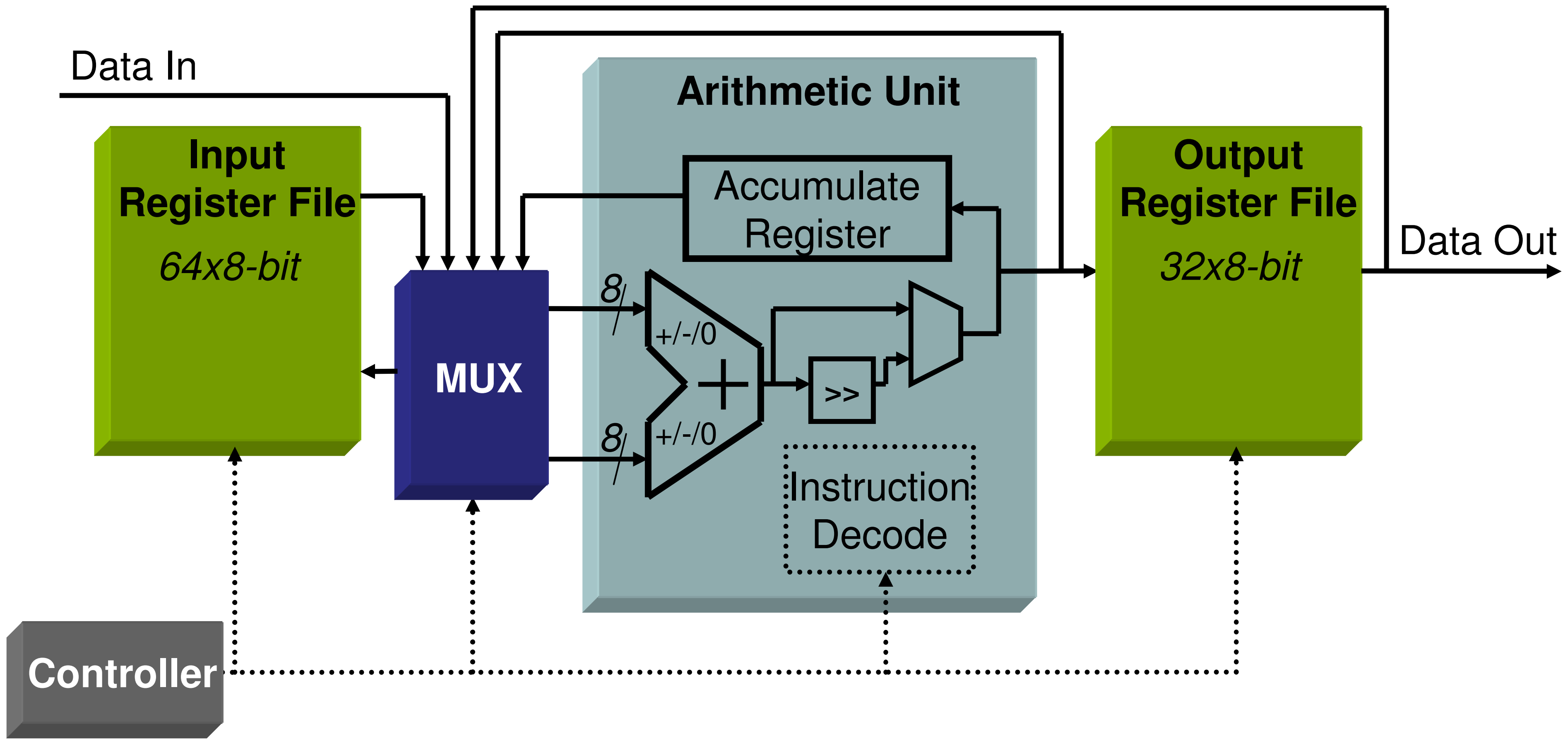
SDML-decoding:
- SD-Repetition-Decode(L) = $\sum(L_i)$
- SD-Degenerate-Decode(L) = L

GMC-decoding:



- $RM_{1,m}$ = Repetition Code → SDML
- $RM_{r',r'}$ = Degenerate Code → SDML

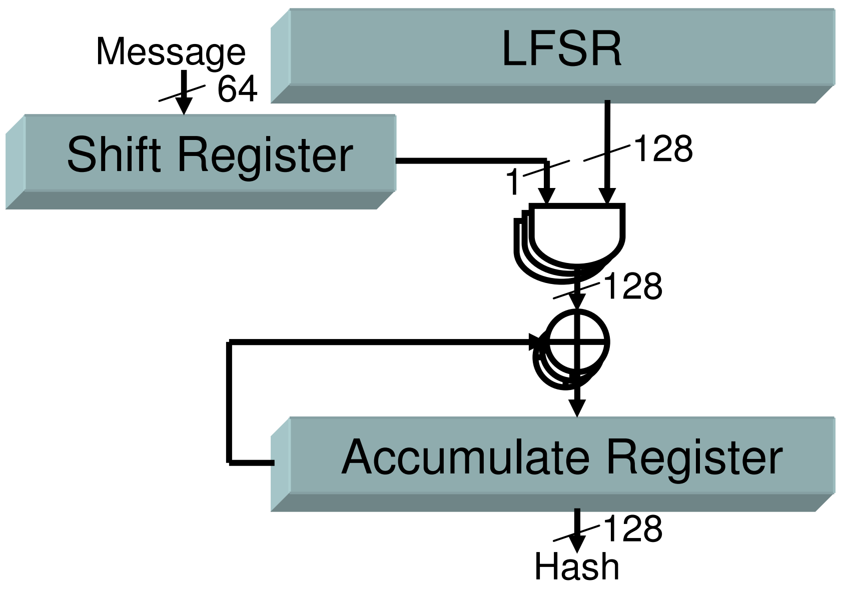
Soft-decision decoder: Datapath



Overview

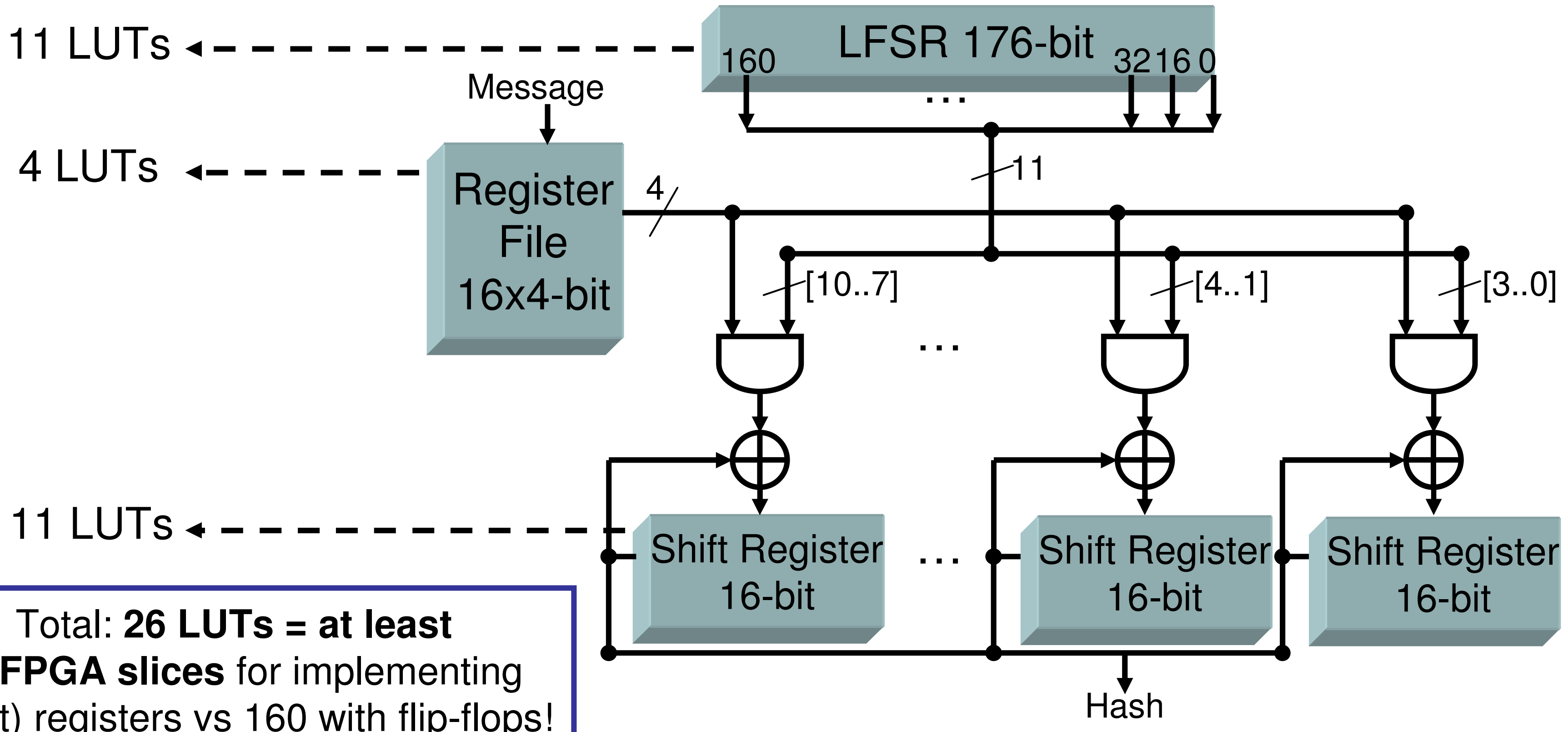
Related Work: Toeplitz Universal Hash

- (2-)Universal Hash Family $H = \{h_i: A \rightarrow B\}_{i=1..n}$
  - ➢ $\forall\ a_1 \neq a_2 \in A$ and $r \leftarrow [1,n]$: $\mathbf{Pr}[h_r(a_1)=h_r(a_2)] \leq |B|^{-1}$
- Multiplication with random Toeplitz matrix is Universal Hash → yields efficient LFSR-based implementation [Krawczyk-Crypto94]
  - Straightforward implementation not optimized for FPGA
  - e.g. if |Message| = 64 bit and |Hash| = 128 bit then: 128-bit LFSR + 64-bit SR + 128-bit accumulator = **320 flip-flops = at least 160 FPGA slices**



Toeplitz Hash: Datapath

- Optimize for FPGA: use resource-efficient shift registers based on Look-up-tables (LUTs)



Total: **26 LUTs = at least 13 FPGA slices** for implementing (shift) registers vs 160 with flip-flops!

Overview

Implementation Results

- Implemented on Xilinx Spartan 3E-500 FPGA
- Compare with best results from [BGSST-CHES2008]
  - Setting:
    - Average response bit error probability: 15%
    - Min-entropy of response bits: 78%
    - Extract 128-bit key
  - Results:

	Proposed SD-HDA Implementation	[BGSST-CHES2008] PUF-optimized DP	[BGSST-CHES2008] HDA-optimized DP
HDA size (slices)	**237**	≥ 907	≥ 429
Cycles	**10298**	≥ 24024	≥ 29925
Performance	205µs @ 50.2MHz	159µs @ 151.5MHz	171µs @ 175.4MHz
SRAM PUF size	**1536 bit**	3696 bit	6160 bit
Helper Data length	13952 bit	3824 bit	6288 bit

Overview

Conclusion

- PUF-based secret key storage is very appealing, but *implementation overhead* should be small!
- Efficient Soft-Decision Decoder reduces min-entropy loss of HDA → ***smaller PUF (-58.4%)***
- Using FPGA-optimized shift registers significantly reduces implementation cost of Universal Hash → ***smaller HDA (-44.8%)***
- Increased Helper Data length and more effort during enrollment are *trade-offs*

Thank you!