



# Error Correction for Physical Unclonable Functions Using Generalized Concatenated Codes

Sven Muelich<sup>\*</sup>, Sven Puchinger<sup>\*</sup>, Martin Bossert<sup>\*</sup>,  
Matthias Hiller<sup>◇</sup>, Georg Sigl<sup>◇</sup>

<sup>\*</sup>Institute of Communications Engineering, Ulm University, Germany

<sup>◇</sup>Institute for Security in Information Technology, TU Munich, Germany

ITG Fachgruppe "Angewandte Informationstheorie"

Munich, October 9, 2014

# Outline

- 1 Motivation
- 2 Physical Unclonable Functions (PUFs)
- 3 Example Code Construction
- 4 Conclusion

## Challenges when implementing a cryptosystem:

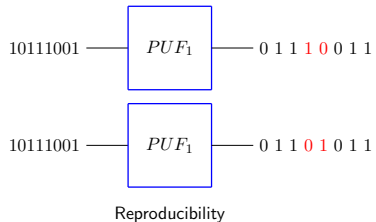
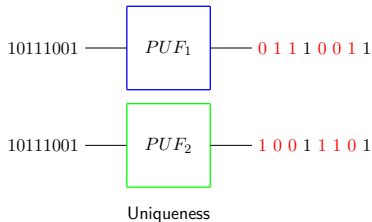
- Secure key generation
  - Random, unique and unpredictable keys
  - Satisfying these properties is hard to achieve
- Secure key storage
  - Key bits in a non-volatile memory
  - Adversaries can gain physical access to (protected) memories

**Physical Unclonable Functions (PUFs) can be used to realize secure key generation and secure key storage**

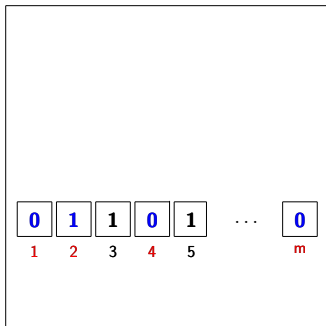
# Physical Unclonable Functions (PUFs)

## What is a PUF?

- Physical entity with challenge-response behavior
- Properties:
  - Uniqueness
  - Reproducibility



# Physical Unclonable Functions (PUFs)



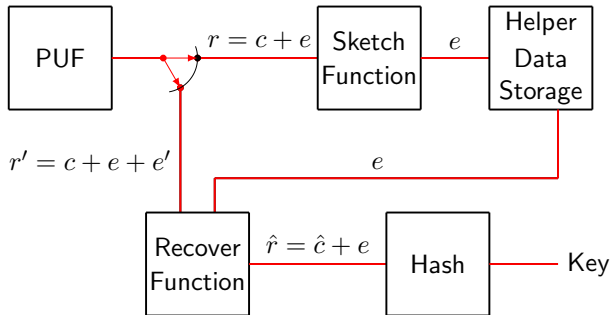
## Example: SRAM PUFs

- device with memory cells
- random initialization when powering on
- randomness static over lifetime
- **Challenge:** Subset of memory cells
- **Response:** Values in selected memory cells

# Physical Unclonable Functions (PUFs)

## Why coding theory?

- Responses are not perfectly reproducible and hence cannot be used as key directly



# Example Code construction

**Challenge: Find good codes for Secure Sketches**

**Constraints:**

- Time and area consumption
- Binary codes
- Dimension  $\geq$  key length
- Codelength as small as possible

# Example Code construction

## Existing scheme given in [Maes2012]<sup>1</sup>:

- Binary Symmetric Channel with  $p = 0.14$
- Generate 128 bit key with block error probability  $P_{\text{err}} = 10^{-9}$
- Concatenation of  $(318, 174, 35)$  BCH code and  $(7, 1, 7)$  code

## What is our goal?

- Generate 128 bit key with block error probability  $P_{\text{err}} < 10^{-9}$
- Improve existing scheme in
  - Codelength
  - Block error probability
  - Simple implementation

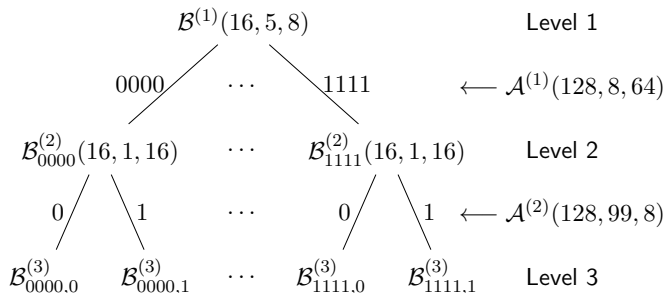
---

<sup>1</sup>R. Maes, A. Herreweghe, I. Verbauwhede, "PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator", CHES, 2012



# Reed-Muller Example Code Construction

## Partitioning:



# Reed-Muller Example Code Construction

## Used decoding methods:

- Generalized Concatenated Codes (GC Codes)
- RM Error Erasure Decoding
- Generalized Minimum Distance (GMD) Decoding

## How good is our code construction?

Code	$P_{\text{err}}$	Length	Largest Field
[Maes2012]	$\approx 10^{-9}$	2226	$\mathbb{F}_{2^8}$ (BCH)
New	$\approx 5.37 \cdot 10^{-10}$	2048	$\mathbb{F}_2$

Thank you for your attention.