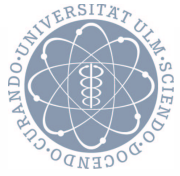


ulm university universität ulm

On Error Correction for Physical Unclonable Functions

Sven Müelich*, Sven Puchinger*, Martin Bossert*,
Matthias Hiller◇, Georg Sigl◇

*Institute of Communications Engineering, Ulm University, Germany
◇Institute for Security in Information Technology, TU Munich, Germany

Rennes, December 12, 2014

Outline

1. Motivation
2. Physical Unclonable Functions (PUFs)
3. Example Code Constructions
   - Reed-Muller Example Code Construction
   - Reed-Solomon Example Code Constructions
4. Conclusion

Motivation

Challenges when implementing a cryptosystem:

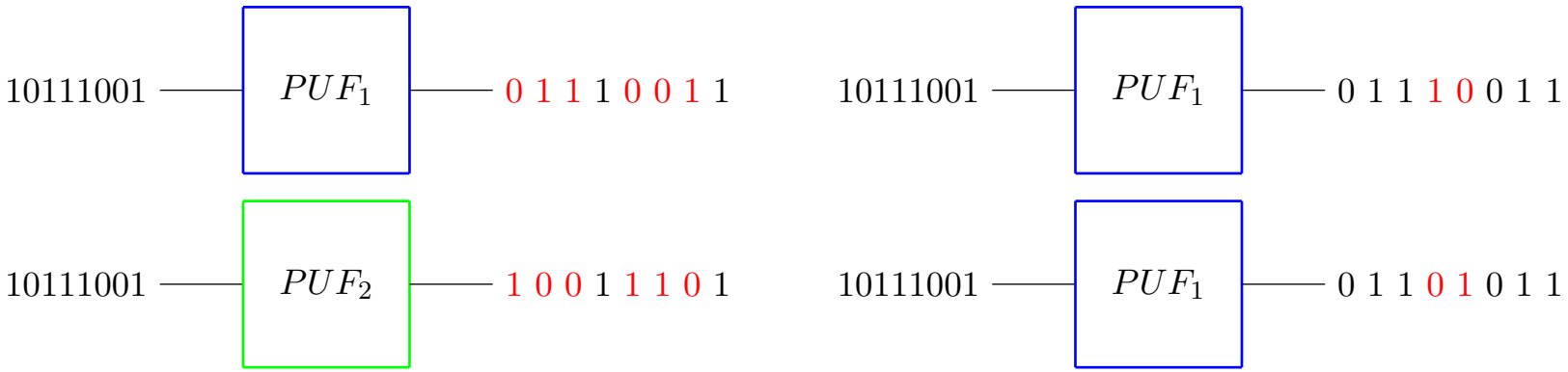
- Secure key generation
  - Random, unique and unpredictable keys
  - Satisfying these properties is hard to achieve
- Secure key storage
  - Key bits in a non-volatile memory
  - Adversaries can gain physical access to (protected) memories

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

Physical Unclonable Functions (PUFs)

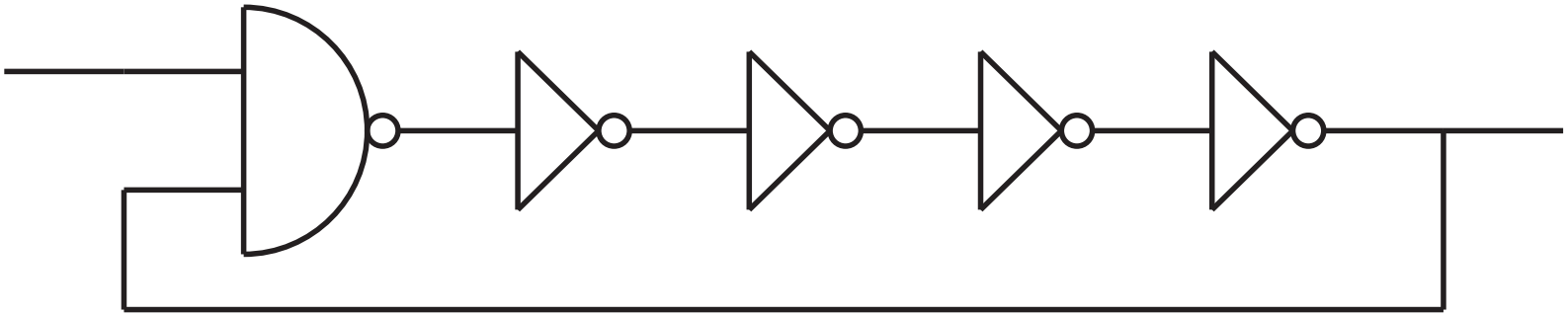
What is a PUF?

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



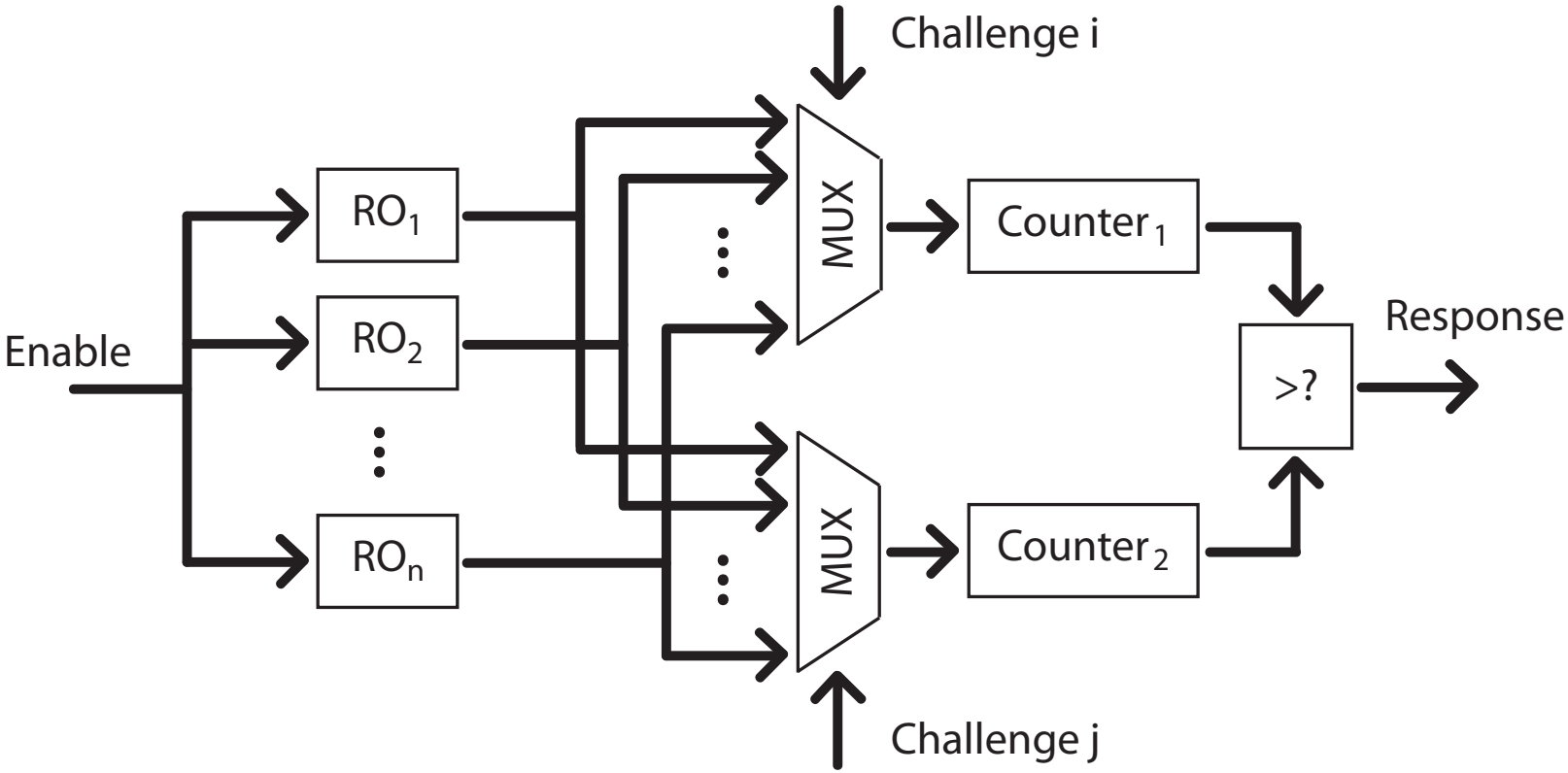
Physical Unclonable Functions (PUFs)

Example: Delay-based Intrinsic PUF

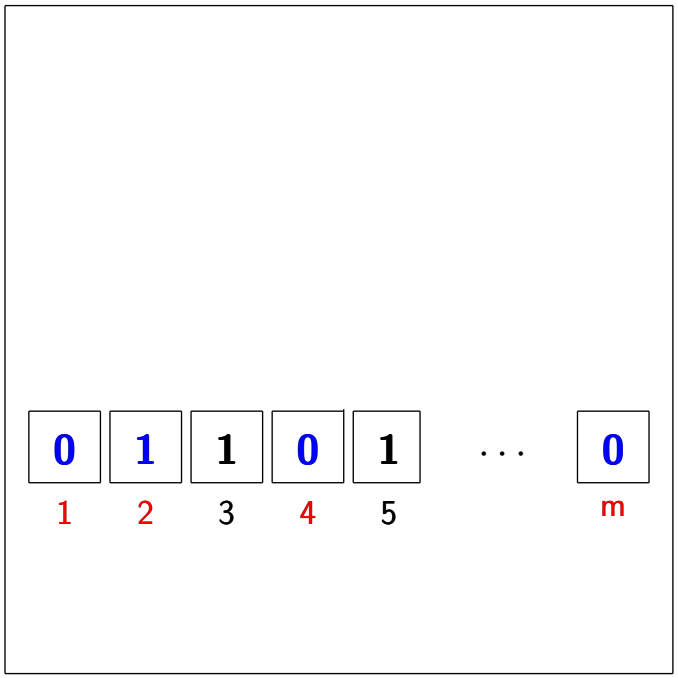


- Ring Oscillator
- Built from logic gates

Physical Unclonable Functions (PUFs)



Physical Unclonable Functions (PUFs)



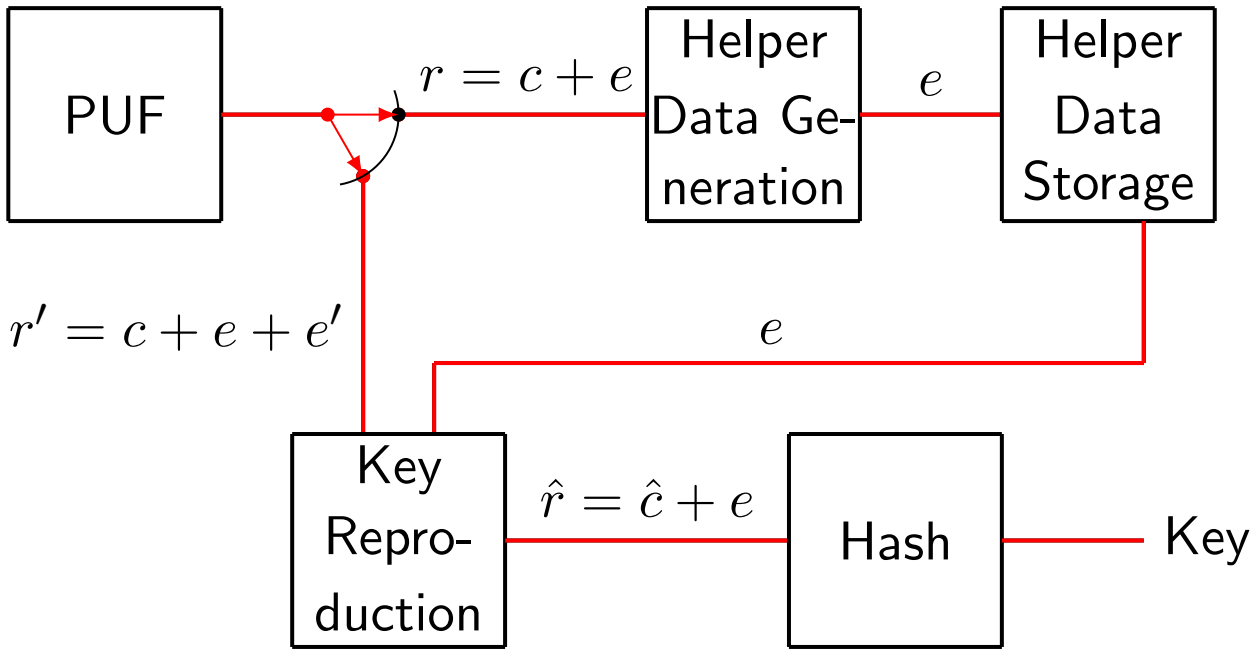
Example: Memory-based Intrinsic PUF

- SRAM PUF, Butterfly PUF, Latch PUF, Flipflop PUF
- 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 Constructions

Challenge: Find good codes for Secure Sketches

Constraints:

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

Typical goal:

- Design code with block error probability $\mathrm{P}_{\mathrm{err}}$ smaller than a certain threshold

Example Code Constructions

Existing scheme given in [Maes2012][1]:

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

What is our goal?

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

[1] R. Maes, A. Herrewege, I. Verbauwhede, "PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator", CHES, 2012

Reed-Muller Codes

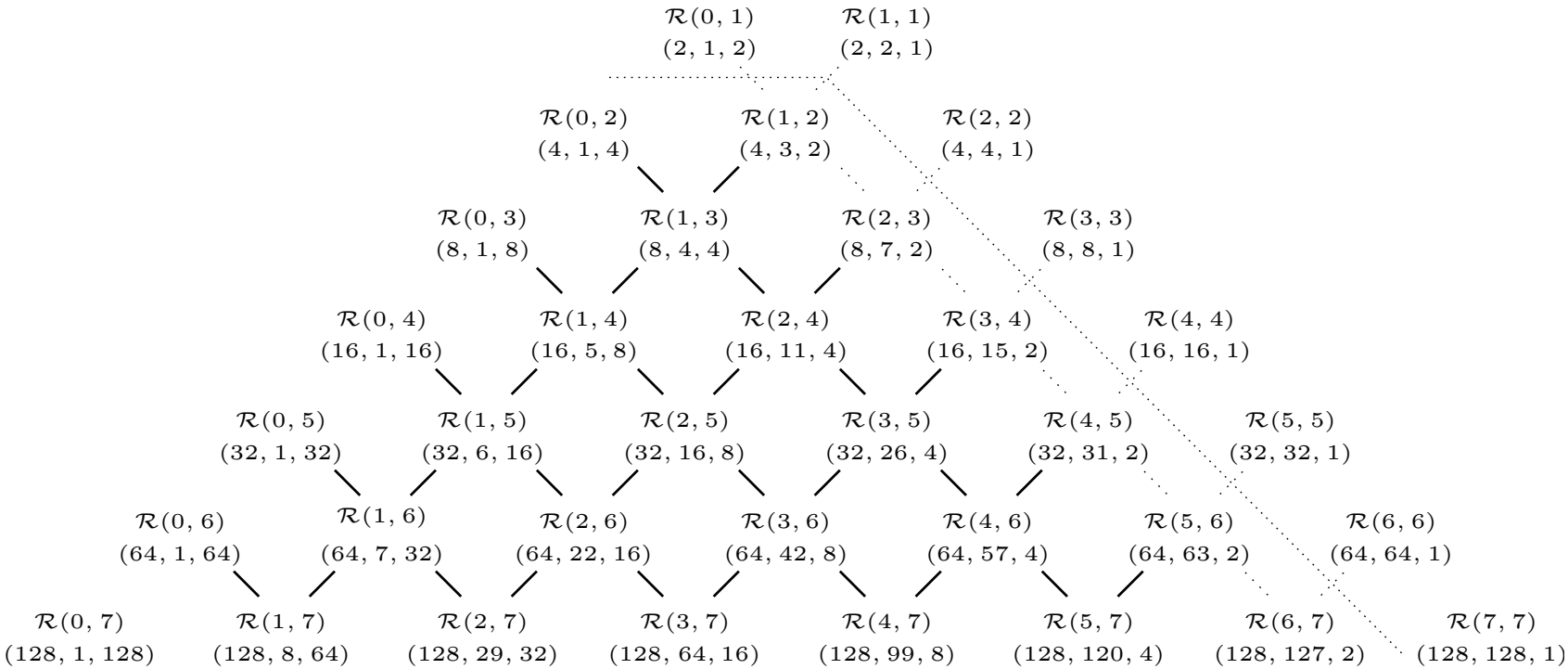
Recursive Construction:

- Reed-Muller code $\mathcal{RM}(r, m)$
  - Binary linear code
  - Order of the code: $r \leq m$
  - Code length: $n = 2^m$
  - Dimension: $k = \sum_{i=0}^{r} \binom{n}{k}$
  - Minimum distance: $d = 2^{m-r}$
- Plotkin Construction

$$\mathcal{RM}(r, m) := \left\{ (\mathbf{a}|\mathbf{a}+\mathbf{b}) : \begin{array}{l} \mathbf{a} \in \mathcal{RM}(r, m-1) \\ \mathbf{b} \in \mathcal{RM}(r-1, m-1) \end{array} \right\}$$

Reed-Muller Codes

Recursive Construction:



Reed-Muller Codes

Why Reed-Muller Codes?

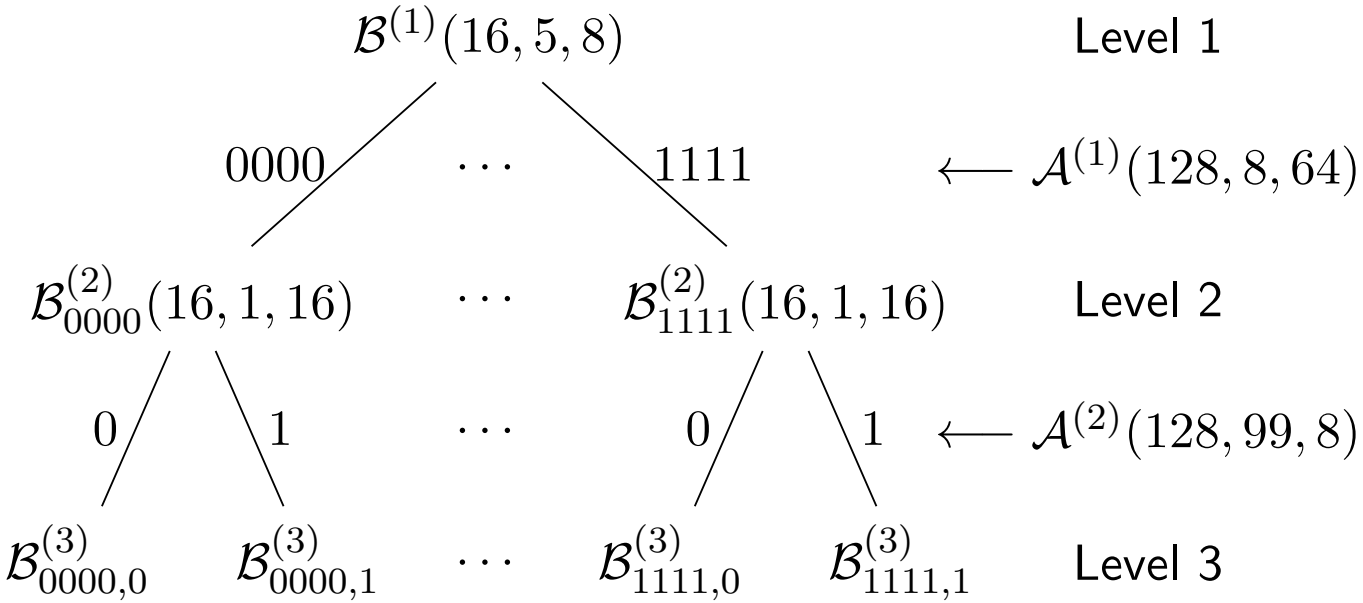
- Decoding easy to implement (recursively)
- Up to τ errors and δ erasures correctable if $2\tau + \delta < d$
  $\Rightarrow$ enables GMD decoding
- Decoding of Repetition and Parity Check Codes as usual

Why GC Codes?

- Simple decoding
- More codewords than an ordinary concatenated code with same parameters n and d
  $\Rightarrow$ higher coderate

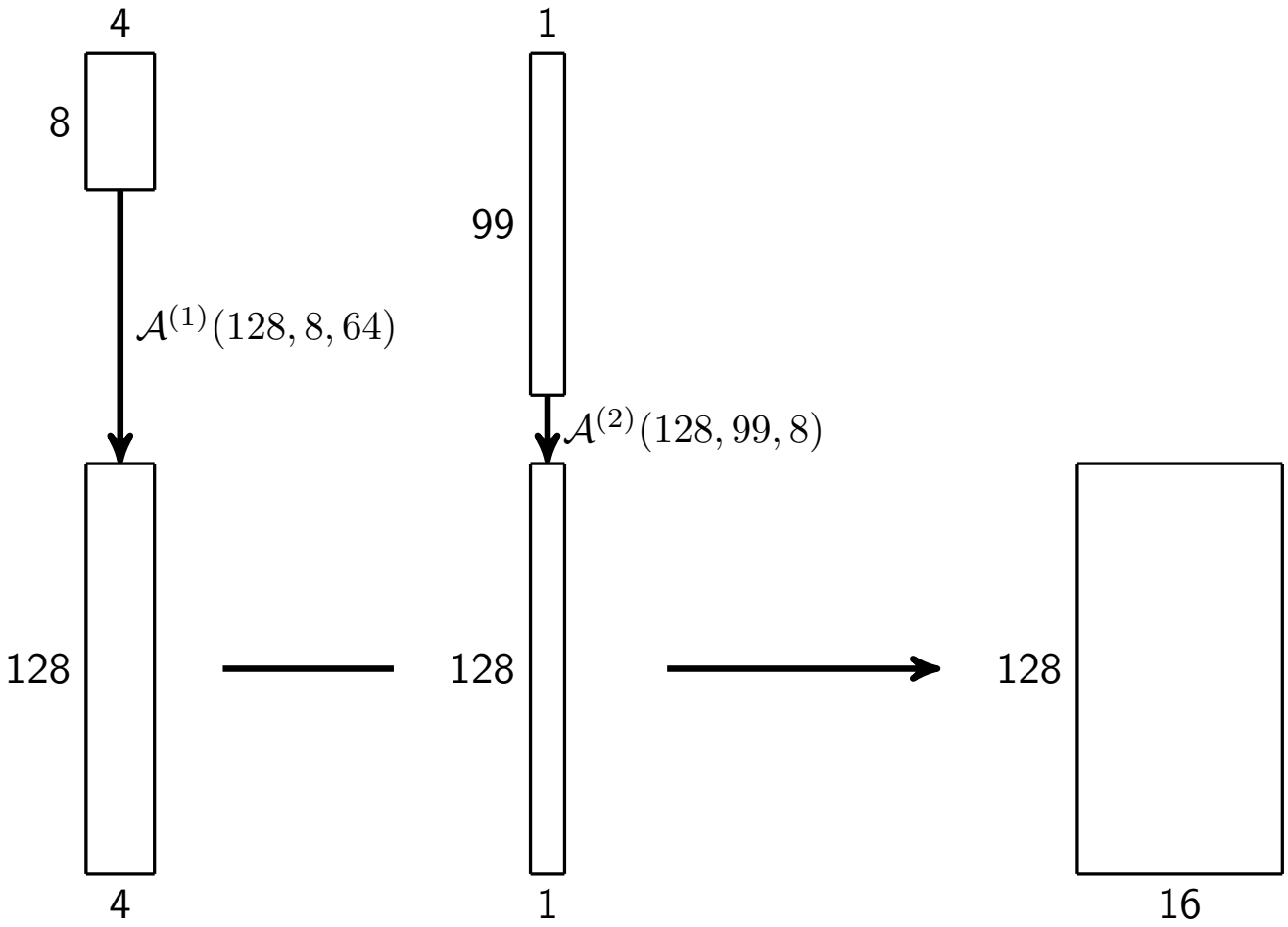
Reed-Muller Example Code Construction

Partitioning:



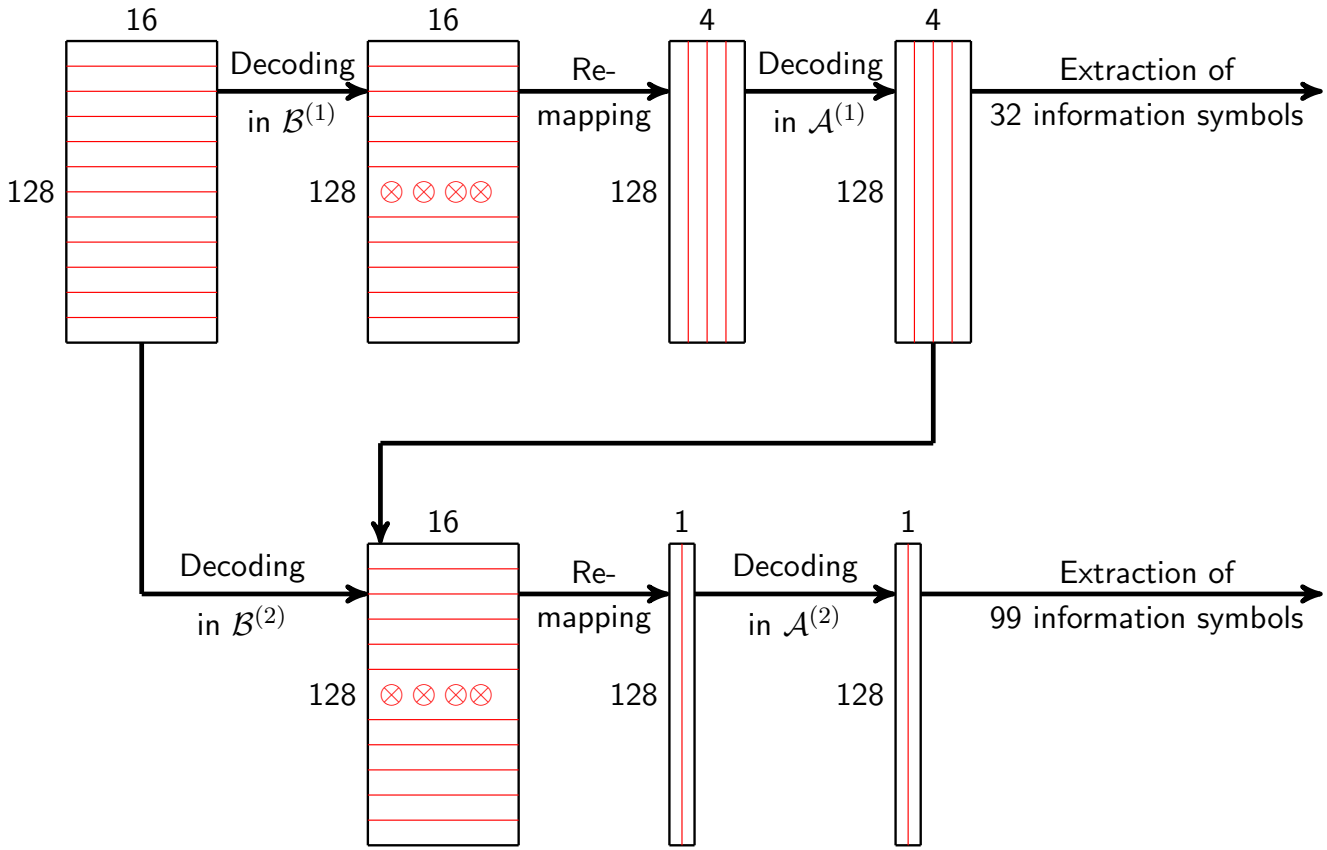
Reed-Muller Example Code Construction

Encoding:



Reed-Muller Example Code Construction

Decoding:



Reed-Muller Example Code Construction

Used decoding methods:

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

Reed-Muller Example: Analysis

Upper bound on $\mathbf{P_{err}}$:

- Event S_1: Decoding fails in step 1.
- Event S_2: Decoding fails in step 2.
- $\mathrm{P}_{\mathrm{err}} = \mathrm{P}(S_1 \cup S_2) \leq \mathrm{P}(S_1) + \mathrm{P}(S_2)$

Transform BSC to binary error and erasure channel

- $\mathrm{P}(\mathrm{error}) = 0.020698$
- $\mathrm{P}(\mathrm{erasure}) = 0.155532$

Reed-Muller Example: Analysis

The error-erasure decoder of the outer $(128, 8, 64)$ code can decode correctly if $2\tau + \delta < 64$

- $$\begin{aligned} \mathrm{P}(S_1) &= \mathrm{P}(2\tau + \delta \geq 64) \\ &= \sum_{i=0}^{128} \mathrm{P}(\delta = i)\mathrm{P}(2\tau \geq 64 - i | \delta = i) \\ &\approx 9.51 \cdot 10^{-12} \end{aligned}$$

Calculate $\mathrm{S_2}$ similarly

- $\mathrm{P}(S_2) \approx 1.48 \cdot 10^{-9}$

Together:

- $\mathrm{P_{err}} \leq 9.51 \cdot 10^{-12} + 1.48 \cdot 10^{-9} \approx 1.49 \cdot 10^{-9}$

Using GMD decoding:

- $\mathrm{P_{err}} \approx 5.37 \cdot 10^{-10}$ (remember goal: $\mathrm{P_{err}} < 10^{-9}$)

Reed-Muller Example: Summary

How good is this code construction?

Code	$\mathrm{P}_{\mathrm{err}}$	Length	Largest Field
BCH Rep.	10^{-9}	2226	$\mathbb{F}_{2^8}$ (BCH)
GC RM	$5.37 \cdot 10^{-10}$	2048	$\mathbb{F}_2$

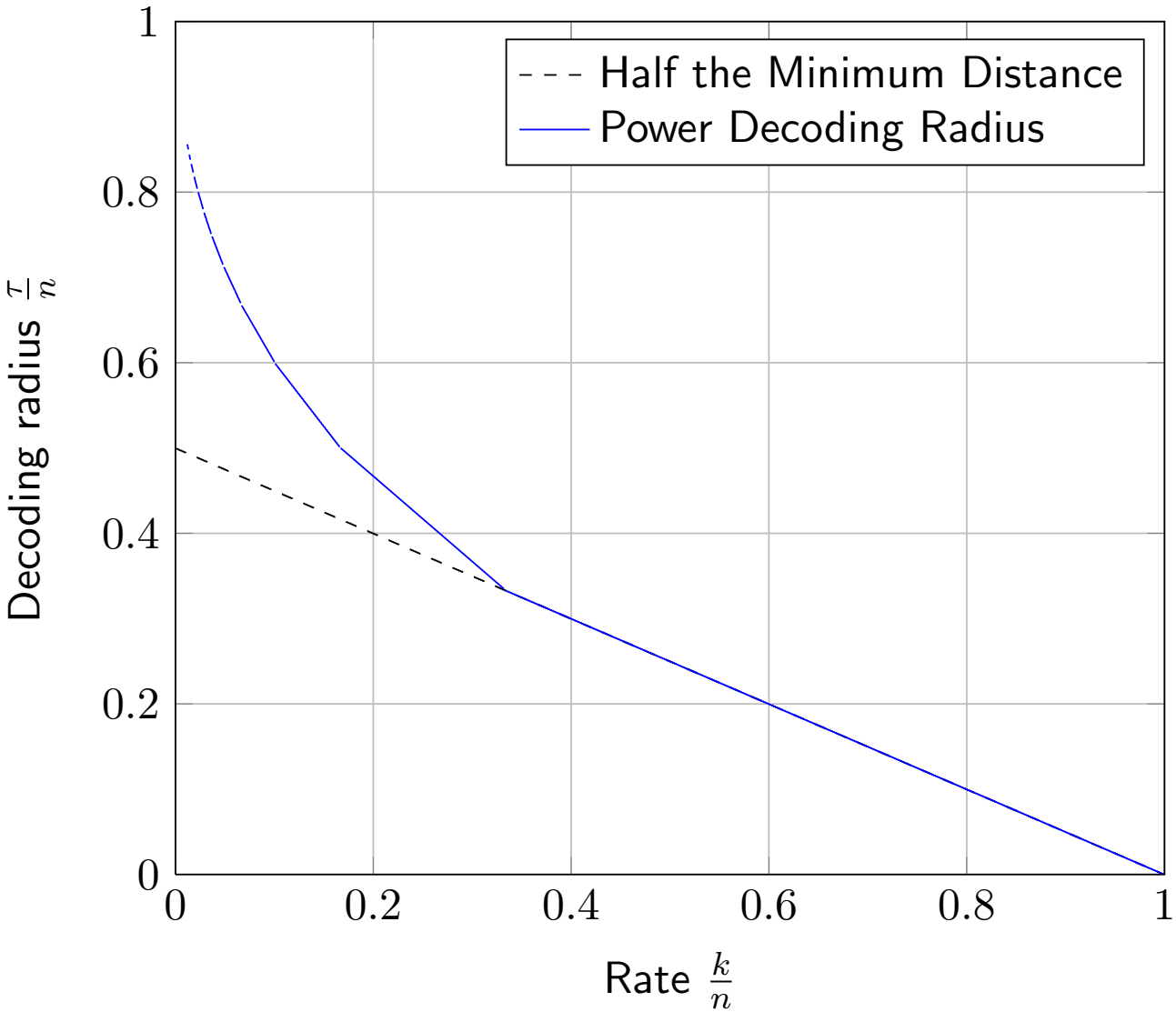
Can we do better?

Reed-Solomon Codes

Evaluation Codes

- $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, all nonzero and $\alpha_i \neq \alpha_j$
- $C = \{(f(\alpha_1), \ldots, f(\alpha_n)) \in \mathbb{F}_q^n | deg(f) < k\}$
- Maximum Distance Separable
- More flexible than Reed-Muller codes

Reed-Solomon Codes: Power Decoding



Reed-Solomon Example Code Constructions

Ordinarily Concatenated Scheme 1

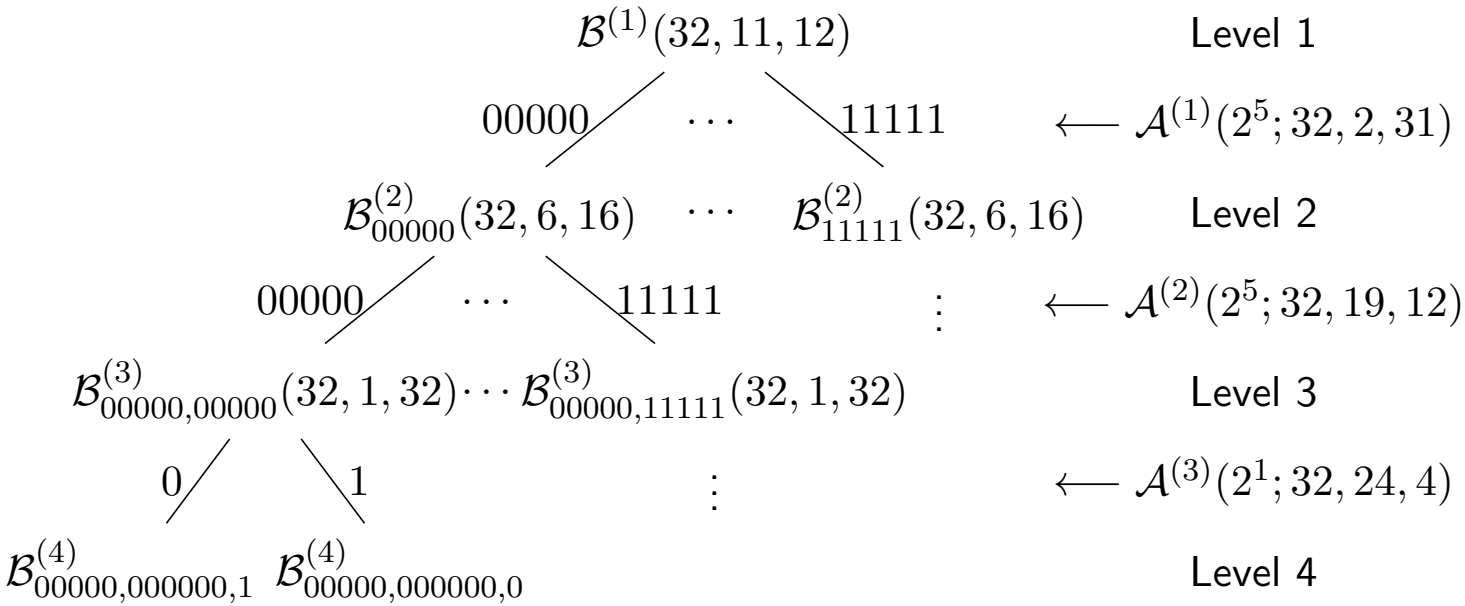
- Inner Code: $\mathcal{RM}(1,5) = \mathcal{C}(32,6,16)$
- Outer Code: $\mathcal{RS}(2^6; 64, 22)$
- $n = 64$ for outer code $\Rightarrow n = 64 \cdot 32 = 2048$
- $\mathrm{P}_{\mathrm{err}} \approx 6.79 \cdot 10^{-37}$

Ordinarily Concatenated Scheme 2

- Inner Code: $\mathcal{RM}(1,5) = \mathcal{C}(32,6,16)$
- Outer Code: $\mathcal{RS}(2^6; 36, 22)$
- Code length can be reduced to $n = 1152$
- $\mathrm{P}_{\mathrm{err}} \approx 1.19 \cdot 10^{-10}$

Reed-Solomon Example Code Constructions

Partitioning:



Encoding and Decoding: Similar to Reed-Muller Example

Reed-Solomon Example: Analysis

Step 1:

- ML decoding inner code $\Rightarrow$ Binary error and erasure channel with $\mathrm{P}(\mathrm{error}) = 0.037808$ and $\mathrm{P}(\mathrm{erasure}) = 0.174488$
- Decoding outer code $\Rightarrow \mathrm{P}(S_1) \approx 1.03 \cdot 10^{-8}$
- Power Decoding $\Rightarrow \mathrm{P}(S_1) \approx 1.48 \cdot 10^{-11}$

Step 2:

- Binary error and erasure channel with $\mathrm{P}(\mathrm{error}) = 0.0032167$ and $\mathrm{P}(\mathrm{erasure}) = 0.0175397$
- $\mathrm{P}(S_2) \approx 3.11 \cdot 10^{-10}$

Reed-Solomon Example: Analysis

Step 3:

- $\mathrm{P}(S_3) \approx 2.13 \cdot 10^{-11}$

Overall block error probability:

- $\mathrm{P}_{\mathrm{err}} \leq \mathrm{P}(S_1) + \mathrm{P}(S_2) + \mathrm{P}(S_3) \approx 3.47 \cdot 10^{-10}$

Advantage of this construction:

- Code length reduced to $n = 32 \cdot 32 = 1024$

Conclusion

How good are our code constructions?

Code	P_{err}	Length	Largest Field
BCH Rep.	10^{-9}	2226	$\mathbb{F}_{2^8}$ (BCH)
GC RM	$5.37 \cdot 10^{-10}$	2048	$\mathbb{F}_2$
RS I	$6.79 \cdot 10^{-37}$	2048	$\mathbb{F}_{2^6}$
RS II	$1.19 \cdot 10^{-10}$	1152	$\mathbb{F}_{2^6}$
GC RS	$3.47 \cdot 10^{-10}$	1024	$\mathbb{F}_{2^5}$

Thank you for your attention.