



Generalized Code Concatenation: Recapitulation and Examples

M.Sc. Sven Muelich

Institute of Communications Engineering, Ulm University, Germany

Channel Coding Lecture

Ulm, January 28, 2015

- 1 Recapitulation of Code Concatenation
- 2 Application Scenario: Physical Unclonable Functions (PUFs)
- 3 Example Code Constructions
 - Reed-Muller Example Code Construction
 - Reed-Solomon Example Code Constructions
- 4 Conclusion

Code Concatenation

Distinguish:

- (Ordinary) Code Concatenation (CC)
- Generalized Code Concatenation (GCC)

Advantages of (Generalized) Code Concatenation:

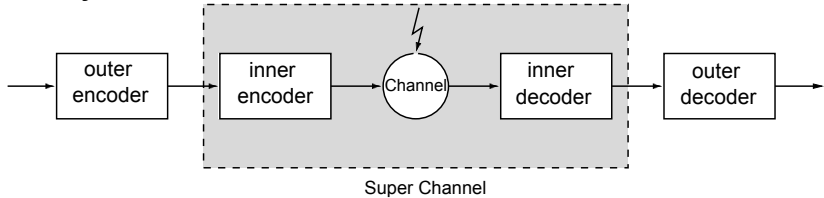
- Very long codes can be constructed by concatenation based on short codes
- Smaller decoding complexity compared to arbitrary code of same length
- Concatenated codes can correct burst errors and independent single errors at the same time

Disadvantage:

- Soft decision decoding has not been satisfactorily solved

(Ordinary) Code Concatenation

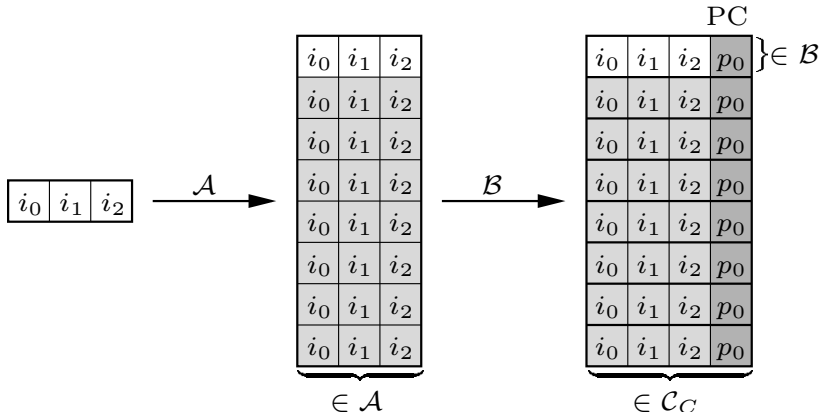
Forney 1966



(Ordinary) Code Concatenation

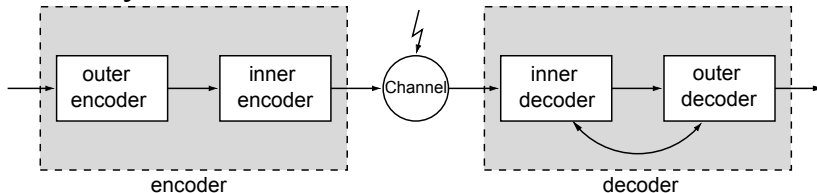
Example:

- Outer Code: $\mathcal{A}(2^3; 8, 1, 8)$
- Inner Code: $\mathcal{B}(2; 4, 3, 2)$
- Concatenated Code: $\mathcal{C}_C(32, 3, 2 \cdot 8 = 16)$



Generalized Code Concatenation

Blokh, Zyablov 1974:



Generalized Code Concatenation

Further advantages:

- Codes with unequal error protection can be constructed (UEP)
- More codewords (and hence a higher coderate) than an ordinary concatenated code with same parameters n and d
- Larger minimum distance than an ordinary concatenated code with same number of codewords (or with same coderate)

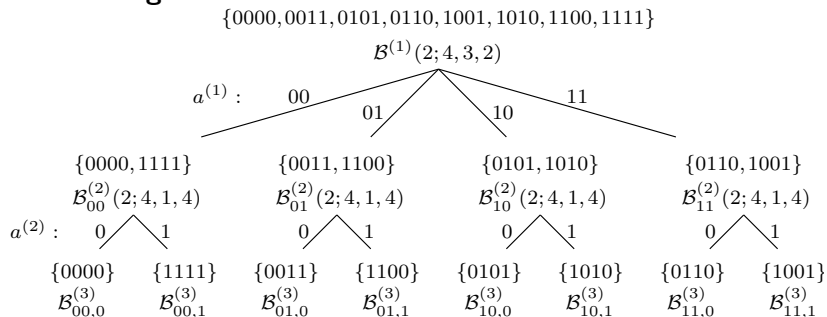
Generalized Code Concatenation

Example:

- Inner Code: $\mathcal{B}(2; 4, 3, 2)$
- Outer Codes: $\mathcal{A}(2^2; 8, 1, 8)$, $\mathcal{A}_H(2; 8, 4, 4)$
- Generalized Concatenated Code: $\mathcal{C}_{GC}(2; 32, 6, 16)$
Note: same minimum distance than before, but $2^6 = 64$ codewords (before: $2^3 = 8$ codewords)

Generalized Code Concatenation

Partitioning of inner code:



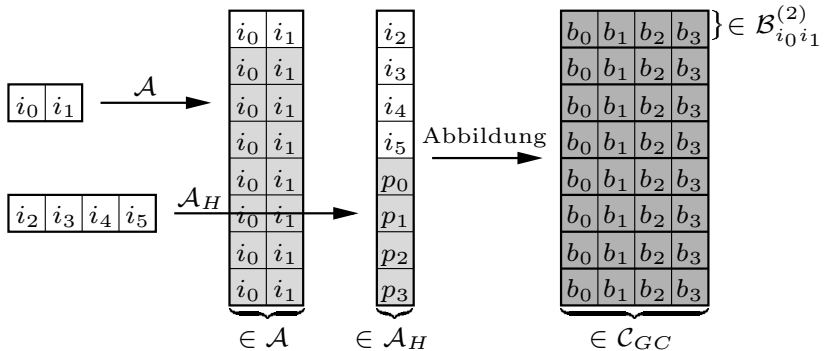
Generalized Code Concatenation

Partitioning of inner code:

- Partition code such that minimum distance in subcodes is as large as possible
- Each codeword can be identified uniquely by the numeration of the path in the partition tree
- Protect numeration with an outer code at each level

Generalized Code Concatenation

Encoding:



Generalized Code Concatenation

Given:

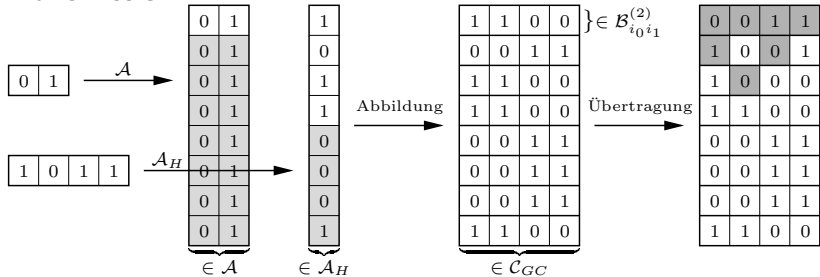
- Inner codes: $\mathcal{B}^{(\ell)}(p; n_i, k_i^{(\ell)}, d_i^{(\ell)})$
- Outer codes: $\mathcal{A}^{(\ell)}(p^m; n_o, k_o^{(\ell)}, d_o^{(\ell)}), \ell = 1, \dots, l$

Parameters of a GCC:

- Length $n = n_i n_o$
- Dimension $k = \sum_{\ell=1}^l k_o^{(\ell)}$
- Minimum Distance $\mathbf{d} \geq \min_{\ell=1, \dots, l} \left\{ \mathbf{d}_i^{(\ell)} \cdot \mathbf{d}_o^{(\ell)} \right\}$

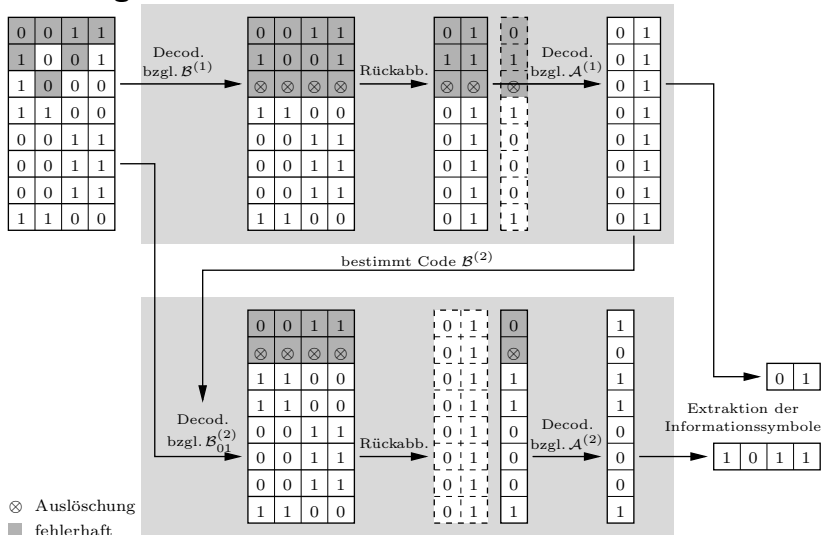
Generalized Code Concatenation

Transmission:



Generalized Code Concatenation

Decoding:



Physical Unclonable Functions (PUFs)

Challenges when implementing a cryptosystem:

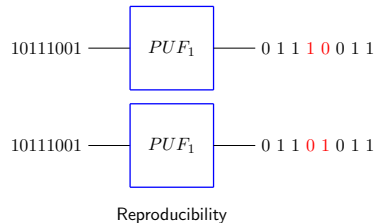
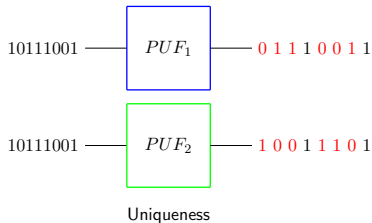
- Secure key generation
 - Random, unique and unpredictable keys
 - Satisfying these properties is hard to achieve
- Secure key storage
 - Key bits in a non-volatile memory
 - Adversaries can gain physical access to (protected) memories

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

Physical Unclonable Functions (PUFs)

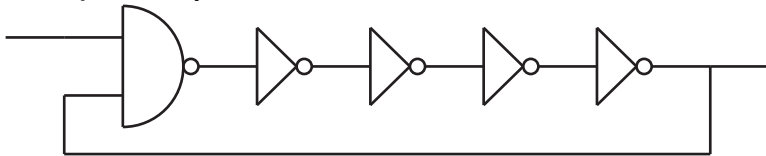
What is a PUF?

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



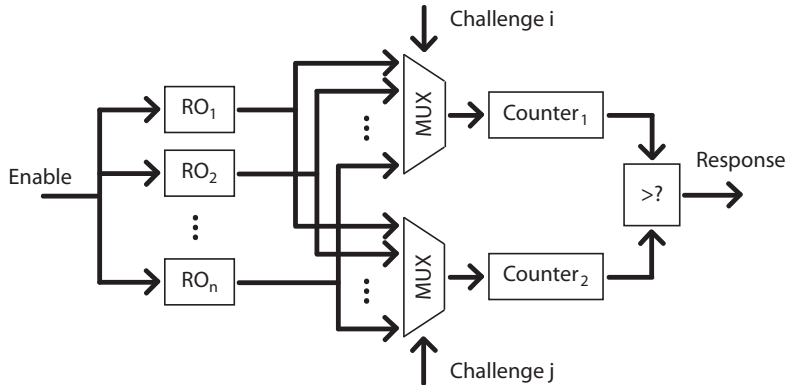
Physical Unclonable Functions (PUFs)

Example: Delay-based Intrinsic PUF



- Ring Oscillator
- Built from logic gates

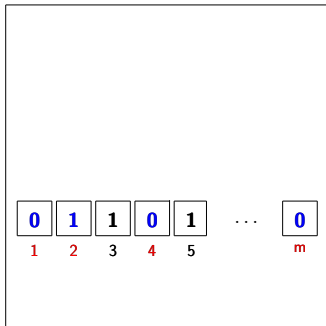
Physical Unclonable Functions (PUFs)



Physical Unclonable Functions (PUFs)

Example: Memory-based Intrinsic PUF

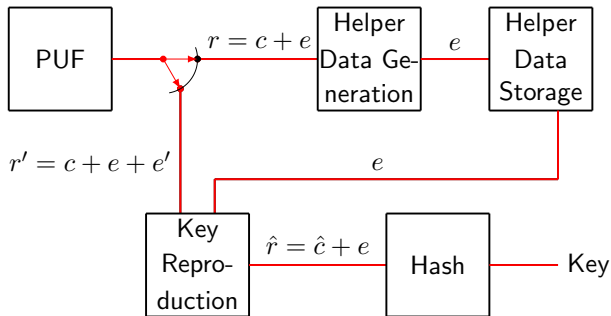
- SRAM PUF, Butterfly PUF, Latch PUF, Flipflop PUF
- device with memory cells
- random initialization when powering on
- randomness static over lifetime
- **Challenge:** Subset of memory cells
- **Response:** Values in selected memory cells



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 P_{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 $P_{\text{err}} = 10^{-9}$
- 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

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

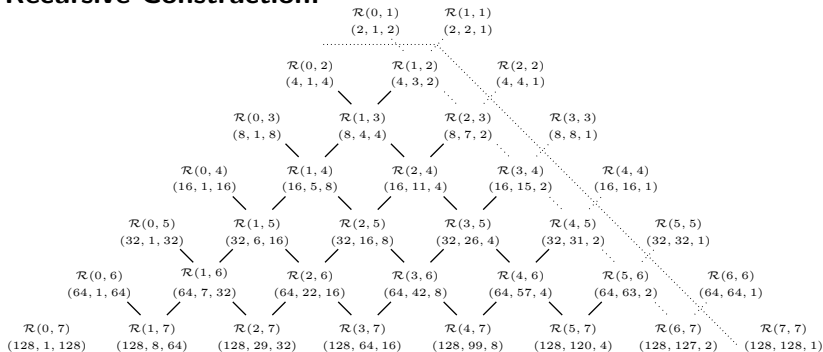
Recursive Construction:

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



Why Reed-Muller Codes?

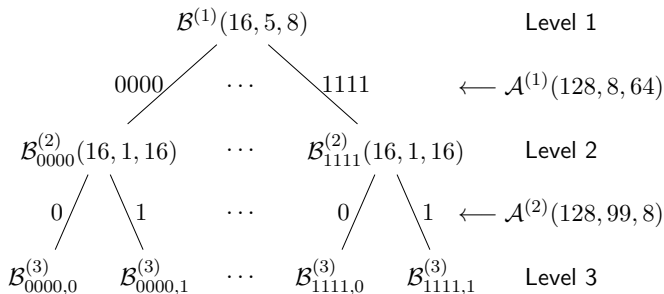
- Decoding easy to implement (recursively)
- Up to τ errors and δ erasures correctable if $2\tau + \delta < d$
 $\Rightarrow$ enables GMD decoding
- Decoding of Repetition and Parity Check Codes as usual

Why GC Codes?

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

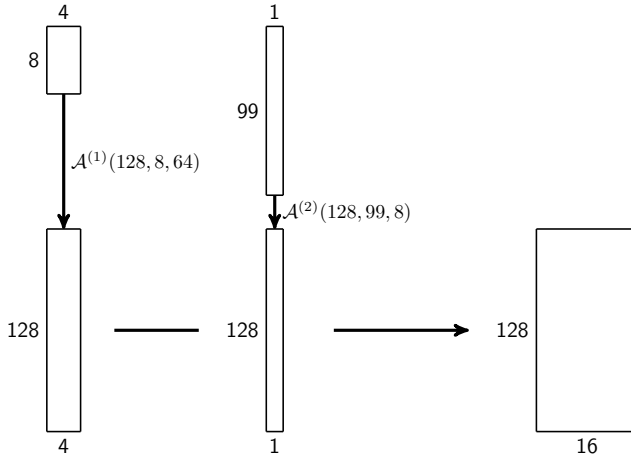
Reed-Muller Example Code Construction

Partitioning:



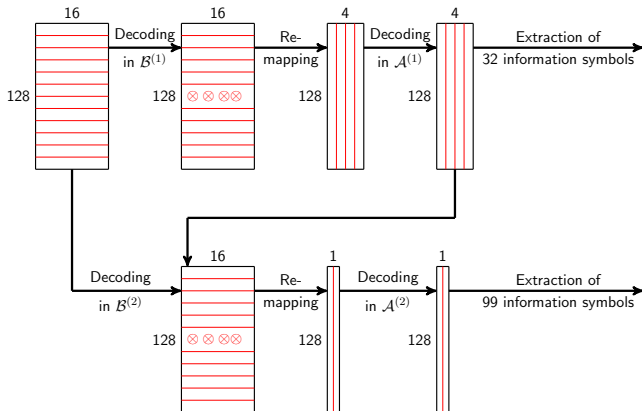
Reed-Muller Example Code Construction

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 P_{err} :

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

Transform BSC to binary error and erasure channel

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

Reed-Muller Example: Analysis

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

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

Calculate S_2 similarly

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

Together:

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

Using GMD decoding:

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

Reed-Muller Example: Summary

How good is this code construction?

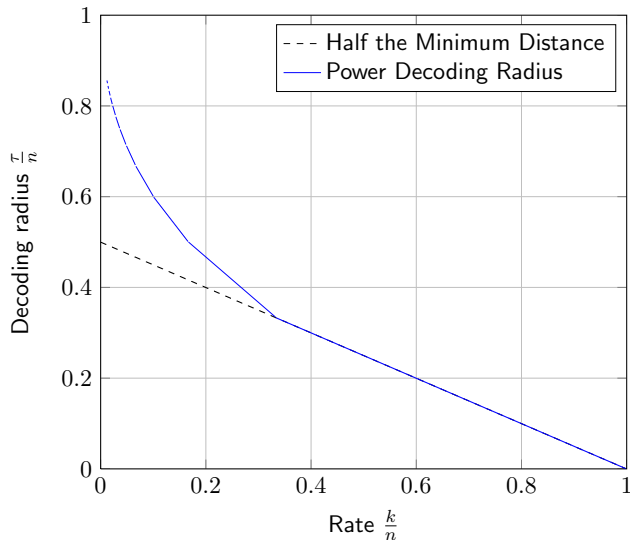
Code	P_{err}	Length	Largest Field
BCH Rep.	10^{-9}	2226	$\mathbb{F}_{2^8}$ (BCH)
GC RM	$5.37 \cdot 10^{-10}$	2048	$\mathbb{F}_2$

Can we do better?

Evaluation Codes

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

Reed-Solomon Codes: Power Decoding



Reed-Solomon Example Code Constructions

Ordinarily Concatenated Scheme 1

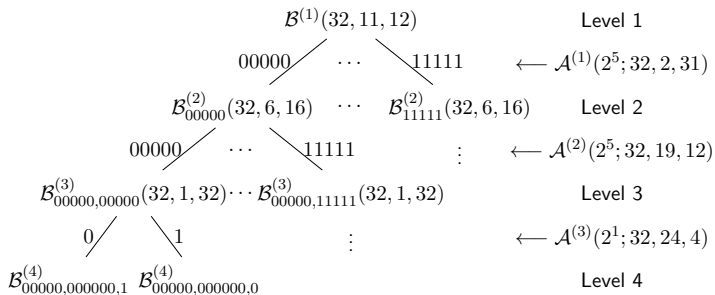
- Inner Code: $\mathcal{RM}(1, 5) = \mathcal{C}(32, 6, 16)$
- Outer Code: $\mathcal{RS}(2^6; 64, 22)$
- $n = 64$ for outer code $\Rightarrow n = 64 \cdot 32 = 2048$
- $P_{\text{err}} \approx 6.79 \cdot 10^{-37}$

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 $n = 1152$
- $P_{\text{err}} \approx 1.19 \cdot 10^{-10}$

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 $P(\text{error}) = 0.037808$ and $P(\text{erasure}) = 0.174488$
- Decoding outer code $\Rightarrow P(S_1) \approx 1.03 \cdot 10^{-8}$
- Power Decoding $\Rightarrow P(S_1) \approx 1.48 \cdot 10^{-11}$

Step 2:

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

Reed-Solomon Example: Analysis

Step 3:

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

Overall block error probability:

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

Advantage of this construction:

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

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	$\mathbb{F}_{2^5}$

Generalized Code Concatenation

- Martin Bossert, Channel Coding for Telecommunications, John Wiley & Sons, 1999
- Martin Bossert, Kanalkodierung, 3. Auflage, Oldenbourg Wissenschaftsverlag, 2013

Physical Unclonable Functions

- Roel Maes, Physically Unclonable Functions: Constructions, Properties and Applications (Dissertation), 2012

Example Code Constructions presented in this lecture

- S. Muelich, S. Puchinger, M. Bossert, M. Hiller, G. Sigl Error Correction for Physical Unclonable Functions Using Generalized Concatenated Codes, Svetlogorsk 2014
- S. Muelich, S. Puchinger, M. Bossert, M. Hiller, G. Sigl On Error Correction for Physical Unclonable Functions, Hamburg 2015