

**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

BACHELOR THESIS

Petr Sedláček

Limitations of incompressible encodings

Computer Science Institute of Charles University

Supervisor of the bachelor thesis: Mgr. Pavel Hubáček, Ph.D.

Study programme: Mathematics

Study branch: General Mathematics

Prague 2021

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature, and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular, the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date

Author's signature

I dedicate this work to my bigger half.

Title: Limitations of incompressible encodings

Author: Petr Sedláček

Department: Computer Science Institute of Charles University

Supervisor: Mgr. Pavel Hubáček, Ph.D., Computer Science Institute of Charles University

Abstract: This thesis studies the limitations of incompressible encodings with information-theoretic security. We demonstrate a flaw in the existing proof of the impossibility of constructing incompressible encodings information-theoretically. Our main contribution is a full proof of impossibility of existence of non-trivial information-theoretically secure incompressible encoding schemes. In the first part of the thesis, we introduce the basics of incompressible encodings and provide the necessary definitions. Next, we present the flaws in the existing argument and provide explicit counterexamples to them. Throughout the rest of the thesis, we gradually construct a complete proof. We start by showing the impossibility under a few additional restrictions on the correctness and structure of the schemes that we subsequently remove one by one. Finally, we present an adversary able to break any non-trivial incompressible encoding scheme.

Keywords: incompressible encodings, plain model, information-theoretic security

Contents

1	Introduction	1
1.1	Our contributions	1
2	Incompressible encodings	3
3	Issues in the proof-sketch in [MW20]	6
3.1	Counterexample	6
4	Impossibility of information-theoretic security	8
4.1	Schemes with perfect correctness and deterministic decoding	8
4.2	Schemes with imperfect correctness and deterministic decoding . . .	10
4.3	Schemes with imperfect correctness and probabilistic decoding . . .	10
5	Conclusions	16

1 Introduction

The focus of this work is a primitive called *incompressible encoding scheme*, recently introduced by Moran and Wichs [MW20].

An incompressible encoding scheme gets some data as input message and encodes it into an effectively incompressible code word. The original data can be easily decoded from the code word by anyone. However, it is not possible for any probabilistic polynomial-time adversary to compress the code word so that they can later decompress it to the code word, even knowing the original message.

In their article, Moran and Wichs provided constructions of incompressible encodings under various computational hardness assumptions as well as negative results regarding their security. They also studied various other properties of incompressible encodings, such as their composability.

Even though the incompressibility property of the incompressible encodings may seem pointless, Moran and Wichs demonstrated that they have interesting practical applications.

Proofs of replicated storage without timing assumptions. Incompressible encodings can be used as *replica encodings*. Replica encodings are important building block for proofs of replicated storage without timing assumptions, which were presented by Damgård, Ganesh, and Orlandi [DGO19]. It is a new technique that allows to check whether a server stores multiple copies of the same file and does not store only one copy. Other types of proofs of replicated storage are currently used in Filecoin [Lab17] - a decentralized storage network.

Big-key cryptography with useful keys. Big-key cryptography in the bounded-retrieval model [Dzi06] uses large keys (gigabytes or terabytes in size) that are hard to exfiltrate. That means that any attacker cannot obtain the key without a massive (internet) traffic that can be easily detected and aborted. Big-key encryption schemes remain secure even if part of the secret key was exfiltrated. The incompressibility property of any incompressible encoding scheme allows to use an encoding of arbitrary data (e.g. collection of family videos) as the key and, thus, saving up gigabytes of space that would be occupied by a regular key without any additional purpose.

1.1 Our contributions

In this work, we deepen our knowledge about the limitations of the incompressible encodings constructions. Moran and Wichs conjectured that it is not possible to create a non-trivial information-theoretically secure incompressible encoding, i.e., without any computational restrictions on the adversary. In this work, we show that the sketch of a proof given in [MW20] has a few significant issues, which we describe in detail in Section 3. We present our own full proof of the impossibility of existence of information-theoretically secure incompressible encodings in the plain model in Section 4.

Our results are somewhat analogous to the limitation of perfectly secure symmetric encryption. The one time pad scheme is secure against unbounded attackers, but the key must be as long as the message. It was proven by Shannon [Sha49] that this cannot be improved - every symmetric encryption scheme resilient to unbounded

attackers cannot have the key shorter than the message. That gives solid ground to the studies of “weaker” encryption schemes that rely on computational hardness assumptions and are secure only against computationally bounded adversaries.

Our full proof of the impossibility of non-trivial incompressible encoding scheme resilient against computationally unbounded adversaries substantiates focused research of incompressible encodings relying on computational hardness assumptions.

2 Incompressible encodings

We use $\lambda \in \mathbb{N}$ to denote the security parameter.

Definition 2.1. An *Encoding scheme* $\Pi = (\text{Enc}, \text{Dec})$ for a message space $M = \{m; m \in \{0, 1\}^k\}$ consist of a pair of *Probabilistic Polynomial Time* (PPT) algorithms *Encode* (Enc) and *Decode* (Dec).

Definition 2.2 (p -correctness). Let $p: \mathbb{N} \rightarrow [0, 1]$ be a function. An encoding scheme $\Pi = (\text{Enc}, \text{Dec})$ is p -correct if: $\forall \lambda \in \mathbb{N}, \forall m \in M, \Pr[\text{Dec}(\text{Enc}(1^\lambda, m)) = m] \geq p(\lambda)$.

Definition 2.3. We define a *negligible function* as a function $f: \mathbb{N} \rightarrow \mathbb{R}$ satisfying that for every $n \in \mathbb{N}$ there exists $K_n \in \mathbb{N}$ such that for every $\lambda \in \mathbb{N}, \lambda \geq K_n$ it holds that $\lambda^{-n} > |f(\lambda)|$. We denote the set of all negligible functions as $\text{negl}(\lambda)$.

Definition 2.4 ((α, β) -incompressibility). Let $\alpha, \beta: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ be functions. Then encoding scheme $\Pi = (\text{Enc}, \text{Dec})$ is (α, β) -incompressible if the following holds:

1. α -Bounding: $\forall m \in \{0, 1\}^k, \forall \lambda \in \mathbb{N} : \Pr[|\text{Enc}(1^\lambda, m)| \leq \alpha(\lambda, k)] = 1$
2. β -Incompressibility: For each adversary $\mathcal{A} = (\mathcal{A}.\text{Select}, \mathcal{A}.\text{Compress}, \mathcal{A}.\text{Expand})$, it holds that $\Pr[\text{CompExp}_{\mathcal{A}, \beta}^\Pi(1^\lambda) = 1] \in \text{negl}(\lambda)$, where the $\text{CompExp}_{\mathcal{A}, \beta}^\Pi$ denotes a compression experiment defined in Figure 2.1 and visualized in Figure 2.2.

Consider the following construction. We can append a random string of length r to any message m of length k . The random part cannot be compressed and, thus, $\alpha(k) = k + r$ and $\beta(k) = r$. Therefore, encoding schemes satisfying $\beta \leq \alpha(k) - k$ are easy to obtain. From now on, we will focus solely on *non-trivial* encoding schemes satisfying $\beta(k) > \alpha(k) - k$.

Definition 2.5. A non-trivial (α, β) -incompressible encoding scheme is a p -correct (α, β) -incompressible encoding scheme with $\beta(\lambda, k) > \alpha(\lambda, k) - k$, and $(1 - p(\lambda)) \in \text{negl}(\lambda)$.

In the rest of the work we rely on the following notation.

Notation 2.6.

For $k, \lambda \in \mathbb{N}$:

- $M := \{0, 1\}^k$ is a set of all possible messages with length equal to k .
- $W := \bigcup_{i=0}^{\alpha(\lambda, k) - k} \{0, 1\}^i$ is a set of all bit strings with a length less than or equal to $\alpha(\lambda, k) - k$.
- $C := \{c \in \bigcup_{i=0}^{\alpha(\lambda, k)} \{0, 1\}^i; \exists m \in \{0, 1\}^k, \Pr[\text{Enc}(1^\lambda, m) = c] > 0\}$ is the set of all possible code words for messages of a length k with respect to the security parameter λ .

Figure 2.1: Compression experiment $\text{CompExp}_{\mathcal{A},\beta}^{\Pi}(1^\lambda)$

For all encoding schemes $\Pi = (\text{Enc}, \text{Dec})$, and adversaries $\mathcal{A} = (\mathcal{A}.\text{Select}(1^\lambda), \mathcal{A}.\text{Compress}(\text{aux}, c), \mathcal{A}.\text{Expand}(\text{aux}, w))$, and $\beta = \beta(\lambda, k)$, the compression experiment $\text{CompExp}_{\mathcal{A},\beta}^{\Pi}(1^\lambda)$ is defined as follows:

$\text{CompExp}_{\mathcal{A},\beta}^{\Pi}(1^\lambda)$

1. $(m, \text{aux}) \leftarrow \mathcal{A}.\text{Select}(1^\lambda)$.
2. $c \leftarrow \text{Enc}(1^\lambda, m)$.
3. $w \leftarrow \mathcal{A}.\text{Compress}(\text{aux}, c)$.
4. $c' \leftarrow \mathcal{A}.\text{Expand}(\text{aux}, w)$.
5. Output 1 if and only if $c = c'$ and $|w| \leq \beta(\lambda, |m|)$.

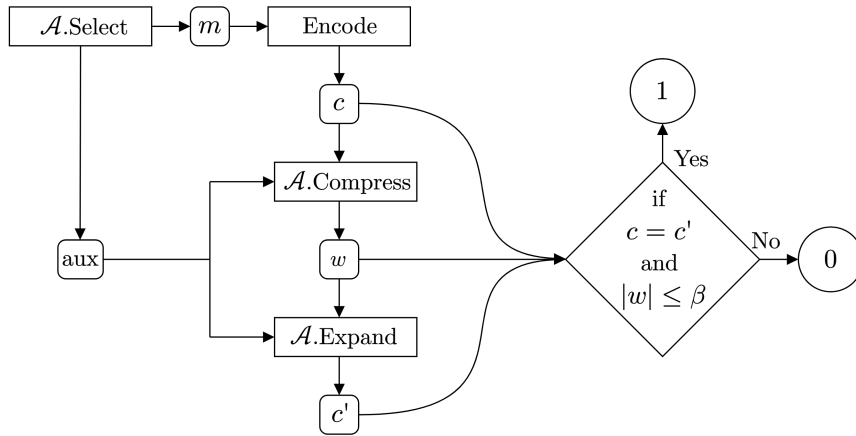


Figure 2.2: Compression experiment flow

For $m \in \{0, 1\}^k$, $\lambda \in \mathbb{N}$:

- $C_m := \{c \in C; \Pr[\text{Enc}(1^\lambda, m) = c] > 0\}$ is the set of all possible code words of the message m with respect to the security parameter λ .
- $D_m := \{c \in C; \text{Dec}(c) = m\}$ is the set of all code words that are decoded to the message m .

The use of the notation is depicted in Figure 2.3. The message m_1 can be encoded to any code word c from C_{m_1} , which is a subset of the set of all code words C . Every code word from the set D_{m_2} is decoded to the message m_2 .

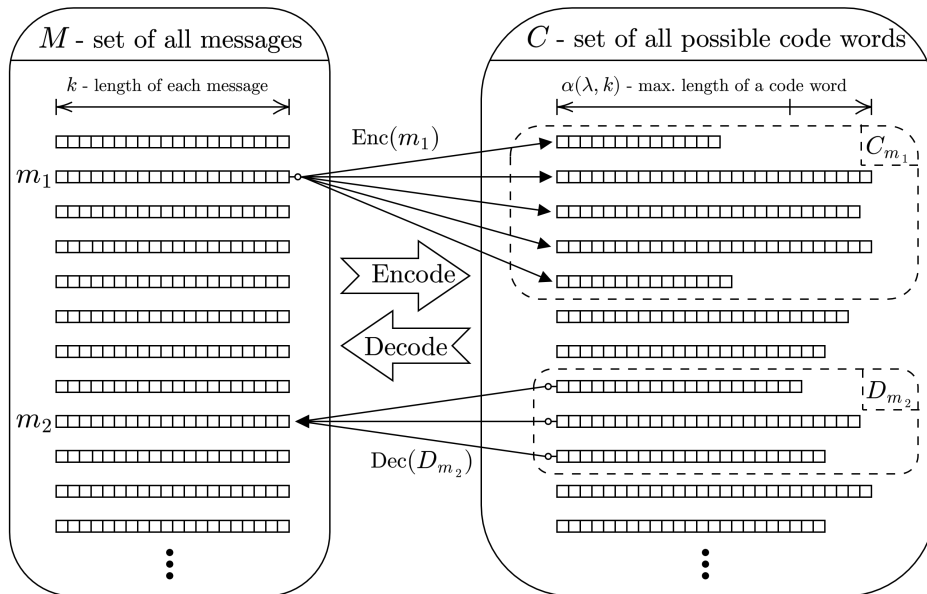


Figure 2.3: Encoding and decoding sketch

3 Issues in the proof-sketch in [MW20]

Statement 3.1 (Original sketch of proof by Moran and Wichs). *“It is easy to see that non-trivial incompressible encodings cannot be constructed information theoretically. This is because, there are at most $2^{\alpha(k)-k}$ possible codewords per message on average, and therefore also certainly for the worst-case message m . A pair of inefficient compression/ decompression procedures can enumerate the list of all such codewords (e.g., in lexicographic order) and compress/decompress any codeword in the list just by writing down its index using $\beta(k) = \alpha(k) - k$ bits.”*

Statement 3.2 (Decomposition). In their article, Moran and Wichs presented a broad definition of incompressible encodings and provided two possible constructions. The statement above holds for those constructions, but it does not hold for every construction possible that the definition allows. We identified two main issues.

The problematic part of the sketch is this sentence: *“This is because, there are at most $2^{\alpha(k)-k}$ possible code words per message on average, and therefore also certainly for the worst-case message m ”*. The first issue is that with probabilistic encoding each message can be encoded to any number of code words with non-zero probability, although to satisfy the *p-correctness* some of those probabilities must be negligible. We provide a counterexample in the next section.

The second issue is that the notation of $\alpha(k)$ allows different lengths of code words. Therefore, there are more than $2^{\alpha(k)}$ possible code words. This issue is a technical one and can be solved in the following way. The adversary \mathcal{A} is allowed to use the length of a bit sequence as an additional information, i.e., \mathcal{A} is able to distinguish between “101” and “00101”. Then the effects of the variable code word length cancel out (with slight technical difficulties). However, both of those issues show that the sentence is unsubstantiated because there could be more than $2^{\alpha(k)-k}$ possible code words per message on average.

3.1 Counterexample

Idea. We leverage an arbitrary encoding scheme and “weaken” its encoding algorithm in a way that allows every message to be encoded into any possible code word. Clearly, this scheme is of no practical interest but it is an explicit counterexample to the assumptions of [MW20]

Lemma 3.3. *For all p -correct (α, β) -incompressible encoding scheme Π there exists a \hat{p} -correct (α, β) -incompressible encoding scheme $\hat{\Pi}$, such that $(p - \hat{p}) \in \text{negl}(\lambda)$ and for all $\lambda \in \mathbb{N}$, $m \in \{0, 1\}^k$, $C_m = C$, i.e., each message can be encoded into any possible code word with a non-zero probability.*

Proof. Let $\Pi = (\text{Enc}, \text{Dec})$ be a p -correct, (α, β) -incompressible encoding scheme. For this example, we need an arbitrary negligible function, which we denote as $q(\lambda)$.

We define the encoding algorithm $\widehat{\text{Enc}}$ in the following way. Before encoding a message m , $\widehat{\text{Enc}}$ generates a uniformly random number $p \in [0, 1]$. Then it encodes the message m in the following way:

$$\widehat{\text{Enc}}(m) = \begin{cases} \text{Enc}(m) & \text{if } p \in (q(\lambda), 1], \\ \widehat{c}, & \text{if } p \in [0, q(\lambda)], \end{cases}$$

where \widehat{c} is a uniformly chosen random code word satisfying $|\widehat{c}| \leq \alpha(\lambda, k)$ generated “on the fly”.

$\widehat{\text{Enc}}(m)$ is correctly defined, because $\forall \lambda > 1, q(\lambda) \in [0, 1)$. Next, we discuss the properties of $\widehat{\Pi} = (\widehat{\text{Enc}}, \text{Dec})$.

First, we verify that $\widehat{\Pi}$ is indeed an (α, β) -incompressible encoding scheme. The α -bounding property holds by the definition of $\widehat{\text{Enc}}$. Next, we focus on β -incompressibility. We know from the definition of Π that an arbitrary adversary \mathcal{A} wins the compression experiment against the Enc algorithm with at most negligible probability. In the $\widehat{\Pi}$ scheme, the probability of the message being encoded as a randomly chosen code word is negligible. In all other cases, the message is encoded via the Enc algorithm. Therefore, the chance of success of the adversary \mathcal{A} against the $\widehat{\text{Enc}}$ algorithm is also negligible.

The $\widehat{\Pi}$ scheme is \widehat{p} -correct, where $\widehat{p}(\lambda) \geq (1 - q(\lambda))p(\lambda) = p(\lambda) - p(\lambda)q(\lambda)$ because the Dec algorithm correctly decodes a code word encoded by Enc with the probability $p(\lambda)$ and the Enc algorithm is used for encoding with the probability $(1 - q(\lambda))$. It follows from the definition of negligible function that $p(\lambda)q(\lambda)$ is negligible. Thus, the $\widehat{\Pi}$ scheme satisfies the definition of a \widehat{p} -correct (α, β) -incompressible encoding scheme.

Let us denote C the set of all possible code words. From the definition of $\widehat{\text{Enc}}$ follows that

$$\forall c \in C, \forall m \in M : \Pr[\widehat{\text{Enc}}(m) = c] \geq \frac{q(\lambda)}{|C|}.$$

That means that every message could be encoded to any possible code word with a non-zero probability, thus the proof is complete. \square

4 Impossibility of information-theoretic security

Theorem 4.1. *Let Π be a p -correct, α -bounded encoding scheme. Then*

$$\forall \beta(\lambda, k) > \alpha(\lambda, k) - k \exists \mathcal{A} : \Pr[\text{CompExp}_{\mathcal{A}, \beta}^{\Pi}(1^\lambda)] \geq p(\lambda).$$

If an α -bounded encoding scheme Π satisfies that $(1 - p(\lambda)) \in \text{negl}(\lambda)$, then from Theorem 4.1 it follows that $\beta(\lambda, k) \leq \alpha(\lambda, k) - k$. Hence, Π does not satisfy the definition of non-trivial incompressible encoding scheme. Thus, the following corollary holds.

Corollary 4.2. *It is not possible to construct a non-trivial incompressible encoding scheme information-theoretically.*

In the followings sections, we prove Theorem 4.1. For clarity, we start with some additional assumptions on the encoding scheme. First, we assume perfect correctness and deterministic decoding.

4.1 Schemes with perfect correctness and deterministic decoding

Definition 4.3. An encoding scheme $\Pi = (\text{Enc}, \text{Dec})$ has *perfect correctness*, if it is p -correct scheme with $p(\lambda) = 1 \forall \lambda \in \mathbb{N}$.

Definition 4.4. An encoding scheme $\Pi = (\text{Enc}, \text{Dec})$ has *deterministic decoding* if the algorithm Dec is deterministic, i.e., for each code word $c \in C$ there exists a unique message $m \in M$ such that $\text{Dec}(c) = m$ with probability one.

In Lemma 4.5, we show that when the decoding is deterministic there exists a message m such that the set of code words D_m is small enough that all its members can be indexed using at most $\alpha(\lambda, k) - k$ bits.

Lemma 4.5. *Let $\Pi = (\text{Enc}, \text{Dec})$ be an encoding scheme with deterministic decoding and message space $M = \{0, 1\}^k$. If $|C| \leq 2^{\alpha(\lambda, k)+1} - 1$, then $\min_{m \in M} |D_m| \leq 2^{\alpha(\lambda, k)-k+1} - 1$.*

Proof. Suppose to the contrary that $\min_{m \in M} |D_m| > 2^{\alpha(\lambda, k)-k+1} - 1$. Therefore, $\min_{m \in M} |D_m| \geq 2^{\alpha(\lambda, k)-k+1}$. Subsequently,

$$|C| = \sum_{m \in M} |D_m| \geq \sum_{m \in M} \min_{m \in M} |D_m| = 2^k \min_{m \in M} |D_m| \geq 2^k (2^{\alpha(\lambda, k)-k+1}) > 2^{\alpha(\lambda, k)+1} - 1,$$

where the first equality follows from Dec being deterministic and the second equality follows from $|M| = 2^k$. We derived that $|C| > 2^{\alpha(\lambda, k)+1} - 1$, a contradiction to the assumption about the cardinality of C . Hence, the Lemma holds. \square

Using Lemma 4.5, we construct an adversary that wins the compression experiment CompExp using the index of a codeword c in D_m as a form of compression.

Theorem 4.6. *Let $\Pi = (\text{Enc}, \text{Dec})$ be an α -bounded encoding scheme with deterministic decoding and perfect correctness. Then*

$$\forall \beta(\lambda, k) > \alpha(\lambda, k) - k \exists \mathcal{A} : \Pr[\text{CompExp}_{\mathcal{A}, \beta}^{\Pi}(1^\lambda)] = 1,$$

i.e., it is not a non-trivial (α, β) -incompressible encoding scheme.

Proof. Our construction of the adversary $\mathcal{A} = (\mathcal{A}.\text{Select}, \mathcal{A}.\text{Compress}, \mathcal{A}.\text{Expand})$ is given in Figure 4.4. The $\mathcal{A}.\text{Select}$ algorithm chooses a message m such that every member of D_m can be uniquely represented using an index (bit string) with a length less than or equal to β bits. Then the algorithm chooses an injective function f from D_m to the set of indices. The $\mathcal{A}.\text{Compress}$ gets a codeword c from the Enc algorithm and returns its index $w = f(c)$. Finally, the $\mathcal{A}.\text{Expand}$ returns $f^{-1}(w)$, which is equal to c .

All messages are k bits long, therefore $|M| = 2^k$. On the other hand, the code words are α bits long *at most*. Hence, all code word lengths from 1 to α bits are possible. The set of all possible code words is $C \subseteq \bigcup_{i=0}^{\alpha(\lambda, k)} \{0, 1\}^i$. Thus, $|C| \leq \left| \bigcup_{i=0}^{\alpha(\lambda, k)} \{0, 1\}^i \right| = 2^0 + 2^1 + \dots + 2^{\alpha(\lambda, k)} = 2^{\alpha(\lambda, k)+1} - 1$. By perfect correctness of the scheme, for all $c \in C$ there exists a unique $m \in \{0, 1\}^k$ such that $c \in D_m$. Equivalently, $C = \bigcup_{m \in M} D_m$ while for all $m_i, m_j \in M$ satisfying $(i \neq j)$ it holds that $D_i \cap D_j = \emptyset$.

Let $W = \bigcup_{i=0}^{\alpha(\lambda, k)-k} \{0, 1\}^i$ be the set of bit strings defined in Notation 2.6. Then $|W| = 2^{\alpha(\lambda, k)-k+1} - 1$. We choose $m^* = \text{argmin}_{m \in M} |D_m|$ and let $f : D_{m^*} \rightarrow W$ be an injective function satisfying that for all $c_1, c_2 \in D_{m^*}$ such that $(c_1 \neq c_2)$ it holds that $f(c_1) \neq f(c_2)$. This function is possible to construct, because we have $|D_{m^*}| \leq |W|$ from Lemma 4.5. Consequently, we construct $f^{-1} : f(D_{m^*}) \rightarrow D_{m^*}$, such that $f^{-1}(f(c)) = c$. Then, the $\text{CompExp}_{\mathcal{A}, \beta}^{\Pi}$ follows:

- $(m^*, \text{aux} = (f, f^{-1})) \leftarrow \mathcal{A}.\text{Select}(1^\lambda)$
- $c \leftarrow \text{Enc}(1^\lambda, m^*)$
- $w \leftarrow \mathcal{A}.\text{Compress}(\text{aux}, c)$
- $c' = \mathcal{A}.\text{Expand}(\text{aux}, w)$
- if $(c' = c) \wedge |w| \leq \beta(\lambda, k)$ output 1, else output 0

The $|w| \leq \beta(\lambda, k)$ holds, because $w \in W = \bigcup_{i=0}^{\alpha(\lambda, k)-k} \{0, 1\}^i$. The $c = c'$ follows from f being injective, which implies that the inverse f^{-1} is defined for all $f(c) \in W$ and $f^{-1}(f(c)) = c$. Therefore, the compression experiment always outputs one. \square

Figure 4.4: Adversary \mathcal{A}

Adversary $\mathcal{A} = (\mathcal{A}.\text{Select}(1^\lambda), \mathcal{A}.\text{Compress}(\text{aux}, c), \mathcal{A}.\text{Expand}(\text{aux}, w))$	
$\mathcal{A}.\text{Select}(1^\lambda)$: <ul style="list-style-type: none"> • Choose $m^* = \operatorname{argmin}_{m \in M} D_m$. • Choose an injective $f : D_{m^*} \rightarrow W$. • Compute $f^{-1} : f(D_{m^*}) \rightarrow D_{m^*}$ from f. • Output $(m^*, \text{aux} = (f, f^{-1}))$. 	
$\mathcal{A}.\text{Compress}(\text{aux}, c)$: <ul style="list-style-type: none"> • Parse aux as $\text{aux} = (f, f^{-1})$. • Output $w = f(c)$ 	$\mathcal{A}.\text{Expand}(\text{aux}, w)$: <ul style="list-style-type: none"> • Parse aux as $\text{aux} = (f, f^{-1})$. • Output $c' = f^{-1}(w)$

4.2 Schemes with imperfect correctness and deterministic decoding

In this section, we consider incompressible encodings schemes with imperfect correctness, i.e., where decoding succeeds only with some probability $p(\lambda) < 1$. We show that the adversary \mathcal{A} defined in Figure 4.4 wins the compression experiment with probability equal to or greater than the correctness $p(\lambda)$ of the scheme.

Lemma 4.7. *Let Π be a p -correct, α -bounded encoding scheme with deterministic decoding. Then*

$$\forall \beta(\lambda, k) > \alpha(\lambda, k) - k, \exists \mathcal{A}: \Pr[\text{CompExp}_{\mathcal{A}, \beta}^{\Pi}(1^\lambda)] \geq p(\lambda),$$

where \mathcal{A} is the adversary defined in the Figure 4.4.

Proof. Recall \mathcal{A} from Figure 4.4. $\mathcal{A}.\text{Select}$ chooses the message $m^* = \operatorname{argmin}_{m \in M} |D_m|$. The algorithm Enc outputs a code word $c_{m^*} \in C_{m^*}$. If $c_{m^*} \in D_{m^*}$, $\mathcal{A}.\text{Compress}$ will output the $w = f(c_{m^*})$ and, consequently, $\mathcal{A}.\text{Expand}$ will expand w to the original code word c_{m^*} . Therefore, if $c_{m^*} \in D_{m^*}$, then the adversary \mathcal{A} will correctly compress and expand the code word c_{m^*} . Out of the definition of p -correctness follows $\Pr[c_{m^*} \in D_{m^*}] \geq p(\lambda)$. Hence, we get $\Pr[\text{CompExp}_{\mathcal{A}, \beta}^{\Pi}(m^*) = 1] \geq p(\lambda)$. \square

4.3 Schemes with imperfect correctness and probabilistic decoding

In this section, we present the complete proof of the Theorem 4.1 without any additional assumptions.

Because of the decoding being probabilistic, we cannot use the argument $\min_{m \in M} |D_m| \leq 2^{\alpha(\lambda, k) - k + 1} - 1$ from Lemma 4.5. Instead, we show that there exists a message m^* and a set $S \subset C$, such that the S can be indexed using $\alpha(\lambda, k) - k$ bits at most and the probability of $\text{Enc}(m^*) \in S$ is greater than or equal to the correctness $p(\lambda)$ of the scheme. Using Lemma 4.9 we construct an adversary that succeeds when the message m^* is encoded into any codeword in S , therefore $\Pr[\text{CompExp}_{\mathcal{A}, \beta}^{\Pi}(m^*) = 1] \geq p(\lambda)$.

Notation 4.8. We denote the $p_{\text{ENC}}(m, c) = \Pr[\text{ENC}(m) = c]$ and the $p_{\text{DEC}}(c, m) = \Pr[\text{DEC}(c) = m]$.

Lemma 4.9. *Let $m^* = \text{argmin}_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m^*)$, then there exists a set $S \subset C$ such that $|S| = 2^{\alpha(\lambda, k) - k + 1} - 1$ and $\sum_{c \in S} p_{\text{ENC}}(m^*, c) \geq p(\lambda)$.*

The proof of Lemma 4.9 uses the two following auxiliary lemmata.

Lemma 4.10. *There exists $m^* \in M$ such that $\sum_{c \in C} p_{\text{DEC}}(c, m^*) \leq 2^{\alpha(\lambda, k) - k + 1} - 1$.*

Proof. The $|C| = 2^{\alpha(\lambda, k) + 1} - 1$ and the $|M| = 2^k$. We suppose for the contrary, that

$$\forall m \in M : \sum_{c \in C} p_{\text{DEC}}(c, m^*) > 2^{\alpha(\lambda, k) - k + 1} - 1,$$

that implies that

$$\min_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m^*) > 2^{\alpha(\lambda, k) - k + 1} - 1.$$

Thence $\min_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m^*) \geq 2^{\alpha(\lambda, k) - k + 1}$.

$$\begin{aligned} |C| &= \sum_{c \in C} 1 = \sum_{c \in C} \left(\sum_{m \in M} p_{\text{DEC}}(c, m) \right) = \sum_{m \in M} \left(\sum_{c \in C} p_{\text{DEC}}(c, m) \right) \\ &\geq \sum_{m \in M} \left(\min_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m^*) \right) = |M| \min_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m^*) \\ &\geq 2^k 2^{\alpha(\lambda, k) - k + 1} = 2^{\alpha(\lambda, k) + 1} \\ &> |C|. \end{aligned}$$

We have a contradiction, thence the lemma holds. \square

Lemma 4.11. *Let $n \in \mathbb{N}$, $K \in [0, n]$, $\mathbf{y} = (y_1, \dots, y_n)$, such that for all $i, j \in \{0, \dots, n\}$ $y_i \geq 0$ and $(i > j) \Rightarrow (y_i < y_j)$. $\mathcal{X} = \{\mathbf{x} = (x_1, \dots, x_n) \mid \forall k \in \{0, \dots, n\} x_k \in [0, 1], \sum_{k=1}^n x_k = K\}$.*

$$\text{Let us define } \mathbf{z} = \begin{cases} 1 & \text{if } i \leq \lfloor K \rfloor, \\ K - \lfloor K \rfloor & \text{if } \lfloor K \rfloor < i \leq \lfloor K \rfloor + 1, \\ 0 & \text{if } i > \lfloor K \rfloor + 1. \end{cases}$$

Then

$$\mathbf{z} = \text{argmax}_{\mathbf{x} \in \mathcal{X}} \sum_{k=1}^n x_k y_k.$$

Proof. First, we verify that $\mathbf{z} \in \mathcal{X}$.

- $z_k \in [0, 1]$ for all $k \in \{0, \dots, n\}$, because $1 \in [0, 1], 0 \in [0, 1]$ and $K - \lfloor K \rfloor \in [0, 1]$.
- $\sum_{k=1}^n z_k = \underbrace{1 + \dots + 1}_{\lfloor K \rfloor \text{-times}} + (K - \lfloor K \rfloor) = \lfloor K \rfloor + (K - \lfloor K \rfloor) = K$.

Let us suppose to the contrary that there exists $\bar{\mathbf{z}} \in \mathcal{X}$, such that

$$\bar{\mathbf{z}} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \sum_{k=1}^n x_k y_k, \quad \sum_{k=1}^n \bar{z}_k y_k > \sum_{k=1}^n z_k y_k.$$

We denote $i = \operatorname{argmin}_{k \in \{0, \dots, n\}} z_i \neq \bar{z}_i$. We distinguish the following cases:

- If $i \leq \lfloor K \rfloor$. Then $z_i = 1, \bar{z}_i \in [0, 1]$ and $\bar{z}_i \neq z_i$. That implies $\bar{z}_i < z_i$.
- If $\lfloor K \rfloor < i \leq \lfloor K \rfloor + 1$. Then $z_i = K - \lfloor K \rfloor$. From the definition of i we get $\bar{z}_k = z_k \forall k \in \mathbb{N}, k < i$. Thus $\sum_{k=1}^{i-1} \bar{z}_k = \sum_{k=1}^{i-1} z_k = \lfloor K \rfloor$ and we know that $\sum_{k=1}^n \bar{z}_k = K$. That implies that $\bar{z}_i \leq K - \lfloor K \rfloor = z_i$. Then $\bar{z}_i \neq z_i$ implies $\bar{z}_i < z_i$.
- If $i > \lfloor K \rfloor + 1$, then from the definition of i we obtain that for all $k \in \mathbb{N}, k < i$: $\bar{z}_k = z_k$. That implies $\sum_{k=1}^{i-1} \bar{z}_k = \sum_{k=1}^{i-1} z_k = K$. However, $\bar{\mathbf{z}} \in \mathcal{X}$. Thus for all $k \geq i$: $\bar{z}_k = 0 = z_k$, which contradicts the definition of i . Hence $i \leq \lfloor K \rfloor + 1$.

We notice that $\sum_{k=1}^i \bar{z}_k < \sum_{k=1}^i z_k \leq K$ and $\bar{\mathbf{z}} \in \mathcal{X} \Rightarrow \sum_{k=1}^n \bar{z}_k = K$. Therefore, there exists an index $l > i$, such that $\bar{z}_l > 0$. We define $j = \operatorname{argmax}_{k \in \{0, \dots, n\}} \bar{z}_k \neq 0$. We choose arbitrary $\varepsilon \in (0, \min\{\bar{z}_j, 1 - \bar{z}_i\})$. We define $\hat{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_{i-1}, \bar{z}_i + \varepsilon, \bar{z}_{i+1}, \dots, \bar{z}_{j-1}, \bar{z}_j - \varepsilon, \bar{z}_{j+1}, \dots, \bar{z}_n)$. Then

$$\sum_{k=1}^n \hat{z}_k y_k = \sum_{k=1}^n \bar{z}_k y_k + \underbrace{\varepsilon(y_i - y_j)}_{\geq 0} \geq \sum_{k=1}^n \bar{z}_k y_k.$$

If $y_i > y_j$, then $\sum_{k=1}^n \hat{z}_k y_k > \sum_{k=1}^n \bar{z}_k y_k$. Thus $\bar{\mathbf{z}} \neq \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \sum_{k=1}^n x_k y_k$. On the other hand, if $y_i = y_j$, then from the definition of \mathbf{y} we obtain that

$$\forall k \in \{i, \dots, j\}, y_i = y_k = y_j. \quad (1)$$

We show that $j \geq \lfloor K \rfloor + 1$. Suppose to the contrary that $j \leq \lfloor K \rfloor$. $K = \sum_{k=1}^j \bar{z}_k \leq \sum_{k=1}^j 1 = j \leq \lfloor K \rfloor$. If $K \notin \mathbb{N}$, then $\lfloor K \rfloor < K$, thus we have a contradiction. If $K \in \mathbb{N}$, then $K = \lfloor K \rfloor = j$ and for all $k \in \{1, \dots, j\}$ $\bar{z}_k = 1$. Nevertheless,

$$z_k = \begin{cases} 1 & \text{if } k \in \{0, \dots, \lfloor K \rfloor = j\}, \\ 0 & \text{if } k \in \{j+1, \dots, n\}. \end{cases}$$

Thence $\bar{\mathbf{z}} = \mathbf{z}$, which contradicts the definition of $\bar{\mathbf{z}}$. Now we proof that if the eq. (1) holds, than $\sum_{k=1}^n z_k y_k = \sum_{k=1}^n \bar{z}_k y_k$.

$$\begin{aligned}
\sum_{k=1}^n z_k y_k &= \sum_{k=1}^{i-1} z_k y_k + \sum_{k=i}^j z_k y_k + \sum_{k=j+1}^{\lfloor K \rfloor + 1} z_k y_k + \sum_{k=\lfloor K \rfloor + 2}^n z_k y_k \\
&= \sum_{k=1}^{i-1} z_k y_k + y_i \sum_{k=i}^j z_k \\
&= \sum_{k=1}^{i-1} \bar{z}_k y_k + y_i \sum_{k=i}^j \bar{z}_k \\
&= \sum_{k=1}^n \bar{z}_k y_k
\end{aligned}$$

Where the second equation follows from eq. (1), the fact that $j \geq \lfloor K \rfloor + 1$ and the definition of \mathbf{z} . The third equation holds because $K = \sum_{k=1}^{i-1} z_k + \sum_{k=i}^n z_k = \sum_{k=1}^{i-1} \bar{z}_k + \sum_{k=i}^n z_k$, thus $\sum_{k=i}^n z_k = \sum_{k=i}^n \bar{z}_k$. Therefore, we have a contradiction to the supposal that $\sum_{k=1}^n \bar{z}_k y_k > \sum_{k=1}^n z_k y_k$. Hence, the lemma holds. \square

Next, we proceed with the proof of Lemma 4.9.

Proof of Lemma 4.9. Suppose to the contrary that $\forall S \subset C$ satisfying $|S| = 2^{\alpha(\lambda, k) - k + 1} - 1$ it holds that $\sum_{c \in S} p_{\text{ENC}}(m^*, c) < p(\lambda)$. We denote $K = \sum_{c \in C} p_{\text{DEC}}(c, m^*) \leq 2^{\alpha(\lambda, k) - k + 1} - 1$, where the last inequality comes from Lemma 4.10. That implies $K \leq |S|$.

The p-correctness can be expressed in the following ways:

$$\begin{aligned}
&\forall \lambda \in \mathbb{N}, \forall m \in M, \left(\Pr[\text{Dec}(\text{Enc}(1^\lambda, m)) = m] \right) \geq p(\lambda), \\
&\forall \lambda \in \mathbb{N}, \forall m \in M, \left(\sum_{c \in C} p_{\text{ENC}}(m, c) p_{\text{DEC}}(c, m) \right) \geq p(\lambda).
\end{aligned}$$

Next, we sort the C in a way that $\forall i, j \in \{1, \dots, |C|\}, (i \leq j) \Rightarrow p_{\text{ENC}}(m, c_i) \leq p_{\text{ENC}}(m, c_j)$. Let $S = \{c_1, \dots, c_{|S|}\}$, where $|S| = 2^{\alpha(\lambda, k) - k + 1} - 1$. We assume that the distribution of $p_{\text{DEC}}(m^*, c_i)$ is as follows

$$p_{\text{DEC}}(m^*, c_i) = \begin{cases} 1 & \text{if } i \leq \lfloor K \rfloor, \\ K - \lfloor K \rfloor & \text{if } \lfloor K \rfloor < i \leq \lfloor K \rfloor + 1, \\ 0 & \text{if } i > \lfloor K \rfloor + 1. \end{cases}$$

The proof that the sum $\sum_{c \in C} p_{\text{ENC}}(m^*, c) p_{\text{DEC}}(c, m^*)$ is maximized for the p_{DEC} distribution defined above follows from Lemma 4.11, where $n = |C|$, $\mathbf{y} = p_{\text{ENC}}(m^*, c)$,

and $\mathbf{z} = p_{\text{DEC}}(c, m^*)$. Then

$$\begin{aligned}
\sum_{c \in C} p_{\text{ENC}}(m^*, c) p_{\text{DEC}}(c, m^*) &= \sum_{i=1}^{|C|} p_{\text{ENC}}(m^*, c_i) p_{\text{DEC}}(c_i, m^*) \\
&= \sum_{i=1}^{|S|} p_{\text{ENC}}(m^*, c_i) p_{\text{DEC}}(c_i, m^*) \\
&\leq \sum_{c \in S} p_{\text{ENC}}(m^*, c) \\
&< p(\lambda),
\end{aligned}$$

where the last inequality comes from our supposal. This contradicts the p -correctness definition, thus the proof is complete. \square

We can now proceed with the proof of Theorem 4.1.

Proof of Theorem 4.1. Our construction of the adversary $\mathcal{A} = (\mathcal{A}.\text{Select}, \mathcal{A}.\text{Compress}, \mathcal{A}.\text{Expand})$ is given in Figure 4.5. The $\mathcal{A}.\text{Select}$ algorithm chooses a message m^* such that there exists a set $S \subset C$ such that every member of S can be uniquely represented using an index (bit string) with a length less than or equal to β bits and $\Pr[c \in S] = \sum_{c \in S} p_{\text{ENC}}(m^*, c) \geq p(\lambda)$. Then the algorithm chooses a bijection g from S to the set of indices $W = \bigcup_{i=0}^{\alpha(\lambda, k)} \{0, 1\}^i$. The $\mathcal{A}.\text{Compress}$ gets a code word c from the Enc algorithm and returns its index $w = f(c)$, if $c \in S$. If $c \notin S$, it returns an arbitrary element of W . Finally, the $\mathcal{A}.\text{Expand}$ returns $f^{-1}(w)$, which is equal to c if the c is in S .

The set S can be obtained by sorting the code words with respect to $p_{\text{ENC}}(m^*, c)$ and taking the first $2^{\alpha(\lambda, k) - k + 1} - 1$ code words with the highest value ($p_{\text{ENC}}(m^*, c)$). The bijection g is possible to construct, because $|S| = 2^{\alpha(\lambda, k) - k + 1} - 1 = |W|$.

Then, the $\text{CompExp}_{\mathcal{A}, \beta}^{\text{II}}$ follows:

- $(m^*, \text{aux} = (g, g^{-1})) \leftarrow \mathcal{A}.\text{Select}(1^\lambda)$
- $c \leftarrow \text{Enc}(1^\lambda, m^*)$
- $w \leftarrow \mathcal{A}.\text{Compress}(\text{aux}, c)$
- $c' = \mathcal{A}.\text{Expand}(\text{aux}, w)$
- if $(c' = c) \wedge |w| \leq \beta(\lambda, k)$ output 1, else output 0

The bound $|w| \leq \beta(\lambda, k)$ holds, because $w \in W$, $W = \bigcup_{i=0}^{\alpha(\lambda, k) - k} \{0, 1\}^i$. The equality $c = c'$ holds if and only if $c \in S$. We know that $\Pr[c \in S] = \sum_{c \in S} p_{\text{ENC}}(m^*, c) \geq p(\lambda)$ from the Lemma 4.9. Therefore the proof is complete. \square

Figure 4.5: Adversary \mathcal{A}

Adversary $\mathcal{A} = (\mathcal{A}.\text{Select}(1^\lambda), \mathcal{A}.\text{Compress}(\text{aux}, c), \mathcal{A}.\text{Expand}(\text{aux}, w))$	
$\mathcal{A}.\text{Select}(1^\lambda)$: <ul style="list-style-type: none"> • Choose $m^* = \operatorname{argmin}_{m \in M} \sum_{c \in C} p_{\text{DEC}}(c, m^*)$. • Choose a bijection $g : S \rightarrow W$. • Compute $g^{-1} : W \rightarrow S$ from g. • Output $(m^*, \text{aux} = (g, g^{-1}))$. 	
$\mathcal{A}.\text{Compress}(\text{aux}, c)$: <ul style="list-style-type: none"> • Parse aux as $\text{aux} = (g, g^{-1})$. • Output $w = g(c)$ 	$\mathcal{A}.\text{Expand}(\text{aux}, w)$: <ul style="list-style-type: none"> • Parse aux as $\text{aux} = (g, g^{-1})$. • Output $c' = g^{-1}(w)$

5 Conclusions

We proved that there cannot exist any non-trivial incompressible encoding schemes in the plain model secure against computationally unbounded adversaries. Our work leaves some open questions:

- In the current version of the compression experiment the adversary \mathcal{A} is allowed to choose the message. Our impossibility results depend on it. Will anything change if the message in the compression experiment is chosen randomly?
- Does the impossibility hold also in the *random oracle model*? (Random oracle model is a model, where all algorithms have access to a truly random function RO, also known as random oracle.)

References

- [DGO19] Ivan Damgård, Chaya Ganesh, and Claudio Orlandi. Proofs of replicated storage without timing assumptions. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 355–380. Springer, 2019.
- [Dzi06] Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 207–224. Springer, 2006.
- [Lab17] Protocol Labs. Filecoin: A decentralized storage network. 2017. Available at <https://filecoin.io/filecoin.pdf>.
- [MW20] Tal Moran and Daniel Wichs. Incompressible encodings. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 494–523. Springer, 2020.
- [Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Syst. Tech. J.*, 28(4):656–715, 1949.