ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Revisiting and Enhancing SCB mode towards Length-Preserving AEAD

Semester Project Report

Ganyuan Cao

7th June, 2023

Advisors: Prof. Dr. Ueli Maurer, Dr. Fabio Banfi
Information Security and Cryptography Research Group
Institute of Theoretical Computer Science
Department of Computer Science, ETH Zürich

Acknowledgement

I would like to thank Prof. Ueli Maurer and Dr. Fabio Banfi for their help and advising throughout this project.

Abstract

Secure Code Book (SCB) mode of operation, proposed by Banfi, is the first length-preserving encryption (LPE) scheme that achieves semantic (IND-CPA) security. SCB mode can be considered as a variant of ECB mode but resolving the problem of block repetition in ECB by constructing repetition signals that include repetition counter and the hash of the corresponding message block. In contrast to prior constructions, SCB mode trades for stronger security at the cost of imperfect correctness, thereby prompting the study on the trade-off between security and correctness in a symmetric encryption scheme.

This work re-evaluates the security and correctness of the SCB mode. We present a proof of security under a relaxed assumption that each block is not repeated for more than 2^{σ} times (where σ denotes the security parameter), showing that semantic security is still achievable with smaller security parameter. We also prove the correctness of the scheme taking counter values into consideration during the decryption, which leads to a better correctness. To enhance the correctness of SCB mode in case of ciphertext reordering, we introduce the sliding windows technique, enabling dynamic updates of the valid counter range. Furthermore, we follow the Encode-then-Encipher paradigm to extend the SCB mode to a length-preserving AEAD scheme under the assumption that there is at least one repeated block. Our construction serves a preliminary attempt towards length-preserving AEAD.

Contents

Chapter 1

Introduction

Secure Code Book (SCB) mode of operation, introduced in [Ban22], is the first length-preserving encryption scheme that achieves semantic (IND-CPA) security. SCB mode shows that semantic security is indeed possible for length-preserving encryption at cost of imperfect correctness. SCB mode can be viewed as a variant of Electronic Code Book (ECB) mode but handles the block repetition in a different way. In SCB mode, repetition signals are generated from the message blocks, incorporating a repetition counter and the hash of the corresponding message block. Rather than re-encrypting the same message block, SCB mode encrypts the repetition signals to avoid block repetition in ciphertexts.

In this work, we revisit the security and correctness of SCB mode. We re-analyze the security of SCB mode by taking practical parameters into consideration, and prove an improved security bound. Additionally, we extend the SCB mode of operation with counter handling mechanism during the decryption process, and prove a better correctness in both cases where blocks are and are not reordered. Furthermore, we propose an authenticated encryption with associated data (AEAD) based on the current SCB mode of operation utilizing a tweakable block cipher, achieving length-preservation in terms of blocks needed.

1.1 Background and Motivation

Semantic Security was initially defined for *public-key encryption* and achieved by Goldwasser and Micali in [GM82]. Bellare et al. [BDJR97] extended this notion to symmetric encryption. Rogaway [Rog04] subsequently made it possible for symmetric encryption scheme to achieve semantic security with the use of *nonce*. Current modes of operation typically require *ciphertext expansion* of at least one block to accommodate the nonce used in the encryption process. An exception to this is the Electronic Codebook (ECB) mode of

operation, which is *length-preserving*, meaning that the length of plaintext is the same as that of ciphertext. However, it is trivial to see that an adversary can easily distinguish between two ciphertexts encrypted using ECB mode (or from a random bit string) by checking for repeated blocks in the ciphertext, demonstrating that ECB mode does not possess *indistinguishability under chosen plaintext attack* (IND-CPA), a formalized notion for semantic security.

Although many symmetric encryption schemes employ expansion to enhance security, the property of length preservation is also a crucial aspect of such schemes. In certain situations, accommodating expansion may not be feasible. For instance, when encrypting a hard drive or other storage device, the encrypted data must be saved to the disk in a way that is compatible with the disk's formatting. If the encrypted data is of a different length than the original data, it may not fit in the same location on the disk, thereby resulting in compatibility problems. Furthermore, ciphertext expansion leads to increased communication costs between the sender and receiver. The larger ciphertext necessitates more bandwidth and network resources than the original message, which can raise the communication cost of the network in terms of both time and resources. If two parties are exchanging numerous messages over the network, the cost of ciphertext expansion can accumulate rapidly.

1.2 Related Work

Many attmepts have been made to achieve *length-preserving encryption* (LPE). It is nevertheless acknowledged that achieving length preservation is feasible if one instead settles for a weaker notion of *pseudorandom permutation* (PRP) security. In [BR99], Bellare and Rogaway initiated the study of *variable-input-length* (VIL) ciphers and raised the question of how to transform a block cipher designed for fixed-length inputs into a VIL cipher while preserving the length. In their paper, Bellare and Rogaway provided a concrete example that can be roughly described as a two-pass CBC-MAC over arbitrary input length. Subsequently, Bleichenbacher and Desai [BD99] refined this scheme to achieve Strong PRP (SPRP) security. Another notable construction presented by Cook et al in [CYK04] introduces *elastic block ciphers* (EBC), which achieve LPE by treating only the round function of the underlying block cipher as a black box, rather than the entire block cipher as in previous work.

Furthermore, there have been attempts to construct *tweakable* LPE from tweakable block ciphers [LRW11]. In [MF07], McGrew and Fluhrer proposed the *Extended Codebook* (XCB) mode of operation, which is the first VIL tweakable LPE scheme by combining a block cipher with a universal hash function to realize an unbalanced three-round Feistel network. Other constructions include EME* [Hal04], an extension of the EME scheme [HR04] that achieves

LPE but not VIL. More recent constructions include TCT_1 and TCT_2 [ST13] by Shrimpton and Terashima, instantiated from their Protected-IV (PIV) construction.

Attempts have also been made to develop authenticated encryption schemes that preserve the length. SIV mode [RB18] eliminates the need for including IV in the ciphertexts by generating a synthetic initialization vector using Galois field multiplication based on a nonce. However, SIV mode still needs an extra block to accommodate the tag for authentication. Another mode of operation known as OCB [RBBK01], along with its tweakable version TAE [LRW11], achieves length preservation in terms of the number of blocks required, under the assumption that the message length is not a multiple of n. An τ-bit tag can be then appended to the last ciphertext block to provide authenticity. Note that these schemes still necessitate the use of a nonce, which still requires expansion unless it is pre-established by both communicating parties.

Above constructions either fail to achieve length-preservation or semantic security. Instead of settling for weaker security notion, recent research has demonstrated that it is possible for LPE to achieve semantic security at the expense of imperfect correctness, while still making it usable in practice. In [Ban22], Banfi proposed *Secure Codebook* (SCB) mode of operation, which is the first length-preserving encryption scheme that effectively achieves the semantic (IND-CPA) security.

SCB mode can be viewed as a variation of the Electronic Codebook (ECB) mode, but it manages block repetition in a distinct manner. The SCB mode uses two types of blocks: actual message blocks and repetition signals. The SCB mode maintains a mapping between the hash of an actual message block and the number of times that the block has been repeated in its encryption state. A repetition signal is then constructed to include the hash of its corresponding message block, the number of repetitions up to that point, and leading zeros for identification. Instead of re-encrypting the actual message block, SCB mode encrypts the repetition signal. SCB can be further extended to handle *variable-length input* by applying the *ciphertext stealing* (CTS) paradigm. In the case of block reordering, SCB also introduces associated algorithms to tag the ambiguous block that has the structure of a repetition block and recover from the actual message block.

Clearly, SCB trades for stronger security at the cost of correctness since collision may happen when hashing the message block, and some actual message blocks may have the structure of a repetition signal, which results in the incorrect decryption. Security and correctness parameters were introduced to analyze the trade-off between security and correctness.

1.3 Our Contribution

In this work, we revisit SCB mode of operation and propose several modifications. Firstly, we reassess the trade-off between the security and correctness of SCB mode. Specifically, the original SCB design lacks verification of the repetition counter during decryption, leading to potential incorrect decryption of blocks. This introduces a trade-off where improving security compromises correctness. As a first step, we expand current SCB scheme to incorporate the counter verification. Moreover, the original paper on SCB mode makes assumption on a total number of blocks queried, which determines the bound on the security parameter. However, we modify this assumption by considering the number of times that each distinct block is repeated. This relaxation allows us to minimize the lower bound on the security parameter. We conduct a re-analysis of the security of SCB mode based on our revised assumption. Our findings indicate that an adversary does not gain a significant advantage under this relaxed assumption, thereby demonstrating that SCB still maintains IND-CPA security with a smaller security parameter. Finally, we provide a proof for correctness of the extended SCB scheme that handles counters, resulting in an improved correctness bound.

In addition to revisiting the security and correctness trade-off, we propose a "sliding window technique" for counter validation in decryption in the case when blocks are reordered. Our technique utilizes dynamically updated windows to maintain a range of valid counter values. This counter validation scheme has enabled us to reduce the advantage of a correctness adversary, which implies better correctness.

Finally, we present an extension of the SCB mode to an authenticated encryption scheme with associated data (AEAD). This is achieved under assumption that there is at least one repetition of a block in the plaintext. Our approach involves instantiating the SCB mode following the Encode-then-Encipher paradigm by Bellare and Rogway [BR00] with a tweakable block cipher. Compared with OCB mode [RBBK01] and TAE [LRW11], that trade for length-preservation at cost of only providing a τ-bit tag, our scheme provides a n-bit authenticity through redundancy in message. Additionally, in our scheme, no pre-established nonce is needed. To achieve this, during encryption, when we encounter the first block repetition, instead of encrypting it, we record the hash value of its corresponding ciphertext block and its position in the plaintext. We then construct a special repetition signal with these information. During decryption, the authenticity is verified through checking the structure of the repetition signal. The hash value is used to retrieve its corresponding message block, and the position value is used to ensure that the block is correctly ordered. Indeed, our construction poses a very strong prerequisite but we conclude that it may be the first step towards constructing an length-preserving authenticated scheme.

Chapter 2

Preliminaries

2.1 Notation

We introduce the following notations that will be used throughout the paper. Let $\mathbb{N} = \{1, 2, \ldots\}$ denote the set of natural numbers. For each $n \in \mathbb{N}$, we define the set $[n] := \{1, \ldots, n\}$. Given a set S, we use the notation $S^{\geq n} := \bigcup_{i \geq n} S^i$ to denote the set of all non-empty sequences of length at least n over S, and we define $S^+ := S^{\geq 1}$. Let $x = (x_1, \cdots, x_\ell) \in S^+$ with $\ell \in \mathbb{N}$ be a sequence. We denote the length of x by $|x| := \ell$. For $y = (y_1, \ldots, y_{\ell'}) \in S'$ with $\ell' \in \mathbb{N}$, we define the concatenation of x and y as $x||y = (x_1, \ldots, x_\ell, y_1, \ldots, y_{\ell'})$. When $S = \{0,1\}$, we refer to such sequences as bit strings. Let $i \in \{0, 1, \ldots\}$, we denote the ℓ-bit string representation of i as $[i]_\ell$, where ℓ represents the number of bits. For each $n \in \mathbb{N}$, we use the notation $\mathcal{F}_n$ to denote the set of all functions from $\{0,1\}^n$ to $\{0,1\}^n$, and $\mathcal{P}_n$ to denote the set of all permutations (bijections) from $\{0,1\}^n$ to $\{0,1\}^n$.

We model a look-up table $\mathbf{T}$ that maps key bit strings of length k to value bit strings of length v as a function $\{0,1\}^k \to \{0,1\}^v \cup \{\perp\}$, where $\perp$ is a special value not belonging to $\{0,1\}^v$. To initialize $\mathbf{T}$ to an empty table, we use the notation $\mathbf{T} \leftarrow [\,]$. To assign a value V to a key K in $\mathbf{T}$, we use the notation $\mathbf{T}[K] \leftarrow V$. If a value has previously been assigned to K in $\mathbf{T}$, it will be overwritten by V. To read a value associated with a key K in $\mathbf{T}$ and assign it to V, we use the notation $V \leftarrow \mathbf{T}[K]$. If there is no value associated with K in $\mathbf{T}$, V will be assigned the special value $\perp$

Let S be a finite set. We define the notation $x \leftarrow\!\!\$\ S$ to represent the selection of a value from the set S uniformly at random, which we then assign to the variable x. For an algorithm $\mathcal{A}$, we use the notation $y \leftarrow \mathcal{A}^{\mathrm{O}_1, \mathrm{O}_2, \cdots}$ to denote running $\mathcal{A}$ given access to oracles $\mathrm{O}_1, \mathrm{O}_2, \ldots$, and then assigning of the output of $\mathcal{A}$ to y. In the event that $\mathcal{A}$ is a probabilistic algorithm, we use the notation $y \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{O}_1, \mathrm{O}_2, \cdots}$ to indicate the probabilistic procedure.

2.2 Game-based Proof

We prove the security and the correctness of our scheme using the code-based game-playing framework of Bellare and Rogaway [BR06]. This framework utilizes a game G that consists of an *Initialization* procedure (Init), a *Finalization* procedure (Finalize), and a set of oracle procedures, number of which varies depending on the specific game. An adversary $\mathcal{A}$ interacts with the oracles, which return responses to the queries made by the adversary via return statements specified in the oracles' codes.

A game G is initiated with the Init procedure, followed by the adversary's interaction with the oracle. After a number of oracle queries, the adversary halts and outputs an *adversary output*. The procedure Finalize is then executed to generate a *game output*. If a finalization procedure is not explicitly defined, we consider the *adversary output* as the *game output*. We denote $\Pr[\mathcal{A}^{\text{Init},O_1,O_2,\cdots} \Rightarrow b]$ as the probability that the adversary $\mathcal{A}$ outputs a value b after the Init procedure and queries to the oracle $O_1, O_2, \cdots$. We denote $\Pr[G(\mathcal{A}) \Rightarrow b]$ as the probability that a game G outputs b when the adversary $\mathcal{A}$ plays game G. For simplicity, we define $\Pr[G(\mathcal{A})] := \Pr[G(\mathcal{A}) \Rightarrow 0]$.

We define the advantage of the adversary $\mathcal{A}$ against the security or correctness notion N of a scheme F as $\mathbf{Adv}^{N}_{F}(\mathcal{A})$. In particular, for an *indistinguishability*, the adversary's advantage in such a game is defined as $\mathbf{Adv}^{\text{IND}}_{F}(\mathcal{A}) := \Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})]$, where $\mathcal{A}$ plays the "real world" game as G_0 and "random world" game as G_1. In other games, we define the advantage of an adversary $\mathcal{A}$ as $\mathbf{Adv}^{N}_{F}(\mathcal{A}) = \Pr[G(\mathcal{A}) \Rightarrow b]$ where b is the bit specified by different games.

2.3 Symmetric Encryption

2.3.1 LPSE

In this paper, we focus on a specific type of symmetric encryption that possesses the characteristics of being both *length-preserving* and *stateful*. We follow the definition of a *(recoverable) length-preserving stateful encryption* encryption scheme ((R)-LPSE) as in [Ban22] such a way that plaintext and ciphertext spaces are composed of bit strings whose lengths are multiples of a predetermined block length. These definitions enable us to present our results in a more simplified manner. Additionally, in [Ban22], a LPSE scheme can be extended to accommodate *variable length input* (VIL) by employing the technique of *ciphertext stealing*.

Definition 2.1 ([Ban22]) *For some $n \in \mathbb{N}$, a* Length-Preserving Stateful Encryption *(LPSE) scheme $\Pi = (\mathcal{E}, \mathcal{D})$ specifies two* stateful *algorithms*

$$\mathcal{E} : \mathcal{K} \times (\{0,1\}^n)^+ \times \mathcal{S} \to (\{0,1\}^n)^+ \times \mathcal{S}$$

and

$$\mathcal{D} : \mathcal{K} \times (\{0,1\}^n)^+ \times \mathcal{T} \to (\{0,1\}^n)^+ \times \mathcal{T}$$

where $\mathcal{K}$ is the space of keys, $\mathcal{S}$ is the space of encryption states, and $\mathcal{T}$ is the space of decryption states. The encryption algorithm $\mathcal{E}$ takes as input a key $K \in \mathcal{K}$, a message $M \in \{0,1\}^{\ell n}$, for some $\ell \in \mathbb{N}$, and an encryption state $\mathbf{S} \in \mathcal{S}$, and returns a ciphertext-state pair $(C; \mathbf{S}') \leftarrow \Pi.\mathcal{E}(K, M; \mathbf{S})$, such that $C \in \{0,1\}^{\ell n}$ (length preservation). The decryption algorithm $\mathcal{D}$ takes as input a key $K \in \mathcal{K}$, a ciphertext $C \in \{0,1\}^{\ell n}$, for some $\ell \in \mathbb{N}$, and decryption state $\mathbf{T} \in \mathcal{T}$, and returns a message-state pair $(M; \mathbf{T}') \leftarrow \Pi.\mathcal{D}(K, C; \mathbf{T})$, such that $M \in \{0,1\}^{\ell n}$.

We further assume for simplicity that the key space $\mathcal{K}$ is uniformly distributed. For better readability, we will use the following short-hand notation:

- For any key K, encryption state $\mathbf{S}$, message M, decryption state $\mathbf{T}$, and ciphertext C, we define $\mathcal{E}_K^{\mathbf{S}}(M) := \mathcal{E}(K, M; \mathbf{S})$ and $\mathcal{D}_K^{\mathbf{T}}(C) := \mathcal{D}(K, C; \mathbf{T})$.
- We write $C \leftarrow \mathcal{E}_K^{\mathbf{S}}(M)$ to mean the sequence of operations $(C; \mathbf{S}') \leftarrow \mathcal{E}_K^{\mathbf{S}}(M)$, $\mathbf{S} \leftarrow \mathbf{S}'$, and $M \leftarrow \mathcal{D}_K^{\mathbf{T}}(C)$ to mean the sequence of operations $(M; \mathbf{T}') \leftarrow \mathcal{D}_K^{\mathbf{T}}(C)$, $\mathbf{M} \leftarrow \mathbf{M}'$, $\mathbf{T} \leftarrow \mathbf{T}'$, meaning that encryption and decryption algorithms *implicitly modify the state*.

We then define a R-LPSE scheme by introducing two new algorithms that enhances correctness of an LPSE scheme in case transmitted ciphertexts are reordered.

Definition 2.2 ([Ban22]) *For some $n \in \mathbb{N}$, a* Recoverable LPSE *(R-LPSE) scheme $\Pi = (\mathcal{E}, \mathcal{D}, \widetilde{\mathcal{D}}, \mathcal{R})$ is an LPSE scheme $(\mathcal{E}, \mathcal{D})$ which additionally defines a* tagged *decryption algorithm*

$$\widetilde{\mathcal{D}} : \mathcal{K} \times (\{0,1\}^n)^+ \times \mathcal{T} \to (\{0,1\}^n)^+ \times \{0,1\}^+$$

and a recovery *algorithm*

$$\mathcal{R} : \mathcal{K} \times ((\{0,1\}^n)^+ \times \{0,1\}^+)^+ \to ((\{0,1\}^n)^+)^+$$

The tagged decryption algorithm takes as input a key $K \in \mathcal{K}$, decryption state $\mathbf{T} \in \mathcal{T}$, and a ciphertext $C \in \{0,1\}^{\ell n}$, for some $\ell \in \mathbb{N}$, and returns a tagged message $(M, \mathbf{t}) \leftarrow \Pi.\widetilde{\mathcal{D}}_K^{\mathbf{T}}(C)$, such that $M = \Pi.\mathcal{D}_K^{\mathbf{T}}(C)$ and $\mathbf{t} \in \{0,1\}^{\ell}$. The recovery algorithm takes as input a key $K \in \mathcal{K}$ and a list of tagged messages $(M_1, \mathbf{t}_1), \ldots, (M_s, \mathbf{t}_s) \in (\{0,1\}^n)^+$, for some $s \in \mathbb{N}$, and returns a list of messages $(M'_1, \ldots, M'_s) \leftarrow \Pi.\mathcal{R}_K((M_1, \mathbf{t}_1), \ldots, (M_s, \mathbf{t}_s))$, such that for any $i \in [s]$, with $M_i = M_{i,1} || \cdots || M_{i,\ell}$ and $\mathbf{t}_i = t_{i,1} || \cdots || t_{i,\ell}$, for some $\ell \in \mathbb{N}$, (1) $|M'_i| = |M_i| = \ell n$, and (2) for any $j \in [\ell]$ such that $t_{i,j} = 0$, $M'_{i,j} = M_{i,j}$.

An R-LPSE scheme operates on a ciphertext $C = C_1 || \cdots || C_\ell$, where $\ell \in \mathbb{N}$, by tagging ambiguous blocks during the decryption process and recovering them later. Specifically, during decryption, each deciphered block M_i is

tagged with a binary value t_i, where $t_i = 1$ if M_i is deemed ambiguous, and $t_i = 0$ otherwise. The output of the decryption process is a message-state pair $(M_1||\cdots||M_\ell, t_1||\cdots||t_\ell)$, where t_i is the tag value of block M_i. A block is regarded as ambiguous if it indicates a repetition, but no plaintext block can be identified in the decryption state. After a batch of ciphertexts has been transmitted, if it is not guaranteed that the communication channel does not preserve the order of blocks during transmission, running the recovery algorithm with tag value from tagged decryption algorithm resolves the block ambiguity.

2.3.2 LP-AEAD

We introduce the definition of a *length-preserving stateful authenticated encryption with associated data* (LP-AEAD) scheme. In an LP-AEAD scheme, the plaintext, ciphertext, tag, and associated data are represented as bit strings with lengths that are multiples of the block size. The term "length-preserving" implies that the number of blocks required to accommodate the plaintext is equal to the number of blocks needed for the ciphertext plus the tag, thereby length is preserved.

Definition 2.3 *For some $n \in \mathbb{N}$, a* Length-Preserving Stateful Authenticated Encryption with Associated Data *(LP-AEAD) scheme $\Pi = (\mathcal{E}, \mathcal{D})$ specifies two* stateful *algorithms*

$$\mathcal{E} : \mathcal{K} \times (\{0,1\}^n)^+ \times (\{0,1\}^n)^+ \times \mathcal{S} \to (\{0,1\}^n)^+ \times \mathcal{S}$$

and

$$\mathcal{D} : \mathcal{K} \times (\{0,1\}^n)^+ \times (\{0,1\}^n)^+ \times \mathcal{T} \to (\{0,1\}^n)^+ \cup \{\perp\} \times \mathcal{T}$$

where $\mathcal{K}$ is the space of keys, $\mathcal{S}$ is the space of encryption states, $\mathcal{T}$ is the space of decryption states. The encryption algorithm $\mathcal{E}$ takes as input a key $K \in \mathcal{K}$, a message $M \in \{0,1\}^{\ell n}$ for some $\ell \in \mathbb{N}$, associated data $V \in \{0,1\}^{\ell_v n}$ for some $\ell_v \in \mathbb{N}$, and an encryption state $\mathbf{S} \in \mathcal{S}$, and returns a triple of ciphertext, tag and state $(C, T; \mathbf{S}') \leftarrow \Pi.\mathcal{E}(K, M; \mathbf{S})$, such that $C \in \{0,1\}^{(\ell-1)n}$ and $T \in \{0,1\}^n$ (length preservation). The decryption algorithm $\mathcal{D}$ takes as input a key $K \in \mathcal{K}$, associated data $V \in \{0,1\}^{\ell_v n}$ for some $\ell_v \in \mathbb{N}$, a ciphertext with tag $C \in \{0,1\}^{\ell n}$, for some $\ell \in \mathbb{N}$, and decryption state $\mathbf{T} \in \mathcal{T}$, and returns a message-state pair $(M; \mathbf{T}') \leftarrow \Pi.\mathcal{D}(K, C; \mathbf{T})$, such that $M \in \{0,1\}^{\ell n} \cup \{\perp\}$.

2.4 Security Notions

2.4.1 Semantic (IND-CPA) Security

Our definition of IND-CPA aligns with that in [Rog04], where it is described as *indistinguishability from random bits*, denoted as IND$-CPA in the original

paper. Note that this notion capture a stronger security compared to the conventional IND-CPA security, which uses a *left-or-right* oracle, following the proof in [Ros, Claim 7.3]. We have chosen to adopt this definition in our proof as it is conceptually easier to work with while also captures the conventional IND-CPA security.

Let $n \in \mathbb{N}$, a LPSE scheme Π with key space $\mathcal{K}$, and message and ciphertext space $(\{0,1\}^n)^+$ has (q, t, ε)-*indistinguishability from random ciphertext* under chosen plaintext attack (IND-CPA) if for any adversary $\mathcal{A}$ running in time at most t and making at most q queries to the oracle Enc, we have that the advantage $\mathbf{Adv}^{\text{IND-CPA}}_{\Pi}(\mathcal{A}) \leq \varepsilon$. Consider the game $\mathrm{G}^{\text{IND-CPA-0}}_{\Pi}$ and $\mathrm{G}^{\text{IND-CPA-1}}_{\Pi}$ defined in Figure 2.1. We define the advantage of adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{\text{IND-CPA}}_{\Pi}(\mathcal{A}) := \Pr[\mathrm{G}^{\text{IND-CPA-0}}_{\Pi}(\mathcal{A})] - \Pr[\mathrm{G}^{\text{IND-CPA-1}}_{\Pi}(\mathcal{A})]$$

We let $\beta(\mathcal{A})$ denote the number of n-bit block queries made by $\mathcal{A}$.

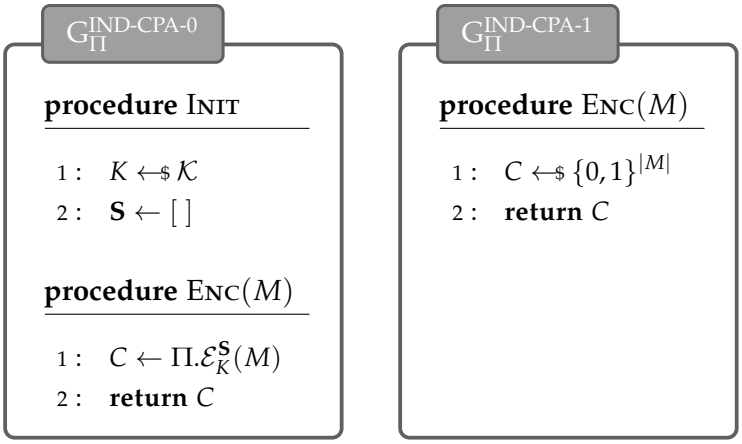

Figure 2.1: IND-CPA game for a LPSE scheme Π

2.4.2 Ciphertext Integrity (INT-CTXT)

In this work, we further extend the SCB mode to an authenticated encryption scheme, which is equivalent to IND-CPA security plus ciphertext integrity. Here we define the security notion of ciphertext integrity (INT-CTXT) following the definition in [BN00]. Let $n \in \mathbb{N}$, a LP-AEAD scheme Π with key space $\mathcal{K}$, message and ciphertext space $(\{0,1\}^n)^+$ has $(q_e, q_t, t, \varepsilon)$-*ciphertext integrity* (INT-CTXT) if for any adversary $\mathcal{A}$ running in time at most t and making at most q_e queries to the oracle Enc, at most q_t queries to the oracle Try, we have that the advantage $\mathbf{Adv}^{\text{IND-CTXT}}_{\Pi}(\mathcal{A}) \leq \varepsilon$. Consider the game $\mathrm{G}^{\text{INT-CTXT}}_{\Pi}$ defined in Figure 2.2. We define the advantage of adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{\text{INT-CTXT}}_{\Pi}(\mathcal{A}) := \Pr[\mathrm{G}^{\text{INT-CTXT}}_{\Pi}(\mathcal{A}) \Rightarrow 1]$$

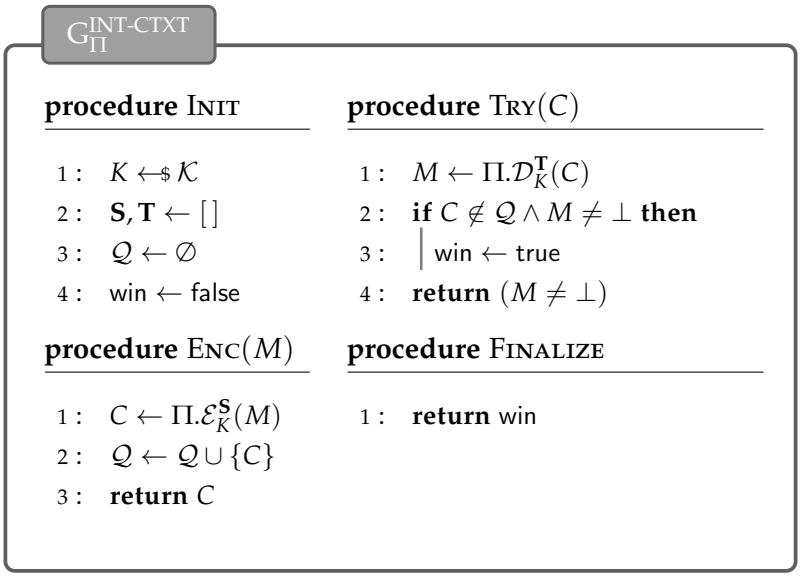

Figure 2.2: INT-CTXT game for a LPSE scheme Π

Note that we allow the adversary $\mathcal{A}$ to make multiple calls the the oracle Try. As long as one of $\mathcal{A}$'s queries makes the the oracle sets win to true, we say that $\mathcal{A}$ wins the INT-CTXT game.

2.4.3 PRP Security

Let $\kappa, n \in \mathbb{N}$, $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, for any $K \in \{0,1\}^\kappa$ and $M \in \{0,1\}^n$, define $E_K(M) := E(K, M)$. Then E is a *block cipher*, if for all $K \in \{0,1\}^\kappa$, E_K is a *permutation* on $\{0,1\}^n$. We say that E is a (q, t, ε)-*pseudorandom permutation* (PRP) if for any adversary $\mathcal{A}$ running in time at most t and making at most q queries to the oracle Enc, we have that the advantage $\mathbf{Adv}_E^{\text{PRP}}(\mathcal{A}) \leq \varepsilon$. Consider the games $G_E^{\text{PRP-0}}$ and $G_E^{\text{PRP-1}}$ defined in Figure 2.3. We define the advantage of the adversary $\mathcal{A}$ as

$$\mathbf{Adv}_E^{\text{PRP}}(\mathcal{A}) := \Pr[G^{\text{PRP-0}}(\mathcal{A})] - \Pr[G^{\text{PRP-1}}(\mathcal{A})]$$

We let $q(\mathcal{A})$ denote the number of queries made by $\mathcal{A}$.

2.4.4 Tweak PRP Security

Tweakable Block Cipher (TBC) is a cryptographic primitive initially proposed by Liskov et al. [LRW11]. In comparison to a basic block cipher, a tweakable block cipher $\widetilde{E} : \{0,1\}^\kappa \times \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$, incorporates an additional input $T \in \{0,1\}^t$, referred a "tweak." A notable advantage of the TBC is that modifying the tweak is significantly more efficient than altering the key. In [LRW11], Liskov et al. define the security of a TBC in terms of its

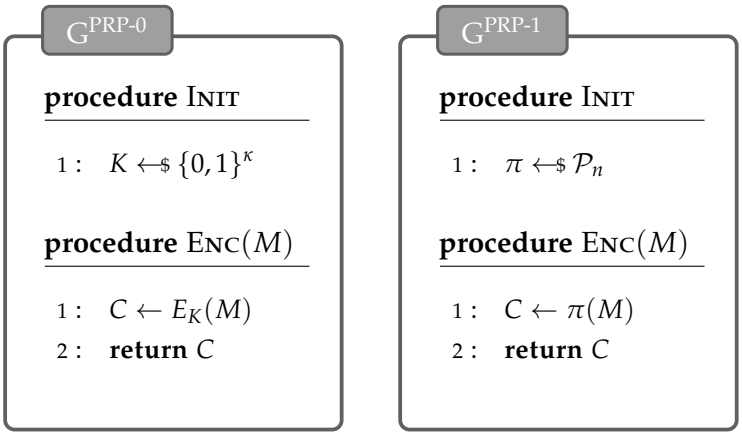

Figure 2.3: PRP game for a block cipher E

indistinguishability from a random "tweakable" permutation. Let $\widetilde{\mathcal{P}}_n$ denote the family of random permutations parameterized by the tweak T, such that $\widetilde{\pi}(T, \cdot)$ is an independently selected at random permutation on $\{0,1\}^n$ for each T. Consider the games shown in Figure 2.4, we further formalize the advantage of a TPRP adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{\text{TPRP}}_{E}(\mathcal{A}) := \Pr[\text{G}^{\text{TPRP-0}}(\mathcal{A})] - \Pr[\text{G}^{\text{TPRP-1}}(\mathcal{A})]$$

We let $q(\mathcal{A})$ denote the number of queries made by $\mathcal{A}$.

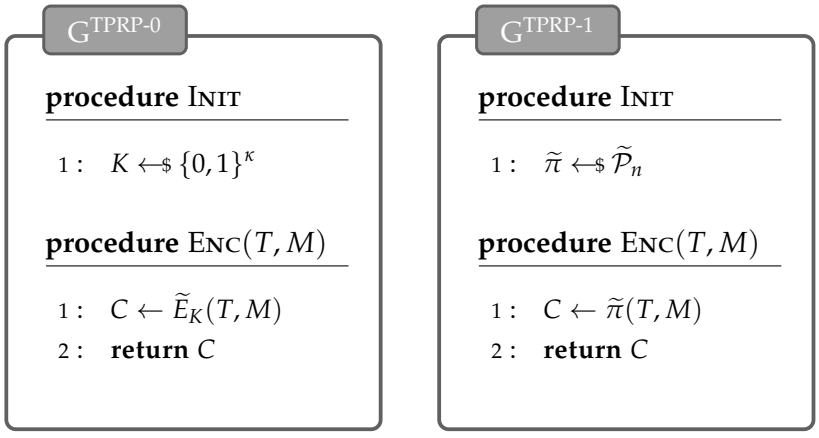

Figure 2.4: TPRP game for a tweakable block cipher $\widetilde{E}$

A stronger notion *S*TPRP can be further defined granting the adversary with access to the oracle that either answers adversary's query with $\widetilde{E}^{-1}$ or $\widetilde{\pi}^{-1}$ as illustrated in Figure 2.5. The advantage of an adversary $\mathcal{A}$ is similarly defined as:

$$\mathbf{Adv}^{\text{STPRP}}_{E}(\mathcal{A}) := \Pr[\text{G}^{\text{STPRP-0}}(\mathcal{A})] - \Pr[\text{G}^{\text{STPRP-1}}(\mathcal{A})]$$

We let $q_e(\mathcal{A})$ denote the number of queries made by $\mathcal{A}$ to Enc, and let $q_d(\mathcal{A})$ denote the number of queries made by $\mathcal{A}$ to Dec.

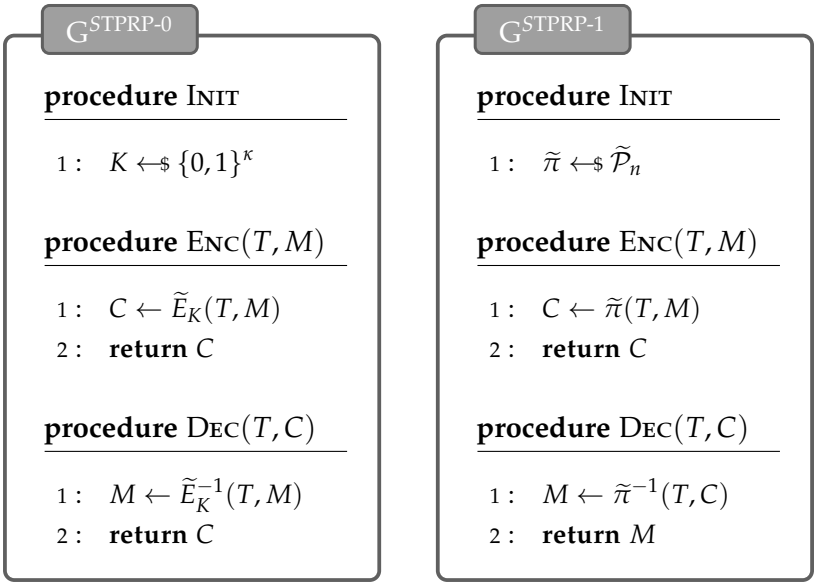

Figure 2.5: STPRP game for a tweakable block cipher $\widetilde{E}$

In [LRW11], Liskov et al. provides a constructions of tweakable block cipher that has TPRP security from a basic block cipher. Let E be a block cipher, then

$$\widetilde{E}_K(T, M) := E_K(T \oplus E_K(M))$$

is TPRP secure [LRW11, Theorem 1] such that the advantage

$$\mathbf{Adv}_{\widetilde{E}}^{\mathrm{TPRP}}(\mathcal{A}) \leq \mathbf{Adv}_{E}^{\mathrm{PRP}}(\mathcal{B}) + \frac{17q^2 - q}{2^{n+1}}$$

where $q := q(\mathcal{A})$.

A tweakable block cipher construction that has *strong* TPRP (STPRP) security with utilization of a *ε-almost 2-xor-universal hash function* is also illustrated in [LRW11]. Let $\mathcal{H}$ be the set of functions $\{0,1\}^t \to \{0,1\}^n$ such that $\Pr[h(x) \oplus h(y) = z] \leq \varepsilon$ for all x, y, z where h is chosen uniformly at random from $\mathcal{H}$. We then say $\mathcal{H}$ is *ε-almost 2-xor-universal* (ε-AXU$_2$). Let $\mathcal{H}$ be ε-AXU$_2$ with $\varepsilon \geq 2^{-n}$, the construction

$$\widetilde{E}_K(T, M) := E_K(M \oplus h(T)) \oplus h(T)$$

is strong TPRP secure. [LRW11, Theorem 2] such that the advantage

$$\mathbf{Adv}_{\widetilde{E}}^{\mathrm{STPRP}}(\mathcal{A}) \leq \mathbf{Adv}_{E}^{\mathrm{SPRP}}(\mathcal{B}) + 3\varepsilon q^2$$

where $q := q_e(\mathcal{A}) + q_d(\mathcal{A})$ and $q \leq 2^{n-1}$.

2.4.5 Collision Resistance

For $m, n \in \mathbb{N}$ with $m \gg n$, we say that $H : \{0,1\}^m \to \{0,1\}^n$ is a *collision resistant (compression) hash function* if it is hard to find $m, m' \in \{0,1\}^m$ with $m \neq m'$ such that $H(m) = H(m')$. We say an adversary $\mathcal{A}$ is a (t, ε)-*collision resistance* (CR) adversary for H if $\mathcal{A}$ runs in time at most t and has the advantage $\mathbf{Adv}_H^{\mathrm{CR}}(\mathcal{A}) \leq \varepsilon$. Consider the game in Figure 2.6. We define the advantage of adversary $\mathcal{A}$ as

$$\mathbf{Adv}_H^{\mathrm{CR}}(\mathcal{A}) = \Pr[\mathrm{G}_H^{\mathrm{CR}}(\mathcal{A}) \Rightarrow 1]$$

It should be noted that given the condition $m \gg n$ and the use of an *unkeyed* (compression) hash function H, there always exists an efficient collision resistance (CR) adversary with advantage 1. To resolve this issue, we follow the approach of [Rog06] and present the proofs by explicitly defining a CR adversary $\mathcal{D}$ through reduction, such that if there exists an efficient adversary $\mathcal{A}$ that violates the IND-CPA security of our scheme, then $\mathcal{D}$ can efficiently break the collision resistance of H with a similar efficiency as $\mathcal{A}$. Still, the proofs can be improved by adapting to *keyed* compression function with the targeted security as *weak* collision resistance (WCR)[BCK96].

$\mathrm{G}_{\Pi}^{\mathrm{CR}}$

procedure TRY(m, m')

```
1 : if H(m) = H(m') ∧ m ≠ m' then
2 :     return 1
3 : else
4 :     return 0
```

Figure 2.6: Collision Resistance game for a (compression) hash function H

2.5 Correctness

For some $n \in \mathbb{N}$, let Π be an (R-)LPSE scheme with key space $\mathcal{K}$ and message and ciphertext spaces $(\{0,1\}^n)^+$. We have two separate notions of correctness. The first notion assumes that ciphertexts are not reordered during transmission. This defines correctness for an LPSE scheme, and a R-LPSE scheme can also satisfy this notion. The second notion assumes that ciphertexts are reordered during transmission, and therefore applies only to R-LPSE schemes.

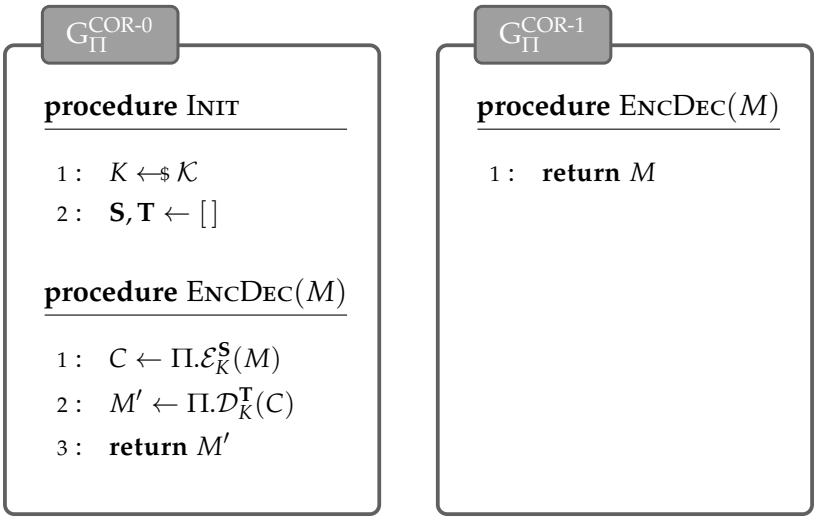

Figure 2.7: Games defining correctness *without* reordering of an LPSE scheme Π.

2.5.1 Without Reordering of Ciphertexts

The notion of *correctness without reordering* (COR) of Π is defined as the problem of distinguishing between the behaviors of two oracles. The first oracle takes a message, encrypts it, and then decrypts the resulting ciphertext to return the resulting message. The second oracle simply returns the message that was queried without any encryption or decryption. Considering games $G_{\Pi}^{\text{COR-0}}$ and $G_{\Pi}^{\text{COR-1}}$ in Figure 2.7, we define the advantage of a COR adversary $\mathcal{A}$ as

$$\mathbf{Adv}_{\Pi}^{\text{COR}}(\mathcal{A}) := \Pr[G_{\Pi}^{\text{COR-0}}(\mathcal{A})] - \Pr[G_{\Pi}^{\text{COR-1}}(\mathcal{A})]$$

We let $\beta(\mathcal{A})$ denote the number of n-bit blocks queried to EncDec by $\mathcal{A}$.

2.5.2 With Reordering of Ciphertexts

Let Π be an R-LPSE scheme. We establish two sets of notions to denote correctness. In the first scenario, we make the assumption that the reordering permutation π satisfies the condition where the repetition signal for an actual message is queried only after the corresponding actual message block has been queried. Under this assumption, we define correctness with reordering, *excluding* the tagged decryption and recovery algorithm (COR-WR*) of Π as follows: given a sequence of $s \in \mathbb{N}$ messages and a permutation on $[s]$, we define two oracles. The first oracle encrypts the messages in the original order, permutes the resulting ciphertext, decrypts them in the permuted order, and provides the sequence of permuted decrypted messages. The second oracle simply returns the permuted sequence of queried messages. We determine the advantage of a COR-WR* adversary $\mathcal{A}$ by considering the

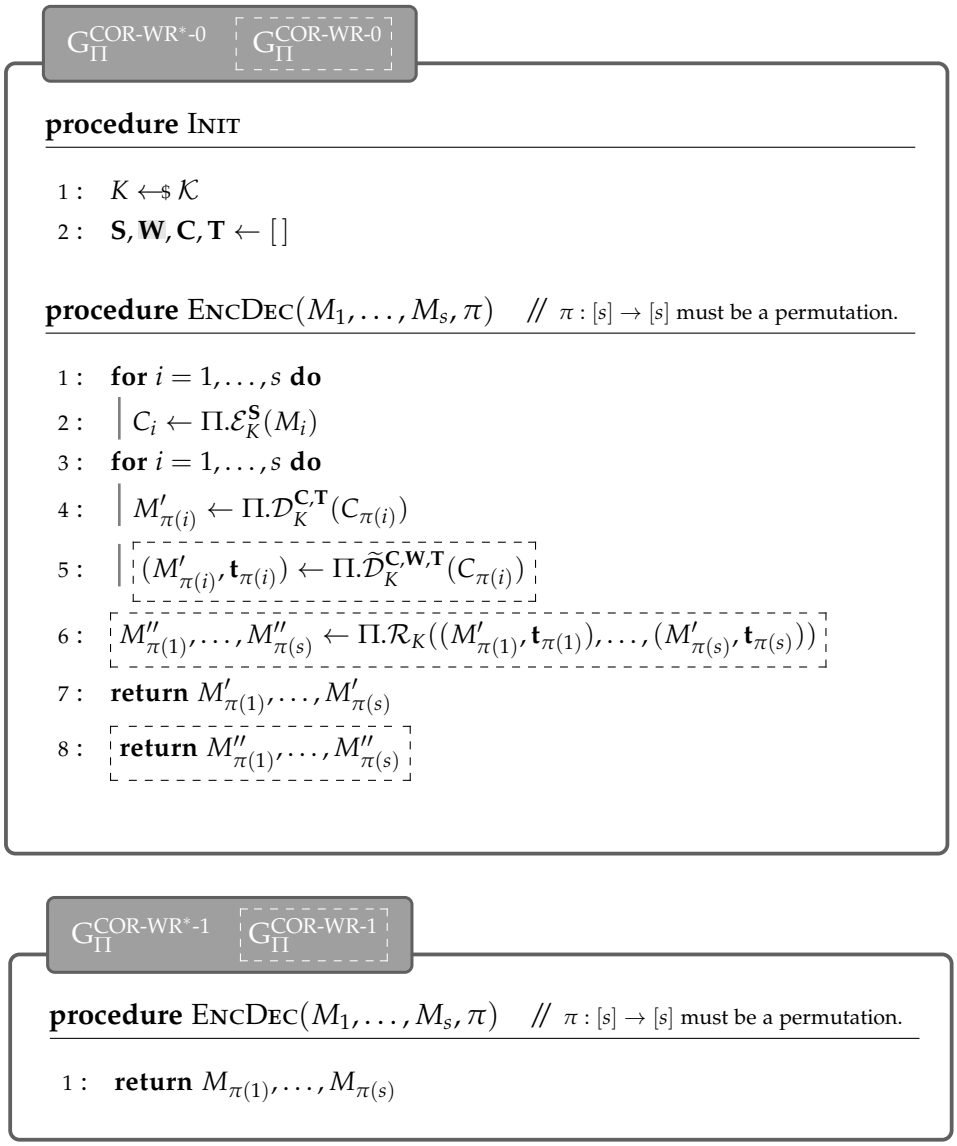

Figure 2.8: Games defining correctness *with* reordering of an R-LPSE scheme Π. Highlighted code is exclusive to $G_{\Pi}^{\text{COR-WR*-0}}$.

games $G_{\Pi}^{\text{COR-WR*-0}}$ and $G_{\Pi}^{\text{COR-WR*-1}}$ shown in Figure 2.8. The advantage of a COR-WR* adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}_{\Pi}^{\text{COR-WR*}}(\mathcal{A}) := \Pr[G_{\Pi}^{\text{COR-WR*-0}}(\mathcal{A})] - \Pr[G_{\Pi}^{\text{COR-WR*-1}}(\mathcal{A})]$$

Here, $\beta(\mathcal{A})$ denotes the total number of n-bit blocks queried to the EncDec oracle by $\mathcal{A}$.

In the second scenario, we define the correctness with reordering *including* tagged decryption and recovery algorithm (COR-WR). We modify the first

oracle by having it perform *tagged decryption* in the permuted order instead, and then applies the *recovery* algorithm on the decrypted message blocks and returns the sequence of permuted recovered messages. The adversary's advantage is defined in the same way as before:

$$\mathbf{Adv}_{\Pi}^{\text{COR-WR}}(\mathcal{A}) := \Pr[\mathrm{G}_{\Pi}^{\text{COR-WR-0}}(\mathcal{A})] - \Pr[\mathrm{G}_{\Pi}^{\text{COR-WR-1}}(\mathcal{A})]$$

2.6 Secure Code Book (SCB) Mode

Secure Code Book (SCB) mode of operation [Ban22], proposed by Banfi is the first length-preserving encryption scheme that achieve semantic IND-CPA security. SCB mode, as illustrated in Figure 2.9, can be seen as a variant of Electronic Code Book (ECB) mode. SCB handles the block repetition problem by introducing two types of blocks, which are actual message blocks and repetition signals. Specifically, SCB enciphers each newly encountered block using E_{K_1} for a block cipher E and some key K_1. Additionally, SCB maintains a lookup table $\mathbf{S}$ that associates blocks with integer counters, enabling us to track the number of times each block has been seen. Instead of re-enciphering blocks that have been seen before (as in ECB), SCB mode constructs a repetition signal that has the structure $0^{n-\sigma-\tau}||c||h$, where h is an τ-bit hash value of the corresponding actual message block, c is a σ-bit counter value represents a block's previous occurrence count (obtained from $\mathbf{S}$), and then performs an XOR operation with a second key K_2 of the same length as a block, and finally encipher the resulting bit string using E_{K_1}

For the decryption, SCB mode maintains a state by employing a lookup table $\mathbf{T}$ that maps hash values to blocks. During the decryption process, each ciphertext block is initially deciphered using $E_{K_1}^{-1}$, then checks whether the deciphered block has the structure of an actual message block or a repetition signal. If it matches with an actual message block, then the block is stored under its associated hash value in $\mathbf{T}$. If the decrypted block exhibits the structure of a repetition signal, i.e., beginning with appropriate zero-padding and its final bits corresponding to a hash value present in $\mathbf{T}$. In such cases, we simply retrieve the corresponding block from $\mathbf{T}$.

SCB mode is proven to be IND-CPA secure under the assumption $\beta := \beta(\mathcal{A}) \leq 2^{\sigma}$, where β is the total number of n-bit block queries made by an adversary $\mathcal{A}$. Specifically, the adversary's advantage is bounded by:

$$\mathbf{Adv}_{\mathsf{SCB}[E,H]}^{\text{IND-CPA}}(\mathcal{A}) \leq \mathbf{Adv}_{E}^{\text{PRP}}(\mathcal{B}) + \frac{\beta^2}{2^n}$$

It is easy to see that the correctness of SCB mode is not perfect. Firstly, it is possible that $H(M_1) = H(M_2)$ for some $M_1 \neq M_2$. Additionally, if an actual message has the structure of a repetition signal, then it will

$\mathsf{SCB}[E,H].\mathcal{E}^{\mathbf{S}}_{K_1,K_2}(M_1||\cdots||M_\ell)$

1 : **for** $i = 1,\ldots,\ell$ **do**
2 : $\quad h \leftarrow H(M_i)$
3 : $\quad$ **if** $\mathbf{S}[h] = \perp$ **then**
4 : $\quad\quad C_i \leftarrow E_{K_1}(M_i)$
5 : $\quad\quad \mathbf{S}[h] \leftarrow 0^\sigma$
6 : $\quad$ **else**
7 : $\quad\quad R \leftarrow 0^{n-\sigma-\tau}||\mathbf{S}[h]||h$
8 : $\quad\quad C_i \leftarrow E_{K_1}(K_2 \oplus R)$
9 : $\quad\quad \mathbf{S}[h] \leftarrow (\mathbf{S}[h]+1) \bmod 2^\sigma$
10 : **return** $C_1||\cdots||C_\ell$

$\mathsf{SCB}[E,H].\mathcal{D}^{\mathbf{T}}_{K_1,K_2}(C_1||\cdots||C_\ell)$

1 : **for** $i = 1,\ldots,\ell$ **do**
2 : $\quad M_i \leftarrow E^{-1}_{K_1}(C_i)$
3 : $\quad R \leftarrow K_2 \oplus M_i$
4 : $\quad h \leftarrow R \bmod 2^\tau$
5 : $\quad c \leftarrow (R \gg \tau) \bmod 2^\sigma$
6 : $\quad$ **if** $R < 2^{\sigma+\tau} \wedge \mathbf{T}[h] \neq \perp$
7 : $\quad\quad M_i \leftarrow \mathbf{T}[h]$
8 : $\quad$ **else**
9 : $\quad\quad h \leftarrow H(M_i)$
10 : $\quad\quad \mathbf{T}[h] \leftarrow M_i$
11 : **return** $M_1||\cdots||M_\ell$

Figure 2.9: Encryption and Decryption algorithms of $\mathsf{SCB}[E,H]$ with counter validation.

be misinterpreted during the decryption. The advantage of a correctness adversary $\mathcal{A}$ is bounded by:

$$\mathbf{Adv}^{\mathrm{COR}}_{\mathsf{SCB}[E,H]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{CR}}_{H}(\mathcal{B}) + \frac{2^\sigma \beta^2}{2^n}$$

Consider that with a larger σ, SCB allows more repetitions, thus provides better security. Similarly, with a larger τ, it is less likely for the collision happens, thus better correctness is provided. One can then derive the bound for the security parameter σ and the correctness parameter τ. Specifically, by the assumption from the security bound, we need to have $\beta \leq 2^\sigma$, which implies

$$\log \beta \leq \sigma \ll n - 2\log \beta$$

Also, by Birthday Bound, an implicit condition is that $\beta \leq 2^{\frac{\tau}{2}}$, and since $\sigma + \tau < n$, the correctness parameter τ is bounded by:

$$2\log \beta \ll \tau \leq n - \sigma$$

Nevertheless, the security and correctness of SCB mode do not explicitly depend on the correctness τ, but the security parameter σ, the block length n, and total number of block queried β. Clearly, one should select a larger σ for better security, while a larger σ effectively contributes to the increase of the advantage of a correctness adversary, which introduces a trade-off that increasing security compromises correctness. In this work, we propose modification to SCB mode such that a more balanced trade-off can be achieved.

SCB mode can be further extended to take *variable-length input* by using the ciphertext stealing construction. Furthermore, in case where the ciphertext blocks are reordered, SCB offers two supplementary algorithms, namely *Tagged Decryption* and *Recovery*, which rectify the misinterpretation of the repetition signals during the decryption procedure. Note that the security and correctness are inherited from the fundamental SCB construction.

Chapter 3

Revisited Security and Correctness

3.1 Decryption with Counter Validation

We propose an extension to the SCB mode of operation, which allows for the counter to be taken into account during decryption. This is achieved through the use of a lookup table $\mathbf{C}$, which is used to store the expected counter value for each message block during decryption. Specifically, for a block M that is not padded as a repetition signal, i.e., $K_2 \oplus M > 2^{\sigma+\tau}$, we assign the value of 0^σ to the corresponding entry in $\mathbf{C}[h]$, where $h = H(M)$.

During the decryption process of a repetition signal R, we first verify whether the hash value h in R corresponds to an entry in the lookup table $\mathbf{T}$. Furthermore, we also check if the counter c in R matches the expected value in $\mathbf{C}[h]$. If both checks are successful, we increment the counter by one. Since we assume no block reordering in this case, it suffices to determine whether a repetition signal is expected by comparing the counters in the repetition signal and the lookup table $\mathbf{C}$.

We define SCB mode of encryption with counter validation in decryption on the top of the definition in [Ban22]. We will then prove the IND-CPA security under a relaxed assumption in Theorem 3.2, and prove the correctness of the modified scheme in Theorem 3.3.

Definition 3.1 *Let $\kappa, n, \sigma, \tau \in \mathbb{N}$ with $\sigma + \tau < n$, $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ a block cipher, and $H : \{0,1\}^n \to \{0,1\}^\tau$ a compression function. Also let $\mathcal{S}$ be the set of $\{0,1\}^\tau \to \{0,1\}^\sigma \cup \{\perp\}$ look-up tables, $\mathcal{C}$ be the set of $\{0,1\}^\tau \to \{0,1\}^\sigma \cup \{\perp\}$ look-up tables, and $\mathcal{T}$ the set of $\{0,1\}^\tau \to \{0,1\}^n \cup \{\perp\}$ look-up tables. The* Secure Codebook *(SCB) mode of encryption is the LPSE scheme $\mathsf{SCB}[E, H] := (\mathcal{E}, \mathcal{D})$ with key space $\mathcal{K} = \{0,1\}^\kappa \times \{0,1\}^n$, encryption states space $\mathcal{S}$, decryption states spaces $\mathcal{C}$ and $\mathcal{T}$, and encryption and decryption algorithms $\mathcal{E}$ and $\mathcal{D}$ as defined in 3.1.*

$\mathsf{SCB}[E,H].\mathcal{E}^{\mathbf{S}}_{K_1,K_2}(M_1||\cdots||M_\ell)$

1 : **for** $i = 1,\ldots,\ell$ **do**
2 : $\quad h \leftarrow H(M_i)$
3 : $\quad$ **if** $\mathbf{S}[h] = \perp$ **then**
4 : $\quad\quad C_i \leftarrow E_{K_1}(M_i)$
5 : $\quad\quad \mathbf{S}[h] \leftarrow 0^\sigma$
6 : $\quad$ **else**
7 : $\quad\quad R \leftarrow 0^{n-\sigma-\tau}||\mathbf{S}[h]||h$
8 : $\quad\quad C_i \leftarrow E_{K_1}(K_2 \oplus R)$
9 : $\quad\quad \mathbf{S}[h] \leftarrow (\mathbf{S}[h]+1) \bmod 2^\sigma$
10 : **return** $C_1||\cdots||C_\ell$

$\mathsf{SCB}[E,H].\mathcal{D}^{\mathbf{C},\mathbf{T}}_{K_1,K_2}(C_1||\cdots||C_\ell)$

1 : **for** $i = 1,\ldots,\ell$ **do**
2 : $\quad M_i \leftarrow E^{-1}_{K_1}(C_i)$
3 : $\quad R \leftarrow K_2 \oplus M_i$
4 : $\quad h \leftarrow R \bmod 2^\tau$
5 : $\quad c \leftarrow (R \gg \tau) \bmod 2^\sigma$
6 : $\quad$ **if** $R < 2^{\sigma+\tau} \wedge \mathbf{T}[h] \neq \perp$
7 : $\quad\quad \wedge c = \mathbf{C}[h]$ **then**
8 : $\quad\quad M_i \leftarrow \mathbf{T}[h]$
9 : $\quad\quad \mathbf{C}[h] \leftarrow (\mathbf{C}[h]+1) \bmod 2^\sigma$
10 : $\quad$ **else**
11 : $\quad\quad h \leftarrow H(M_i)$
12 : $\quad\quad \mathbf{T}[h] \leftarrow M_i$
13 : $\quad\quad \mathbf{C}[h] \leftarrow 0^\sigma$
14 : **return** $M_1||\cdots||M_\ell$

Figure 3.1: Encryption and Decryption algorithms of $\mathsf{SCB}[E,H]$ with counter validation. Changes to original scheme are highlighted.

3.2 Relaxed Security-Correctness Tradeoff

We relax the assumption for the security by assuming that each distinct block $M \in \{0,1\}^n$ is queried for no more than $t \leq 2^\sigma$ times. Under this assumption, we claim in Theorem 3.2 that the scheme remains IND-CPA secure with negligible increase in the adversary's advantage. Under this assumption, the bound of the security parameter σ can be further reducing to

$$\log t \leq \sigma \ll n - 2\log t$$

Theorem 3.2 *For any* IND-CPA *adversary* $\mathcal{A}$ *with* $\beta := \beta(\mathcal{A})$*, assuming that each distinct block is queried for no more than* $t \leq 2^\sigma$ *times, there exist a* PRP *adversary* $\mathcal{B}$ *with* $q(\mathcal{B}) = \beta$ *and a CR adversary* $\mathcal{D}$ *with* $q(\mathcal{D}) = \beta$ *such that*

$$\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{SCB}[E,H]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{PRP}}_{E}(\mathcal{B}) + \mathbf{Adv}^{\text{CR}}_{H}(\mathcal{D}) + \frac{\beta^2}{2^n}$$

Proof The proof describes four games, denoted by G_0–G_3, that are defined in Figure 3.2, Figure 3.3 and Figure 3.4. Game G_0 is defined as $G^{\text{IND-CPA-0}}_{\mathsf{SCB}[E,H]}$, game G_1 is defined as $G^{\text{IND-CPA-0}}_{\mathsf{SCB}[\mathcal{P}_n,H]}$, game G_2 is defined as $G^{\text{IND-CPA-0}}_{\mathsf{SCB}[\mathcal{F}_n,H]}$, and game G_3 is defined as $G^{\text{IND-CPA-1}}_{\mathsf{SCB}[E,H]}$. Note that game G_2 and G_3 are identical until bad_0 or bad_1 is set to true. This means that the behavior of these two games is the

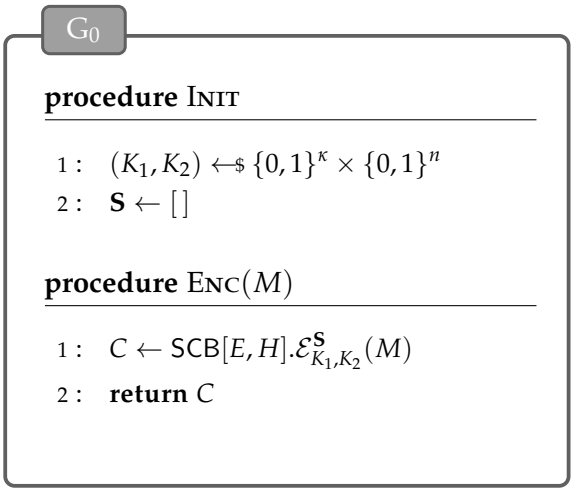

Figure 3.2: Games G_0 for the proof of Theorem 3.2.

same until bad conditions are met, which is when ρ in G_2 is queried with a certain value twice. At this point, the output of Enc is no longer uniformly random and independent. However, it is impossible to query the same M_i at Line 7 twice due to the check at Line 4. We can consider two events: let W_0 be the event that ρ is queried with the same value at Line 18 twice, and let W_1 be the event that ρ is queried with the same value once at Line 7 and once at Line 18.

In the first case, we have $\Pr[W_0] = \Pr[G_2 \text{ sets } \mathsf{bad}_0]$. Here we let t_i and t_j denote the number of queries made by the adversary on block M_i and M_j, respectively. If $H(M_i) = H(M_j)$ and $t_i + t_j > 2^\sigma$, then ρ will be queried with the same value twice at Line 18. In this case, we can create a CR adversary $\mathcal{D}$ as illustrated in Figure 3.4, which runs the adversary $\mathcal{A}$ as a subroutine. The probability of event W_1 occurring is thus upper bounded by the advantage of $\mathcal{A}$ as a collision-resistant (CR) adversary against H, denoted by $\mathbf{Adv}^{\mathrm{CR}}_H(\mathcal{D})$.

In the second case, we have that $\Pr[W_1] = \Pr[G_2 \text{ set } \mathsf{bad}_1]$. The event of collision $M_i = K_2 \oplus R$ occurs with a probability of 2^{-n}, and by multiplying this value with β, which represents the total number of queries made to ρ, an upper bound on $\Pr[W_2]$ can be derived.

By Union Bound and Fundamental Lemma of Game-Playing [BR06, Lemma 1], we have that

$$\begin{aligned}\Pr[G_2(\mathcal{A})] - \Pr[G_3(\mathcal{A})] &= \Pr[W_0] + \Pr[W_1] \\ &\leq \mathbf{Adv}^{\mathrm{CR}}_H(\mathcal{D}) + \frac{\beta}{2^n}\end{aligned}$$

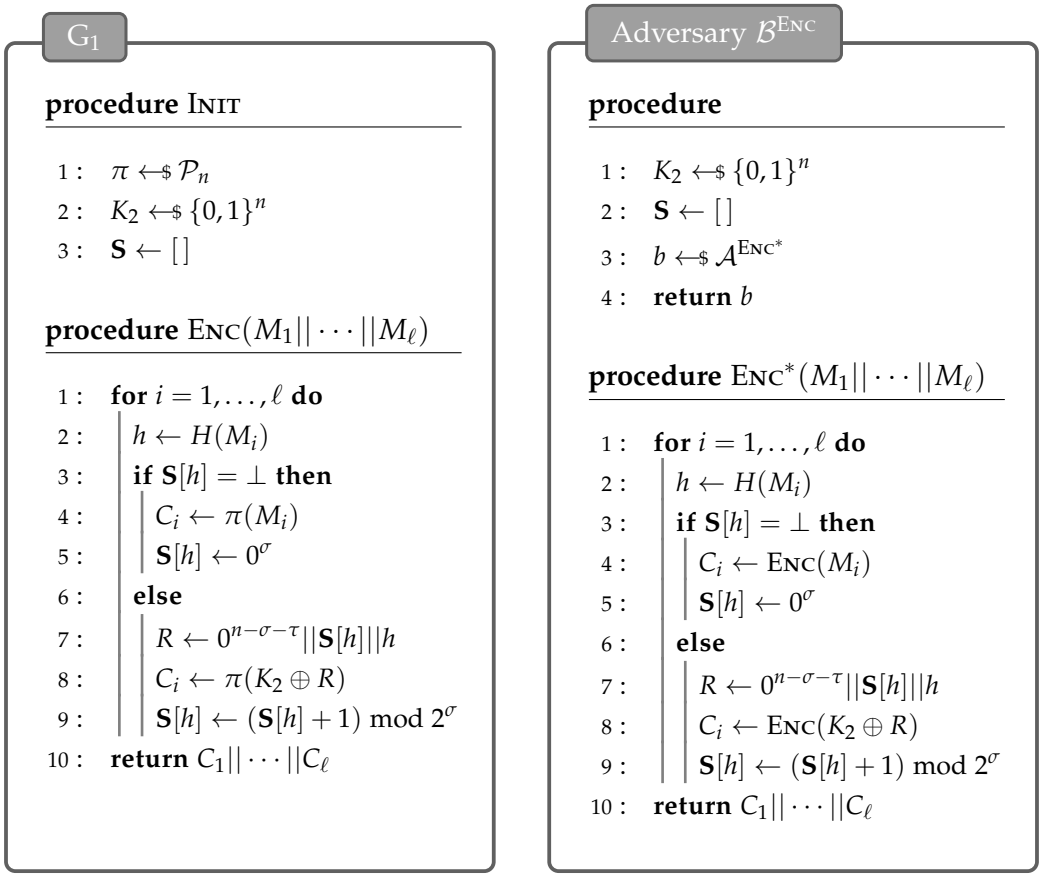

Figure 3.3: Games G_1 and the PRP adversary $\mathcal{B}$ for the proof of Theorem 3.2.

Furthermore, we have

$$\begin{aligned}\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{SCB}[E,H]}(\mathcal{A}) &= \Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_3(\mathcal{A})] \\ &= \sum_{i=0}^{2} \Pr[\mathrm{G}_i(\mathcal{A})] - \Pr[\mathrm{G}_{i+1}(\mathcal{A}]\end{aligned}$$

Let adversary $\mathcal{B}$ be as in Figure 3.3. Then:

$$\begin{aligned}\Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_1(\mathcal{A})] &= \Pr[\mathrm{G}_E^{\text{PRP-0}}(\mathcal{B})] - \Pr[\mathrm{G}_E^{\text{PRP-1}}(\mathcal{B})] \\ &= \mathbf{Adv}^{\text{PRP}}_E(\mathcal{B})\end{aligned}$$

By PRP/PRF Switching Lemma[BR06, Lemma 1], we have

$$\begin{aligned}\Pr[\mathrm{G}_1(\mathcal{A})] - \Pr[\mathrm{G}_2(\mathcal{A})] &= \Pr[\mathcal{B}^{\pi} \mid \pi \leftarrow\!\$\ \mathcal{P}_n] - \Pr[\mathcal{B}^{\rho} \mid \rho \leftarrow\!\$\ \mathcal{F}_n] \\ &= \frac{\beta(\beta-1)}{2^{n+1}}\end{aligned}$$

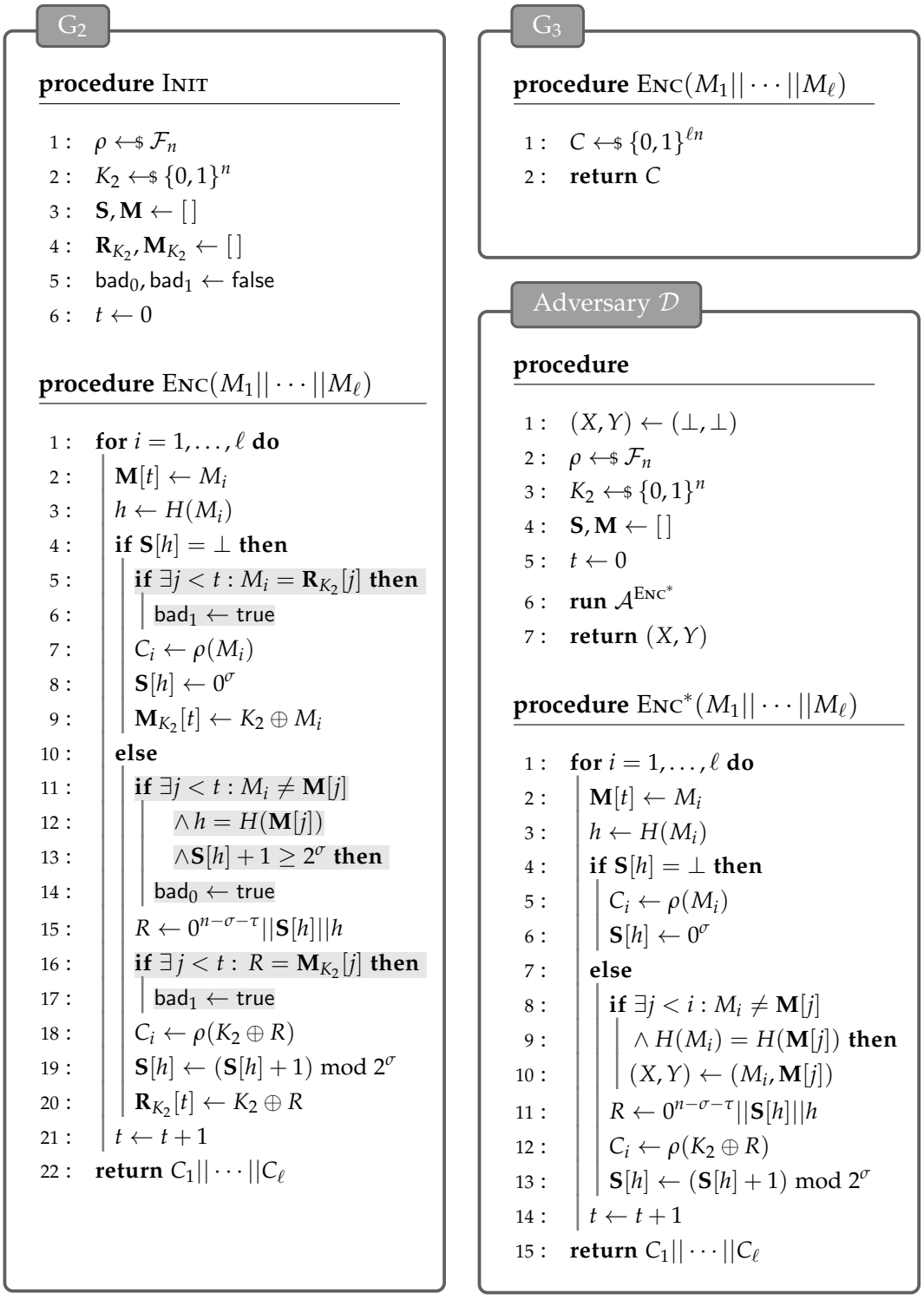

Figure 3.4: Games G_2–G_3 and the CR adversary $\mathcal{D}$ for the proof of Theorem 3.2. Bad events are highlighted.

Finally, we have that

$$\begin{aligned}\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{SCB}[E,H]}(\mathcal{A}) &= \Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_3(\mathcal{A})] \\ &\leq \mathbf{Adv}^{\text{PRP}}_{E}(\mathcal{B}) + \mathbf{Adv}^{\text{CR}}_{H}(\mathcal{D}) + \frac{\beta(\beta-1)}{2^{n+1}} + \frac{\beta}{2^n} \\ &\leq \mathbf{Adv}^{\text{PRP}}_{E}(\mathcal{B}) + \mathbf{Adv}^{\text{CR}}_{H}(\mathcal{D}) + \frac{\beta^2}{2^n}\end{aligned}$$

which concludes the proof. □

3.3 Correctness

In the initial design of the SCB mode, the correctness is primarily influenced by the term $\frac{2^\sigma \beta^2}{2^n}$. However, the presence of the term 2^σ is undesirable because its value grows exponentially with increasing the security parameter σ. This occurrence is due to the original scheme does not verify the counter during decryption, which allows more actual message blocks to be interpreted as a repetition signal. By incorporating the extension illustrated in Figure 3.1, we claim that the term 2^σ can be eliminated and prove it in Theorem 3.3.

Theorem 3.3 *For any* COR *adversary* $\mathcal{A}$ *with* $\beta := \beta(\mathcal{A})$*, we can construct a* CR *adversary* $\mathcal{B}$ *with* $q(\mathcal{B}) = \beta$ *such that*

$$\mathbf{Adv}^{\text{COR}}_{\mathsf{SCB}[E,H]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{CR}}_{H}(\mathcal{B}) + \frac{\beta^2}{2^n}$$

G_0

procedure Init

1 : $(K_1, K_2) \xleftarrow{\$} \{0,1\}^\kappa \times \{0,1\}^n$
2 : $\mathbf{S}, \mathbf{C}, \mathbf{T} \leftarrow [\,]$

procedure EncDec(M)

1 : $C \leftarrow \mathsf{SCB}[E,H].\mathcal{E}^{\mathbf{S}}_{K_1,K_2}(M)$
2 : $M' \leftarrow \mathsf{SCB}[E,H].\mathcal{D}^{\mathbf{C},\mathbf{T}}_{K_1,K_2}(C)$
3 : **return** M'

Figure 3.5: Games G_0 for the proof of Theorem 3.3.

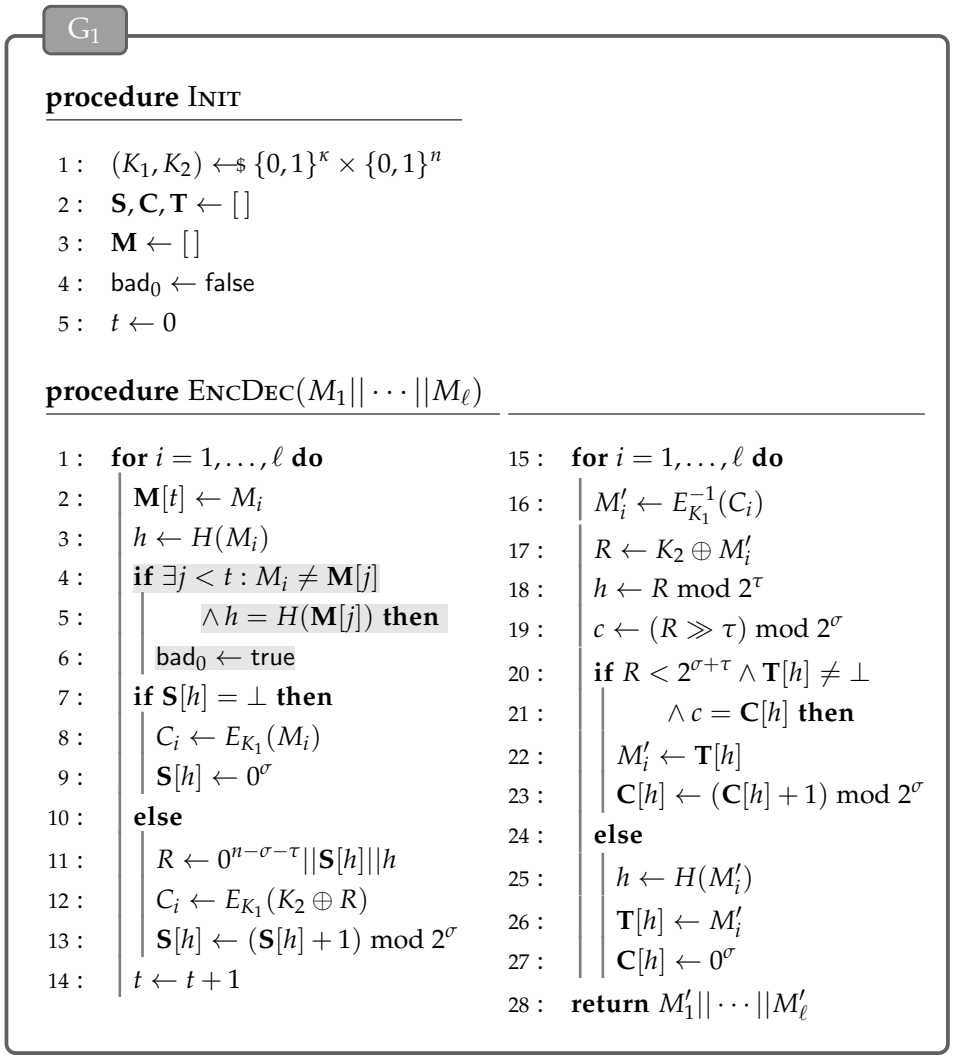

Figure 3.6: Game G_1 for the proof of Theorem 3.3. Bad events are highlighted.

Proof The proof describes five games, denoted by G_0–G_4, as defined in Figure 3.5, Figure 3.6, Figure 3.7, and Figure 3.8. Note that, we have that $G_0 = G^{\text{COR-0}}_{\mathsf{SCB}[E,H]}$ and $G_4 = G^{\text{COR-1}}_{\mathsf{SCB}[E,H]}$. Observe that G_0 and G_1 are equivalent, the behavior of G_1 and G_2 are identical until bad_0 is set to true, and G_2 and G_3 are equivalent until bad_1 is set to true, and G_3 and G_4 are equivalent. We denote W_i as the event that bad_i is set to true respectively.

To see that G_1 and G_2 are identical until the value of bad_0 is set to true, consider the case where there is no M_i such that $H(M_i) = H(\mathbf{M}[j])$, where $\mathbf{M}$ is a lookup table containing the block received before M_i. In this case, no adversary can distinguish between the game that encodes (Line 11–12 of G_1) and subsequently decodes a repetition signal (Line 16–22 of G_1), and the

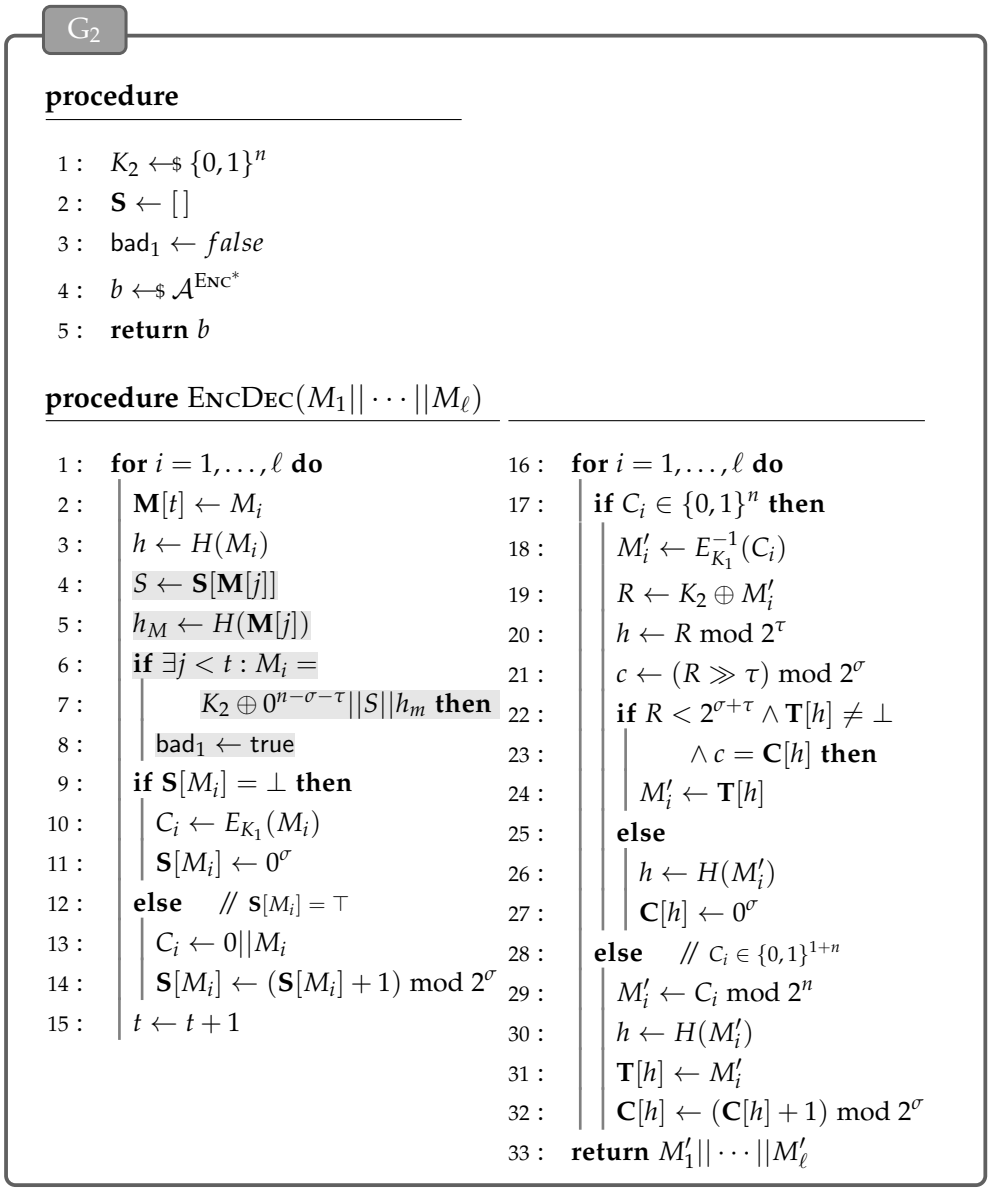

Figure 3.7: Game G_2 for the proof of Theorem 3.3. Bad events are highlighted.

game that simply adds a padding bit before the message block (Line 13 of G_2), and then ignore the bit later (Line 29 of G_2). Let $\mathcal{B}$ be an adversary that breaks collision resistance (CR) of H as defined in Figure 3.8, and we can reduce the probability $\Pr[W_0]$ to the advantage of $\mathcal{B}$ in the CR security game against H, denoted as $\mathbf{Adv}^{\text{CR}}_H(\mathcal{B})$ i.e., $\Pr[W_0] \leq \mathbf{Adv}^{\text{CR}}_H(\mathcal{B})$.

To observe that G_2 and G_3 are identical until bad_1 is set to true, consider the scenario where a block M_i that has never been queried before, after XORed

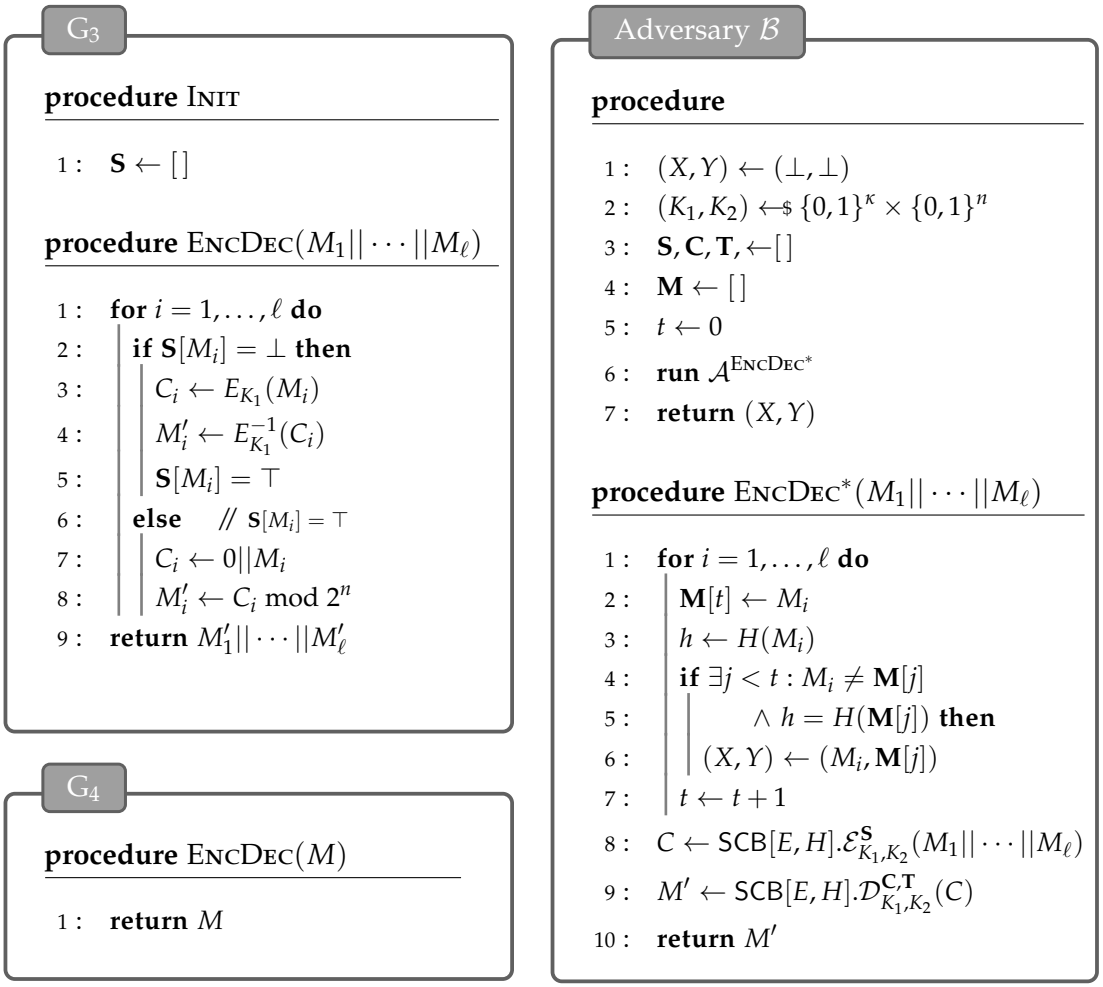

Figure 3.8: Games G_3, G_4 and adversary $\mathcal{B}$ for the proof of Theorem 3.3.

with K_2, has the structure of an expected repetition signal. This includes having $0^{n-\sigma-\tau}$ leading zeros, with the last τ bits corresponding to the hash of a previous valid message block, and the σ bits in between equaling the expected counter value. By Union Bound, we have that such probability happens with:

$$\begin{aligned}\Pr[W_1] &\leq \sum_{i=1}^{\beta} i \cdot 2^{-(n-\sigma-\tau)} \cdot 2^{-\sigma} \cdot 2^{-\tau} \\ &= \frac{\beta(\beta+1)}{2} \cdot 2^{-(n-\sigma-\tau)} \cdot 2^{-\sigma} \cdot 2^{-\tau} \\ &\leq \frac{\beta^2}{2^n}\end{aligned}$$

Furthermore, we have that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{COR}}_{\mathsf{SCB}[E,H]}(\mathcal{A}) &= \Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_4(\mathcal{A})] \\ &= \sum_{i=0}^{3} \mathrm{G}_i(\mathcal{A}) - \mathrm{G}_{i+1}(\mathcal{A})\end{aligned}$$

Note that we have G_0 and G_1 are identical, and G_3 and G_4 are identical. Thus it yields that

$$\begin{aligned}\Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_1(\mathcal{A})] = 0 \\ \Pr[\mathrm{G}_3(\mathcal{A})] - \Pr[\mathrm{G}_4(\mathcal{A})] = 0\end{aligned}$$

Let adversary $\mathcal{B}$ be defined as in Figure 3.8, we have that

$$\begin{aligned}\Pr[\mathrm{G}_1(\mathcal{A})] - \Pr[\mathrm{G}_2(\mathcal{A})] &\leq \Pr[W_0] \\ &\leq \mathbf{Adv}^{\mathrm{CR}}_{H}(\mathcal{B})\end{aligned}$$

By Fundamental Lemma of Game-Playing, we have

$$\begin{aligned}\Pr[\mathrm{G}_2(\mathcal{A})] - \Pr[\mathrm{G}_3(\mathcal{A})] &\leq \Pr[W_1] \\ &\leq \frac{\beta^2}{2^n}\end{aligned}$$

Finally, we have that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{COR}}_{\mathsf{SCB}[E,H]}(\mathcal{A}) &= \Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_4(\mathcal{A})] \\ &\leq \mathbf{Adv}^{\mathrm{CR}}_{H}(\mathcal{B}) + \frac{\beta^2}{2^n}\end{aligned}$$

which concludes the proof. □

Note that when a message block collides with the structure of a repetition signal. There are three possible cases to consider. Let $K_2 \oplus R = 0^{n-\sigma-\tau}||c_r||h$ and $K_2 \oplus M_{\mathsf{bad}} = 0^{n-\sigma-\tau}||c_m||h$ where R is a repetition signal of a message block M, and M_{bad} is an actual message block.

In case 1, it is assumed that $c_r = c_m$ and R is queried before M_{bad}. In this scenario, when processing M_{bad}, the expected counter value has been incremented after processing R, resulting in a correct interpretation of M_{bad} as a message block. In case 2, it is assumed that $c_r = c_m$ but M_{bad} is received before R, causing a misinterpretation of M_{bad} as a repetition signal and R as a message block, resulting in a "position change" of the blocks. In case 3, it is assumed that $c_m - 1$ is equal to the maximum counter value in the repetition signals of M. As a consequence, M_{bad} is misinterpreted as an additional repetition signal of M.

It should be noted that the correctness is not guaranteed in case 2 and case 3. Consequently, we set the bad event as when a new block aligns with the expected repetition signal's structure in the proof of Theorem 3.3. In our proposed scheme that utilizes counter validation, it is possible that a message block and a repetition signal can be misinterpreted having their positions swapped (case 2), leading to the decryption of *two blocks* being incorrect. However, this type of collision is highly unlikely, occurring with a probability of 2^{-n}, as demonstrated in the proof of Theorem 3.3, which is negligible in practice. Furthermore, this potential issue can be addressed by implementing appropriate ciphertext reordering to guarantee that the message block is queried after the repetition signal.

Chapter 4

Counter Validation for Recoverable SCB

4.1 The Scheme

We extend our counter validation scheme for *Recoverable* SCB (RSCB), which addresses the counter-related challenges that may arise from block reordering. This is achieved by incorporating a dynamic counter update mechanism, as well as two additional algorithms - *Tagged Decryption* and *Recovery*. When block reordering happens, it is expected that the repetition signal with a larger counter value may be received before the one with a smaller counter value. Rather than simply checking for counter equivalence, we now check if the counter falls within a window that contains the expected counter values. This window is dynamically updated as each expected repetition signal is processed. Additionally, it is possible that a repetition signal may be received before its corresponding actual message block. In this case, the repetition signals will be interpreted as an actual message block. The *Tagged Decryption* and *Recovery* algorithms are further employed to distinguish those repetition signals that have been wrongly interpreted as message blocks due to the limitations of our decryption algorithm.

We first redefine a *recoverable* SCB scheme with dynamic counter validation. We will then describe the counter-validation mechanism, *Tagged Decryption* and *Recovery* scheme. We will then prove the correctness of the scheme. Note that there is no need to prove the security as it is inherited from Theorem 3.2.

Definition 4.1 *Let $\kappa, n, \sigma, \tau, \delta \in \mathbb{N}$ with $\sigma + \tau < n$ and $\delta \leq 2^{\sigma}$, $E : \{0,1\}^{\kappa} \times \{0,1\}^{n} \to \{0,1\}^{n}$ a block cipher, and $H : \{0,1\}^{n} \to \{0,1\}^{\tau}$ a compression function. Also, let $\mathcal{S}$ be the set of $\{0,1\}^{\tau} \to \{0,1\}^{\sigma} \cup \{\perp\}$ look-up tables, $\mathcal{T}$ be the set of $\{0,1\}^{\tau} \to \{0,1\}^{n} \cup \{\perp\}$ look-up tables, and $\mathcal{C}$ be the set of $\{0,1\}^{\tau} \to \{0,1\}^{\sigma}$ look-up table. Let $\mathcal{W}$ be the set of $\{0,1\}^{\tau} \times \{0,1\}^{\sigma} \to \{0,1\} \cup \{\perp\}$ look-up tables. The Recoverable Secure Code Book (*RSCB*) mode of encryption is defined*

as $\mathsf{RSCB}[E, H] = (\mathcal{E}, \mathcal{D}, \widetilde{\mathcal{D}}, \mathcal{R})$ *with key space* $\mathcal{K} = \{0,1\}^\kappa \times \{0,1\}^n$*, encryption state space* $\mathcal{S}$*, decryption state spaces* $\mathcal{C}$*,* $\mathcal{T}$*, and* $\mathcal{W}$*, encryption scheme* $\mathcal{E}$ *and decryption scheme* $\mathcal{D}$ *as defined in Figure 4.1, and tagged decryption scheme* $\widetilde{\mathcal{D}}$ *and recovery scheme* $\mathcal{R}$ *as defined in Figure 4.2.*

4.1.1 Sliding Windows Counter Update

Our mechanism is based on the assumption that, the block reordering is a *permutation* $\pi : [q] \to [q]$ such that

$$\forall\, i \in [q] \colon \max\{1, i - \delta\} \leq \pi(i) \leq \min\{q, i + \delta\}$$

where i represents the block's position being queried during encryption, q is the total number of n-bit blocks queried and δ is the maximum number of blocks that can be transmitted during a session.

The procedure for validating a repetition signal involves verifying whether the counter value c falls within an acceptable range of values, as illustrated in Figure 4.1. We streamline this validation process using a lookup table $\mathbf{W}$ to keep track of received repetition signals with counter values within the acceptable range. This lookup table is referred to as a *window*. We fix a window $\mathbf{W}$ of size 2δ and dynamically update it to maintain the size to accommodate the possible values of $\pi(i)$. Consider that, in the extreme case, it is possible that 2δ repetition signals of the same actual message block are queried in two consecutive sessions. In order to monitor the current window, we utilize another look-up table $\mathbf{C}$ to indicate the first element of the current window. We call this lookup table $\mathbf{C}$ an *indicator*. When an actual message block with hash h is queried, the window is initialized with values $\mathbf{W}[h, 0^\sigma] \leftarrow 0, \cdots, \mathbf{W}[h, 2\delta - 1] \leftarrow 0$, meaning that we are expecting the repetition signals with such counter values. The indicator is also initialized with the value 0^σ to point the first element of the initial window. For a hash h and a counter value c, we use $\mathbf{W}[h, c] = 0$ to denote that we are expecting the repetition signal with c.

Upon processing a repetition signal, with c as counter value and h as the hash of its corresponding actual message block, the entry $\mathbf{W}[h, c]$ is assigned the value 1, indicating that the repetition signal has been queried. Starting from the current indicator $\mathbf{C}[h]$, we count how many repetition signals with consecutive counters have been queried by performing a search within the current window until encountering entry with value 0. Assuming that $\mathbf{W}[h, c] = 0$, it means that all repetition signals with a counter c' less than c have been queried. Consequently, the window can be "slid" by the number of queried repetition signals, and the indicator is updated respectively. Specifically, the check for $\mathbf{W}[h, c] = \perp$ at Line 15 means that there is no such entry in the window, we then set $\mathbf{W}[h, c] \leftarrow 0$ meaning that we are expecting new counter values.

$\mathsf{RSCB}[E,H].\mathcal{E}^{\mathbf{S}}_{K_1,K_2}(M_1||\cdots||M_\ell)$

1 : **for** $i = 1,\ldots,\ell$ **do**
2 : $\quad h \leftarrow H(M_i)$
3 : $\quad$ **if** $\mathbf{S}[h] = \perp$ **then**
4 : $\quad\quad C_i \leftarrow E_{K_1}(M_i)$
5 : $\quad\quad \mathbf{S}[h] \leftarrow 0^\sigma$
6 : $\quad$ **else**
7 : $\quad\quad R \leftarrow 0^{n-\sigma-\tau}||\mathbf{S}[h]||h$
8 : $\quad\quad C_i \leftarrow E_{K_1}(K_2 \oplus R)$
9 : $\quad\quad \mathbf{S}[h] \leftarrow (\mathbf{S}[h]+1) \bmod 2^\sigma$
10 : **return** $C_1||\cdots||C_\ell$

$\mathsf{RSCB}[E,H].\mathcal{D}^{\mathbf{W},\mathbf{C},\mathbf{T}}_{K_1,K_2}(C_1||\cdots||C_\ell)$

1 : **for** $i = 1,\ldots,\ell$ **do**
2 : $\quad M_i \leftarrow E^{-1}_{K_1}(C_i)$
3 : $\quad R \leftarrow K_2 \oplus M_i$
4 : $\quad h \leftarrow R \bmod 2^\tau$
5 : $\quad c \leftarrow (R \gg \tau) \bmod 2^\sigma$
6 : $\quad$ **if** $R < 2^{\sigma+\tau} \wedge \mathbf{T}[h] \neq \perp$
7 : $\quad\quad \wedge\, \mathbf{W}[h,c] = 0$ **then**
8 : $\quad\quad M_i \leftarrow \mathbf{T}[h]$
9 : $\quad\quad \mathbf{W}[h,c] \leftarrow 1$
10 : $\quad\quad$ **for** $j = \mathbf{C}[h],\ldots,\mathbf{C}[h]+2\delta-1$ **do**
11 : $\quad\quad\quad$ **if** $\mathbf{W}[h,j] = 0$ **then**
12 : $\quad\quad\quad\quad \mathbf{C}[h] \leftarrow j$
13 : $\quad\quad\quad\quad$ **break**
14 : $\quad\quad$ **for** $j = \mathbf{C}[h],\ldots,\mathbf{C}[h]+2\delta-1$ **do**
15 : $\quad\quad\quad$ **if** $\mathbf{W}[h,j] = \perp$ **then**
16 : $\quad\quad\quad\quad \mathbf{W}[h,j] \leftarrow 0$
17 : $\quad$ **else**
18 : $\quad\quad h \leftarrow H(M_i)$
19 : $\quad\quad \mathbf{T}[h] \leftarrow M_i$
20 : $\quad\quad \mathbf{C}[h] \leftarrow 0^\sigma$
21 : $\quad\quad$ **for** $j = \mathbf{C}[h],\ldots,\mathbf{C}[h]+2\delta-1$ **do**
22 : $\quad\quad\quad \mathbf{W}[h,j] \leftarrow 0$
23 : **return** $M_1||\cdots||M_\ell$

Figure 4.1: Encryption and Decryption algorithms of $\mathsf{RSCB}[E,H]$ with sliding windows. Here we use notation $\mathbf{W}[h,c] = 0$ to show that we are expecting repetition signal with counter c, and use $\mathbf{W}[h,c] = 1$ to show that we have seen the repetition signal with counter c.

In the event that repetition signals are queried before their corresponding message block , since no entry has been established in the look-up table $\mathbf{T}$ to map these repetition signals to the corresponding message block, these repetition signals will be interpreted as actual message blocks first during the decryption. In the recovery phase, we need to distinguish and correct these misinterpreted signals separately.

We provide further explanation on our mechanism for validating repetition signals with Example 4.2.

Example 4.2 *Consider the scenario where $\delta = 3$ blocks are received within a session d. Let $M_1,\cdots,M_{12} \in \{0,1\}^n$ represent individual message blocks, and*

let $h := H(M_i)$ *for* $i \in [12]$. *Define* C_i *as the encryption of* M_i *for* $i \in [12]$. *Let* $C_1||C_2||\cdots||C_{12} := \mathsf{RSCB}[E,H].\mathcal{E}(M_1||M_2||\cdots||M_{12})$ *and* $M'_1||M'_2||\cdots||M'_{12} := \mathsf{RSCB}[E,H].\mathcal{D}(C_1||C_2||\cdots||C_{12})$. *Suppose* $M_1 = M_4 = M_5 = M_6 = M_8 = M_9 = M_{10} = M_{11} = M_{12}$. *Additionally, assume that* $K_2 \oplus M_7 = 0^{n-\sigma-\tau}||0^{\sigma}+2||h_1$. *The order in which* $C_1, \cdots, C_{12}$ *are received is as follows:*

$$C_1\ C_2\ C_3\ C_6\ C_4\ C_5\ C_7\ C_9\ C_8\ C_{12}\ C_{11}\ C_{10}$$

When C_1 *is queried, we interpret* C_1 *as an actual message block and initialize the windows* $\mathbf{W}[h_1,0], \cdots, \mathbf{W}[h_1,5]$ *with* 0. *(Here we slightly abuse the notation* $\mathbf{W}[h,i]$ *as* $\mathbf{W}[h,[i]_\delta]$*). We then set the indicator* $\mathbf{C}[h_1]$ *as* 0^{σ}. *When* C_6 *is processed, we know* $c = 2$, *the entry* $\mathbf{W}[h_1,2]$ *is then set to* 1. *Following the processing of* C_4 *and* C_5, *which are the repetition signals with counter values of* 0 *and* 1, *respectively, we then set the entry* $\mathbf{W}[h_1,0]$ *and* $\mathbf{W}[h_1,1]$ *to* 1. *Observe that since we have the repetition signals with counter value* $0,1,2$ *have been queried. We slide the window by initializing another three entries* $\mathbf{W}[h_1,6]$, $\mathbf{W}[h_1,7]$ *and* $\mathbf{W}[h_1,8]$ *with* 0. *And we shift the indicator* $\mathbf{C}[h_1]$ *to* 3 *accordingly. Upon processing* C_7, *since we have set* $\mathbf{W}[h_1,2]$ *to* $\perp$ *when* C_6 *was queried. Thus we interpret it as an actual message block. At this point, we have that the current window includes* $\mathbf{W}[h_1,3] = \mathbf{W}[h_1,4] = \mathbf{W}[h_1,5] = \mathbf{W}[h_1,6] = \mathbf{W}[h_1,7] = \mathbf{W}[h_1,8]$, *which corresponds to the counter value in* $M_8, M_9, M_{10}, M_{11}, M_{12}$. *Thus after the decryption process, it holds that* $M'_1 = M'_6 = M'_8 = M'_9 = M'_{10} = M'_{11} = M'_{12}$, *and* $M'_7 = M_7$. *Note that* $M'_4 \neq M'_1$ *and* $M'_5 \neq M'_1$ *but we will further correct these two blocks in the recovery phase.*

It should be noted that since an entry is marked as 1 once a block with the structure of an expected repetition signal has been queried, in the event of a collision between a message block M and a repetition signal R, i.e., $K_2 \oplus M = R$, the decryption algorithm interprets one as a message block and the other one as a repetition signal. Thus, similar to the decryption algorithm that does not involve block reordering, it is also possible for a colliding message block and a repetition signal to be incorrectly decrypted with their position altered if the message block is queried before the repetition signal. Nevertheless, the probability of this occurrence is 2^{-n}, which is negligible in practice. Moreover, this issue can be resolved by utilizing an appropriate reordering permutation to ensure that the message block is queried after the repetition signal.

4.1.2 Tagged Decryption and Recover

The recovery phase comprises two supplementary algorithms, *Tagged Decryption* and *Recovery*, that have been designed to handle repetition signals that are mistakenly interpreted as message blocks during decryption, as illustrated in Figure 4.2. It should be noted that these misinterpreted signals must be received before the actual message block, according to the

$\mathsf{RSCB}[E,H].\widetilde{\mathcal{D}}^{\mathbf{C},\mathbf{W},\mathbf{T}}_{K_1,K_2}(C_1||\cdots||C_\ell)$

$M_1||\cdots||M_\ell \leftarrow \mathsf{RSCB}[E,H].\mathcal{D}^{\mathbf{C},\mathbf{W},\mathbf{T}}_{K_1,K_2}(C_1||\cdots||C_\ell)$
for $i = 1,\ldots,\ell$ **do**
  $R \leftarrow K_2 \oplus M_i$
  if $R < 2^{\sigma+\tau}$ **then**
    $t_i \leftarrow 1$
  else
    $t_i \leftarrow 0$
return $(M_1||\cdots||M_\ell, t_1||\cdots||t_\ell)$

$\mathsf{RSCB}[E,H].\mathcal{R}_{K_2}((M_{1,1}||\cdots||M_{1,\ell_1}, t_{1,1}||\cdots||t_{1,\ell_1}),\ldots,(M_{s,1}||\cdots||M_{s,\ell_s}, t_{s,1}||\cdots||t_{s,\ell_s}))$

$\mathbf{T},\mathbf{M} \leftarrow [\,]$
for $i = 1,\ldots,s$ **do**
  for $j = 1,\ldots,\ell_i$ **do**
    $h \leftarrow H(M_{i,j})$
    $\mathbf{T}[h] \leftarrow M_{i,j}$
    $\mathbf{M}[h] \leftarrow \perp$
for $i = 1,\ldots,s$ **do**
  for $j = 1,\ldots,\ell_i$ **do**
    $R \leftarrow K_2 \oplus M_{i,j}$
    $h \leftarrow R \bmod 2^\tau$
    $c \leftarrow (R \gg \tau) \bmod 2^\sigma$
    if $t_{i,j} = 1 \wedge \mathbf{T}[h] \neq \perp \wedge \mathbf{M}[h] = \perp$
      $\wedge\, c \in \{0,\cdots,2\delta-1\}$ **then**
      $M'_{i,j} \leftarrow \mathbf{T}[M_{i,j}]$
    else
      $M'_{i,j} \leftarrow M_{i,j}$
      $h \leftarrow H(M_{i,j})$
      $\mathbf{M}[h] \leftarrow \top$
return $M_{1,1}||\cdots||M_{1,\ell_1},\cdots,M_{s,1}||\cdots||M_{s,\ell_s}$

Figure 4.2: Tagged decryption and recovery algorithms of $\mathsf{RSCB}[E,H]$.Note that R does not use K_1, so we slightly abused notation in the function declaration.

decryption algorithm. Additionally, the counter of a repetition signal must be in $\{0,\ldots,2\delta-1\}$, which match with the initial window in the decryption

algorithm.

In the Tagged Decryption algorithm, we first tag the blocks that have a structure of a repetition signal. In the Recovery algorithm, we verify if a block is actually a repetition signal by checking the condition above. We use a flag to check if the actual message block has been received, then any subsequent blocks should be interpreted as an actual message block, since the repetition signals that are queried after the actual message block have been handled during decryption.

4.2 Correctness with Reordering

We first consider the notion of COR-WR* in which that tagged-decryption and recovery algorithm are not involved, and the reordering permutation π ensures that a repetition signal is queried only after the corresponding actual message block has been queried.

Theorem 4.3 *For any* COR-WR* *adversary* $\mathcal{A}$ *with* $\beta := \beta(\mathcal{A})$*, assuming block reordering is a permutation* $\pi : [\beta] \to [\beta]$ *such that* $\forall i \in [\beta] : \max\{1, i-\delta\} \leq \pi(i) \leq \min\{\beta, i+\delta\}$ *where* δ *is the maximum number of blocks that can be queried during a session, and and a repetition signal is queried only after the corresponding actual message block has been queried. We can construct a* CR *adversary* $\mathcal{B}$ *with* $q(\mathcal{B}) = \beta$ *such that*

$$\mathbf{Adv}^{\text{COR-WR}^*}_{\mathsf{RSCB}[E,H]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{CR}}_{H}(\mathcal{B}) + \frac{\beta^2\delta}{2^{n-1}}$$

Proof Let G_0–G_4 be defined as in Figure 4.3, Figure 4.4, Figure 4.5, and Figure 4.6. As in the proof of Therorem 3.3, the behaviors of G_1 and G_2 are identical until bad_0 is set to true, which is bounded by $\mathbf{Adv}^{\text{CR}}_{H}(\mathcal{B})$.

Consider that, by the assumption we have for block reordering, a block queried at position i during encryption may be queried at the position $\pi(i)$ during decryption where $\max\{1, i-\delta\} \leq \pi(i) \leq \min\{\beta, i+\delta\}$ during decryption. To accommodate the case that multiple repetition signals of the same corresponding message blocks are queried within the same or two consecutive session, we set the window of size 2δ. However, this configuration may allow an actual message to be misinterpreted as a repetition signal.

Observe that the behaviors of G_2 and G_3 are identical until bad_1 is set to true. Consider that when bad_1 does not happen, every actual message block is in $\{0,1\}^n$ and every repetition signal is in $\{0,1\}^{n+\sigma+1}$, which is identical to the behavior of game G_3. When bad_1 is set to true, there is an actual message block $M_{\mathsf{bad}} \in \{0,1\}^n$ that has the structure of an expected repetition signal, that is $M_{\mathsf{bad}} \oplus K_2 = R$, where R is an expected repetition signal. We let $M_{\mathsf{bad}} \oplus K_2 = 0^{n-\sigma-\tau}||c_m||h_m$ for some $c_m \in \{0,1\}^\sigma$ and $h_m \in \{0,1\}^\tau$.

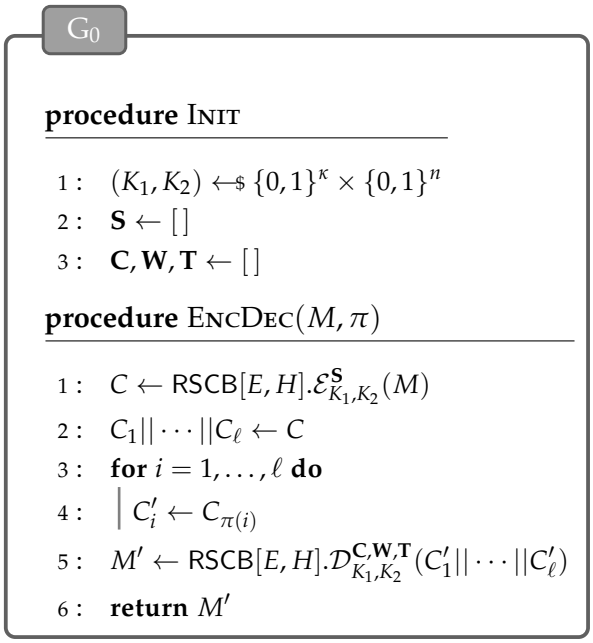

Figure 4.3: Games G_0 for the proof of Theorem 4.3.

Assuming h_m represents the hash of an actual message block M', we consider two cases for a bad event. In the first case, if the block M' is not repeated, M_{bad} is interpreted as a valid repetition signal if c_m falls within the set of values $\{0, \ldots, 2\delta - 1\}$, and M_{bad} is queried after M' during decryption. This is because the initial window, after encountering M' for the first time, is set to $\mathbf{W}[h_m, 0^\sigma] \leftarrow 0, \cdots, \mathbf{W}[h_m, 2\delta - 1] \leftarrow 0$. On the other hand, if M' is repeated, we first consider a scenario where all entries in current window, except for the one indicated by $\mathbf{C}[h]$, have been marked as 1. In this case, M_{bad} is considered valid only if $c_M = \mathbf{C}[h]$. Furthermore, if a window has just been slid, there are 2δ values that can result in c_m being an expected counter value. Therefore, the number of values that can cause c_m to be an expected counter value can vary from 1 to 2δ, depending on the reordering permutation. This range is a result of the dynamically adjusted window, which has a size of 2δ.

Therefore, the probability that the above event happens is bounded by $\frac{2\delta}{2^n}$ in either case. Using the Union Bound, we can determine that this probability

G$_1$

procedure Init

$(K_1, K_2) \xleftarrow{\$} \{0,1\}^\kappa \times \{0,1\}^n$
$\mathbf{S}, \mathbf{C}, \mathbf{T} \leftarrow [\,]$
$\mathbf{M} \leftarrow [\,]$
$\mathsf{bad}_0 \leftarrow \mathsf{false}$
$t \leftarrow 0$

procedure EncDec$(M_1 || \cdots || M_\ell, \pi)$

for $i = 1, \ldots, \ell$ **do**
  $h \leftarrow H(M_i)$
  if $\exists j \neq i : H(M_j) = h$ **then**
    $\mathsf{bad}_0 \leftarrow \mathsf{true}$
  if $\mathbf{S}[h] = \perp$ **then**
    $C_i \leftarrow E_{K_1}(M_i)$
    $\mathbf{S}[h] \leftarrow 0^\sigma$
  else
    $R \leftarrow 0^{n-\sigma-\tau} || \mathbf{S}[h] || h$
    $C_i \leftarrow E_{K_1}(K_2 \oplus R)$
    $\mathbf{S}[h] \leftarrow (\mathbf{S}[h] + 1) \bmod 2^\sigma$
for $i = 1, \ldots, \ell$ **do**
  $C'_i \leftarrow C_{\pi(i)}$
for $i = 1, \ldots, \ell$ **do**
  $M'_i \leftarrow E^{-1}_{K_1}(C'_i)$
  $R \leftarrow K_2 \oplus M'_i$
  $h \leftarrow R \bmod 2^\tau$
  $c \leftarrow (R \gg \tau) \bmod 2^\sigma$
  if $R \leq 2^{\sigma+\tau} \wedge \mathbf{T}[h] \neq \perp$
    $\wedge\, \mathbf{W}[h, c] = 0$ **then**
    $M'_i \leftarrow \mathbf{T}[h]$
    $\mathbf{W}[h, c] \leftarrow 1$
    for $j = \mathbf{C}[h], \ldots, \mathbf{C}[h] + 2\delta - 1$ **do**
      if $\mathbf{W}[h, j] = 0$ **then**
        $\mathbf{C}[h] \leftarrow j$
        break
    for $j = \mathbf{C}[h], \ldots, \mathbf{C}[h] + 2\delta - 1$ **do**
      if $\mathbf{W}[h, j] = \perp$ **then**
        $\mathbf{W}[h, j] \leftarrow 0$
  else
    $h \leftarrow H(M'_i)$
    $\mathbf{T}[h] \leftarrow M'_i$
    $\mathbf{C}[h] \leftarrow 0^\sigma$
    for $j = \mathbf{C}[h], \ldots, \mathbf{C}[h] + 2\delta - 1$ **do**
      $\mathbf{W}[h, j] \leftarrow 0$
return $M'_1 || \cdots || M'_\ell$

Figure 4.4: Game G$_1$ for proof of Theorem 4.3. Bad event is highlighted.

occurs with an upper bound as

$$\begin{aligned}
\Pr[\mathrm{G}_2(\mathcal{A})] - \Pr[\mathrm{G}_3(\mathcal{A})] &\leq \sum_{i=1}^{\beta} i \cdot 2^{-(n-\sigma-\tau)} \cdot 2^{-\sigma} \cdot 2^{-\tau} \cdot 2\delta \\
&= \frac{\beta(\beta+1)}{2} \cdot 2^{-(n-\sigma-\tau)} \cdot 2^{-\sigma} \cdot 2^{-\tau} \cdot 2\delta \\
&\leq \frac{\beta^2 \delta}{2^{n-1}}
\end{aligned}$$

G$_2$

procedure INIT

1 : $(K_1, K_2) \leftarrow\$ \{0,1\}^\kappa \times \{0,1\}^n$
2 : $\mathbf{S} \leftarrow [\,]$
3 : $\mathbf{C}, \mathbf{W}, \mathbf{T} \leftarrow [\,]$
4 : $\mathsf{bad}_1 \leftarrow \mathsf{false}$

procedure ENCDEC$(M_1 || \cdots || M_\ell, \pi)$

1 : **for** $i = 1, \ldots, \ell$ **do**
2 : **if** $\mathbf{S}[M_i] = \perp$ **then**
3 : $C_i \leftarrow E_{K_1}(M_i)$
4 : $\mathbf{S}[M_i] \leftarrow 0^\sigma$
5 : **else**
6 : $C_i \leftarrow 0 || \mathbf{S}[M_i] || M_i$
7 : $\mathbf{S}[M_i] \leftarrow (\mathbf{S}[M_i] + 1) \bmod 2^\sigma$
8 : **for** $i = 1, \ldots, \ell$ **do**
9 : $C'_i \leftarrow C_{\pi(i)}$
10 : **for** $i = 1, \ldots, \ell$ **do**
11 : **if** $C'_i \in \{0,1\}^n$ **then**
12 : $M'_i \leftarrow E^{-1}_{K_1}(C'_i)$
13 : $R \leftarrow K_2 \oplus M'_i$
14 : $h \leftarrow R \bmod 2^\tau$
15 : $c \leftarrow (R \gg \tau) \bmod 2^\sigma$
16 : **if** $R < 2^{\sigma+\tau} \wedge \mathbf{T}[h] \neq \perp$
17 : $\wedge \mathbf{W}[h, c] = 0$ **then**
18 : $\mathsf{bad}_1 \leftarrow \mathsf{true}$
19 : $M'_i \leftarrow \mathbf{T}[h]$
20 : $\mathbf{W}[h, c] \leftarrow 1$
21 : **for** $j = \mathbf{C}[h], \ldots, \mathbf{C}[h] + 2\delta - 1$ **do**
22 : **if** $\mathbf{W}[h, j] = 0$ **then**
23 : $\mathbf{C}[h] \leftarrow j$
24 : **break**
25 : **for** $j = \mathbf{C}[h], \ldots, \mathbf{C}[h] + 2\delta - 1$ **do**
26 : **if** $\mathbf{W}[h, j] = \perp$ **then**
27 : $\mathbf{W}[h, j] \leftarrow 0$
28 : **else**
29 : $\mathbf{C}[h] \leftarrow 0^\sigma$
30 : **for** $j = \mathbf{C}[h], \ldots, \mathbf{C}[h] + 2\delta - 1$ **do**
31 : $\mathbf{W}[h, j] \leftarrow 0$
32 : **else** // $C_i \in \{0,1\}^{1+\sigma+n}$
33 : $M'_i \leftarrow C'_i \bmod 2^n$
34 : $h \leftarrow H(M'_i)$
35 : $c \leftarrow (C'_i \gg n) \bmod 2^\sigma$
36 : $\mathbf{T}[h] \leftarrow M'_i$
37 : $\mathbf{W}[h, c] \leftarrow 1$
38 : **for** $j = \mathbf{C}[h], \ldots, \mathbf{C}[h] + 2\delta - 1$ **do**
39 : **if** $\mathbf{W}[h, j] = 0$ **then**
40 : $\mathbf{C}[h] \leftarrow j$
41 : **break**
42 : **for** $j = \mathbf{C}[h], \ldots, \mathbf{C}[h] + 2\delta - 1$ **do**
43 : **if** $\mathbf{W}[h, j] = \perp$ **then**
44 : $\mathbf{W}[h, j] \leftarrow 0$
45 : **return** $M'_1 || \cdots || M'_\ell$

Figure 4.5: Game G$_2$ for proof of Theorem 4.3. Bad event is highlighted.

Further, we have that

$$\begin{aligned}\mathbf{Adv}^{\text{COR-WR}^*}_{\mathsf{RSCB}[E,H]}(\mathcal{A}) &= \Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_4(\mathcal{A})] \\ &= \sum_{i=0}^{3} \Pr[\mathrm{G}_i(\mathcal{A})] - \Pr[\mathrm{G}_{i+1}(\mathcal{A})]\end{aligned}$$

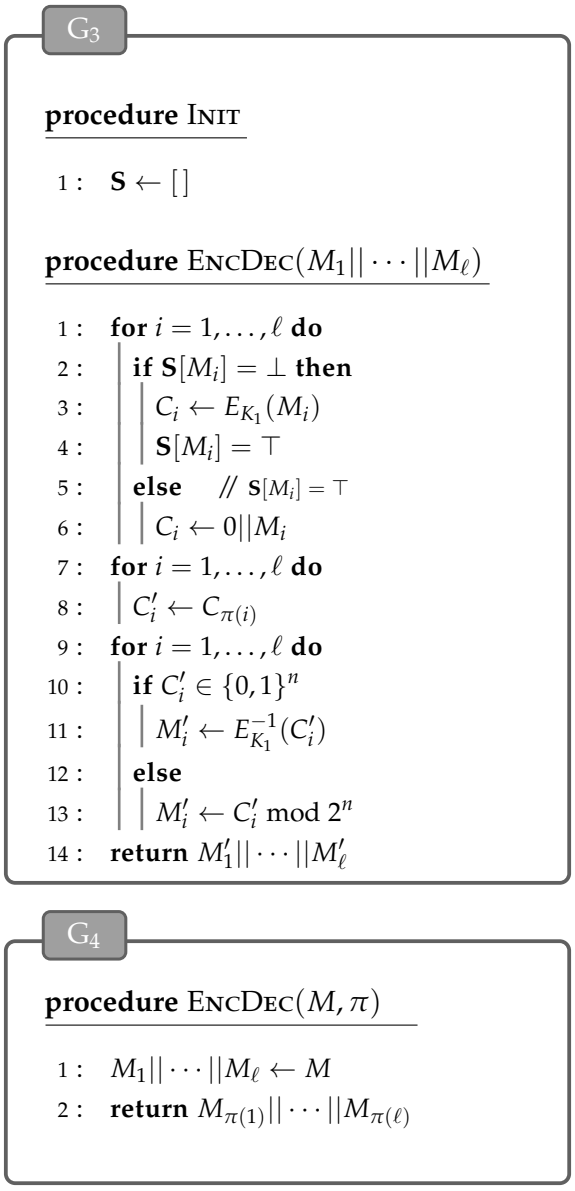

Figure 4.6: Game G_3 and G_4 for proof of Theorem 4.3.

From the proof of Theorem 3.3, it yields that

$$\Pr[G_1(\mathcal{A})] - \Pr[G_2(\mathcal{A})] \leq \mathbf{Adv}_H^{\mathrm{CR}}(\mathcal{B})$$

Since G_0 and G_1 are equivalent, and G_3 and G_4 are equivalent, we have

$$\Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] = 0$$
$$\Pr[G_3(\mathcal{A})] - \Pr[G_4(\mathcal{A})] = 0$$

Finally,

$$\begin{aligned}\mathbf{Adv}^{\text{COR-WR}^*}_{\mathsf{RSCB}[E,H]}(\mathcal{A}) &= \Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_4(\mathcal{A})] \\ &\leq \mathbf{Adv}^{\text{CR}}_{H}(\mathcal{B}) + \frac{\beta^2\delta}{2^{n-1}}\end{aligned}$$

which concludes the proof. □

We now further consider the notion of COR-WR in which the tagged decryption and recovery algorithm is involved.

Theorem 4.4 *For any* COR-WR *adversary* $\mathcal{A}$ *with* $\beta := \beta(\mathcal{A})$*, assuming block reordering is a permutation* $\pi : [\beta] \to [\beta]$ *such that* $\forall\, i \in [\beta] : \max\{1, i-\delta\} \leq \pi(i) \leq \min\{\beta, i+\delta\}$ *where d is the maximum number of blocks that can be queried during a session. We can construct a* CR *adversary* $\mathcal{B}$ *with* $q(\mathcal{B}) = \beta$ *such that*

$$\mathbf{Adv}^{\text{COR-WR}}_{\mathsf{RSCB}[E,H]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{CR}}_{H}(\mathcal{B}) + \frac{\beta^2\delta}{2^{n-1}}$$

Proof (sketch) We modify the games G_0 illustrated in Figure 4.3 but including the tagged decryption and recovery algorithm in the game. We also modify the games G_1 and G_2, which are presented in Figure 4.4 and Figure 4.5 such that the oracle Dec tags the ambiguous block and applies the recovery algorithm. We let the games G_3 and G_4 remain the same as in Figure 4.6.

We further consider a bad event that there is a message block $M_{\mathsf{bad}} = 0^{n-\sigma-\tau}||c||h$ with $c \in \{0, \cdots, 2\delta-1\}$ and $h = H(M)$ where M is an actual message block. Note that this bad event can be considered as a special case of bad_1 in which $\mathbf{W}[h, c] = 0$ for some $c \in \{0, \cdots, 2\delta-1\}$. Thus the probability that the bad event happens is upper bounded by $\frac{2\delta}{2^n}$. Thus similarly, we can derive that

$$\mathbf{Adv}^{\text{COR-WR}}_{\mathsf{RSCB}[E,H]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{CR}}_{H}(\mathcal{B}) + \frac{\beta^2\delta}{2^{n-1}}$$

which concludes the proof. □

Chapter 5

Length-Preserving AEAD with SCB Mode

We demonstrate the feasibility of constructing a semantically secure length-preserving AEAD scheme. Our construction follows a variant of the *Encode-then-Encipher* (EtE) paradigm by Bellare and Rogaway in [BR00]. The core idea of the EtE paradigm is to first encode a message and *encipher* it with a VIL cipher. Authenticity of the ciphertext can be verified through such encoding. Notably, if there is existing redundancy in plaintext, such redundancy can be exploited to enhance or establish the authenticity.

At high level, our construction can be seen as implementing the *Encrypt-then-MAC* paradigm with the SCB mode, while substituting the conventional MAC scheme with a VIL cipher. We create the redundancy in plaintext through repeated blocks, which is the reason why this construction is only viable under the condition that there exists at least one duplicated block within the message. To maintain simplicity, we assume that the plaintext is a multiple of the block size. Alternatively, the ciphertext stealing technique [RWZ12] can be employed if this condition isn't met, adjusting the parsing process to consistently designate the first n bits as M_L and the remainder as M_R.

We begin by providing a formal definition of the proposed scheme. We will then demonstrate the scheme's security by showing that it satisfies indistinguishability under chosen-plaintext attacks (IND-CPA) and ciphertext integrity (INT-CTXT). We omit the proof for correctness of the scheme since it mainly follows from Theorem 3.3.

5.1 The Scheme

SCB mode handles the block repetition by constructing a repetition signal of a special format, which results in redundancy. Abstractly, we can model

AE-SCB$[E, \widetilde{E}, H_1, H_2].\mathcal{E}^{\mathbf{M},\mathbf{S}}_{K_1,K_2,K_3}(M, A)$

$M_1||\cdots||M_\ell \leftarrow M$
for $i = 1, \ldots, \ell$ **do**
  $h \leftarrow H_1(M_i)$
  if $\mathbf{S}[h] = \perp$ **then**
    $C_i \leftarrow E_{K_1}(M_i)$
    $\mathbf{S}[h] \leftarrow 0^\sigma$
    $\mathbf{M}[M_i] \leftarrow C_i$
  else
    if $R^* = \perp$ **then**
      $C' \leftarrow \mathbf{M}[M_i]$
      $R^* \leftarrow 0^{n-\sigma-\tau}||[i]_\sigma||H_1(C')$
      $C_i \leftarrow \varepsilon$
      $R^* \leftarrow \top$
    else
      $R \leftarrow 0^{n-\sigma-\tau}||\mathbf{S}[h]||h$
      $C_i \leftarrow E_{K_1}(K_2 \oplus R)$
      $\mathbf{S}[h] \leftarrow (\mathbf{S}[h] + 1) \bmod 2^\sigma$
$h \leftarrow H_2(C_1||\cdots||C_\ell||A)$
$C^* \leftarrow \widetilde{E}_{K_3}(h, R^*)$
return $C^*||C_1||\cdots||C_\ell$

AE-SCB$[E, \widetilde{E}, H_1, H_2].\mathcal{D}^{\mathbf{C},\mathbf{T}}_{K_1,K_2,K_3}(C, A)$

$C_L||C_R \leftarrow C$ // $|C_L| = n$
$h \leftarrow H_2(C_R||A)$
$R^* \leftarrow \widetilde{E}^{-1}_{K_3}(h, C_L)$
$h^* \leftarrow R^* \bmod 2^\tau$
$C_{R,1}||\cdots||C_{R,\ell-1} \leftarrow C_R$
$t \leftarrow (R^* \gg \tau) \bmod \sigma$
if $R^* > 2^{\sigma+\tau} \vee \forall 1 \le i \le \ell - 1 :$
    $H_1(C_{R,i}) \neq h^*$ **then**
  return $\perp$
for $i = 1, \ldots, \ell - 1$ **do**
  $M_i \leftarrow E^{-1}_{K_1}(C_{R,i})$
  $R \leftarrow K_2 \oplus M_i$
  $h \leftarrow R \bmod 2^\tau$
  $c \leftarrow (R \gg \tau) \bmod 2^\sigma$
  if $R < 2^{\sigma+\tau} \wedge \mathbf{T}[h] \neq \perp$
    $M_i \leftarrow \mathbf{T}[h]$
  else
    $h \leftarrow H_1(M_i)$
    $\mathbf{T}[h] \leftarrow M_i$
    $h' \leftarrow H_1(C_{R,i})$
    $\mathbf{C}[h'] \leftarrow M_i$
$M_t \leftarrow \mathbf{C}[h^*]$
$M \leftarrow \textsc{Insert}(M_t, M_1||\cdots||M_{\ell-1})$
return M

Figure 5.1: Encryption and Decryption algorithms of AE-SCB$[E, \widetilde{E}, H_1, H_2]$. We let $\textsc{Insert}(M_t, M_1||\cdots||M_{\ell-1})$ represent the insertion of M_t at position t to $M_1||\cdots||M_{\ell-1}$ and increment of all the indices after t by 1.

such construction as an encoding scheme.

$$\textsc{Encode} : \{0,1\}^n \to \{0^{n-\sigma-\tau}||c||h \ : c \in \{0,1\}^\sigma, h \in \mathcal{H}\},$$

where $\mathcal{H} \subseteq \{0,1\}^\tau$ is the set of block hash. We can then exploit the structure for authenticity to construct a length-preserving AEAD scheme as illustrated in Figure 5.1.

For encryption, we extract the first repeated block from the plaintext and noting down its position. We denote the remaining plaintext as M_R. We proceed to encrypt M_R using SCB mode. Next, we create a special repetition

signal denoted as $R^* = 0^{n-\sigma-\tau}||[t]_\sigma||h$ from the extracted block, where t represents the position. Compared to the normal repetition signal, we instead calculate h as hash of the ciphertext of the first repeated block. Subsequently, we encipher R^* with the hash of C_R and the associated data as tweak. Finally, we produce the final output by concatenating the obtained ciphertexts.

When decrypting the ciphertext, we begin by extracting the first n bits from it. This extracted portion is then deciphered and we denote the result of the deciphering as R^*. We check if R^* is less than $2^{\sigma+\tau}$ and the last τ bits of R^* corresponds the hash of one of the ciphertext block. If not, it means that authentication fails and we output a decryption failure. We then use SCB to decrypt the remaining ciphertext, extract the actual block for R^*, then output the reordered plaintext with the position of the repeated block embedded in R^*.

We assume for block repetition for this instantiation with SCB mode. Nevertheless, we stress that as long as there is redundancy in plaintext, one can set the part of plaintext with redundancy as the segment M_L and encrypt with our generic composition using a stateful encryption scheme to achieve length-preservation in an AEAD scheme.

5.2 Security

We prove the security of the AEAD scheme under the assumption that the adversary's queries to the oracles Enc and Dec involve no more than 2^σ blocks. Note that this assumption provides a somewhat rough bound. To refine this bound further, we can consider that each message block queried to Enc does not repeat more than 2^σ times, then reduce to a collision-resistance adversary against H_1 to account for the possible appearance of repeated values among repetition signals when two different actual message blocks have the same hash value and in total appear more than 2^σ times, following a similar discussion in Section 3.2.

5.2.1 IND-CPA Security

Theorem 5.1 *For any* IND-CPA *adversary* $\mathcal{A}$ *that queries* β *n-bit blocks in total against* $\Psi = \mathsf{AE\text{-}SCB}[E, \widetilde{E}, H_1, H_2]$ *where* $\beta \leq 2^\sigma$*, there exist an* PRP *adversary* $\mathcal{B}_1$ *against* E*, a* TPRP *adversary* $\mathcal{B}_2$ *against* $\widetilde{E}$*, and a collision resistance adversary* $\mathcal{B}_3$ *against* H_2 *such that:*

$$\mathbf{Adv}^{\text{IND-CPA}}_{\Psi}(\mathcal{A}) \leq \mathbf{Adv}^{\text{PRP}}_{E}(\mathcal{B}_1) + \mathbf{Adv}^{\text{TPRP}}_{\widetilde{E}}(\mathcal{B}_2) + \mathbf{Adv}^{\text{CR}}_{H_2}(\mathcal{B}_3) + \frac{\beta^2}{2^n}.$$

Proof We consider four games G_0–G_3 as defined in Figure 5.2. Note that in G_0, we implement AE-SCB. In G_1, we replace the part of message encrypted

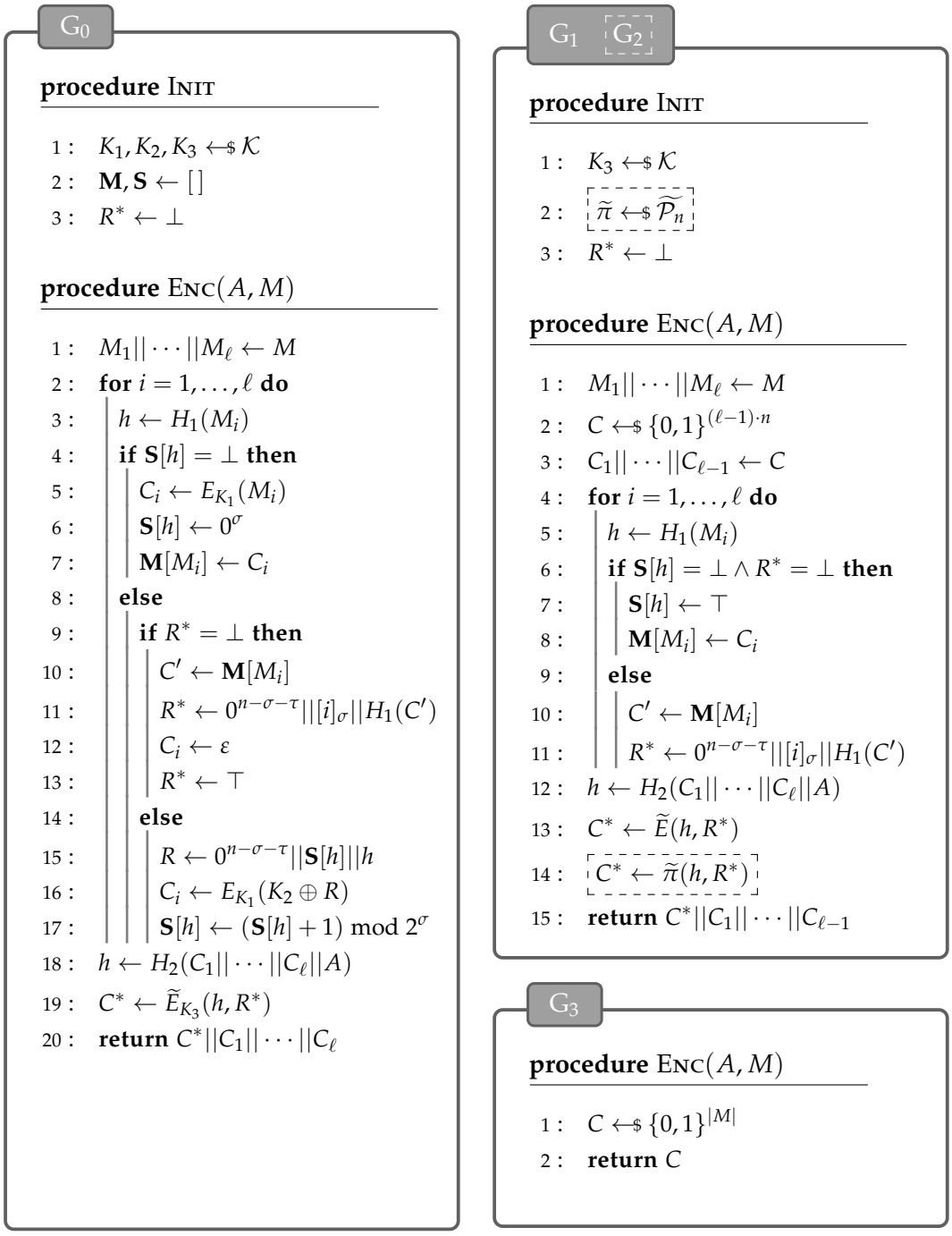

Figure 5.2: Game G_0 – G_3 for proof of Theorem 5.1. The dot-boxed code is exclusive to game G_2.

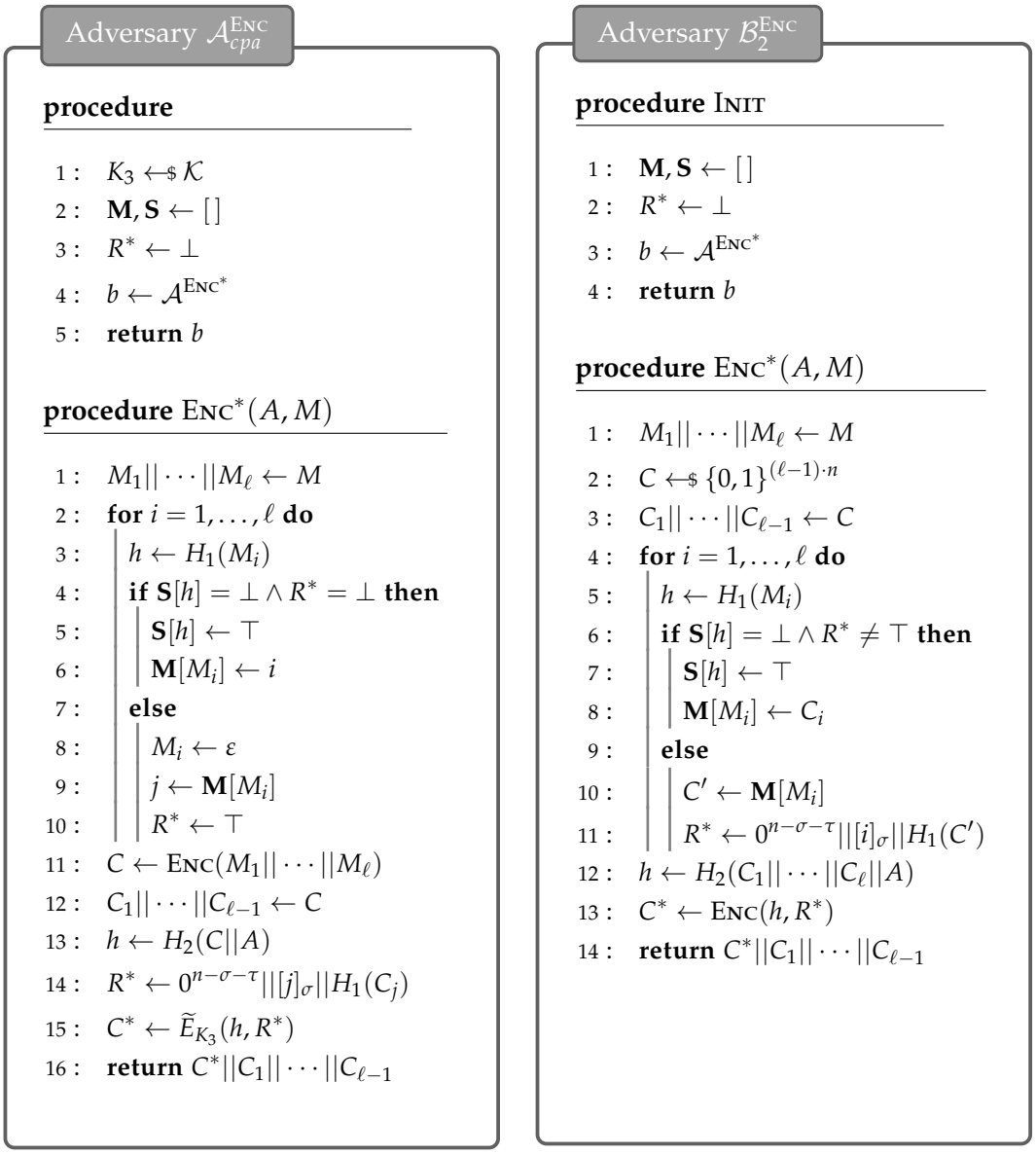

Figure 5.3: IND-CPA adversary $\mathcal{A}_{cpa}$ against SCB mode and TPRP adversary $\mathcal{B}_2$ against the tweak block cipher $\widetilde{E}$.

with SCB by a random bitstring of the same length. In G_2, we replace the tweak block cipher $\widetilde{E}$ by a tweakable random permutation $\widetilde{\pi}$. Finally, in G_3, we returns a random bitstring of the same length as the queried message. Thus we have that

$$\mathbf{Adv}_{\Psi}^{\text{IND-CPA}}(\mathcal{A}) = \sum_{i=0}^{2} \Pr[G_i(\mathcal{A})] - \Pr[G_{i+1}(\mathcal{A})]$$

Note that we can bound the difference between G_0 and G_1 by an IND-CPA adversary against SCB mode as in Figure 5.3. By [Ban22][Theorem 1], we have that

$$\Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] \leq \mathbf{Adv}_E^{\text{PRP}}(\mathcal{B}_1) + \frac{\beta^2}{2^n}$$

for a PRP adversary $\mathcal{B}_1$. Now we can bound the difference between G_1 and G_2 by constructing a TPRP adversary $\mathcal{B}_2$ as in Figure 5.3. Then we have that

$$\Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] = \mathbf{Adv}_{\widetilde{E}}^{\text{TPRP}}(\mathcal{B}_2).$$

Note that if each tweak queried to the tweakable random permutation $\widetilde{\pi}$ is distinct, then an adversary has 0 advantage in distinguishing it from a random bitstring. That is because, a fresh random permutation will be sampled for each tweak, yielding the same distribution as sampling a random bitstring. This happens if a collision is provoked in the hash function. Thus we can bound the difference between G_2 and G_3 by a collision-resistance adversary $\mathcal{B}_3$ against the hash function H_2. Thus we have that

$$\Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] \leq \mathbf{Adv}_{H_2}^{\text{CR}}(\mathcal{B}_3).$$

Finally, we have that

$$\begin{aligned}\mathbf{Adv}_{\Psi}^{\text{IND-CPA}}(\mathcal{A}) &= \sum_{i=0}^{2} \Pr[G_i(\mathcal{A})] - \Pr[G_{i+1}(\mathcal{A})] \\ &\leq \mathbf{Adv}_E^{\text{PRP}}(\mathcal{B}_1) + \mathbf{Adv}_{\widetilde{E}}^{\text{TPRP}}(\mathcal{B}_2) + \mathbf{Adv}_{H_2}^{\text{CR}}(\mathcal{B}_3) + \frac{\beta^2}{2^n}\end{aligned}$$

which concludes the proof.

5.2.2 INT-CTXT Security

Theorem 5.2 *For any* INT-CTXT *adversary $\mathcal{A}$ that queries β n-bit blocks in total against $\Psi = \mathsf{AE\text{-}SCB}[E, \widetilde{E}, H_1, H_2]$ where $\beta \leq 2^\sigma$, there exist an* TPRP *adversary $\mathcal{B}$ against $\widetilde{E}$ and a collision resistance adversary $\mathcal{B}_2$ against H_2 such that:*

$$\mathbf{Adv}_{\Psi}^{\text{INT-CTXT}}(\mathcal{A}) \leq \mathbf{Adv}_{\widetilde{E}}^{\text{TPRP}}(\mathcal{B}_1) + \mathbf{Adv}_{H_2}^{\text{CR}}(\mathcal{B}_2) + \frac{2^\sigma \beta^2}{2^n}.$$

Proof (sketch) We consider two games G_0 – G_1. In G_0, AE-SCB is executed. In G_1, the tweakable block cipher $\widetilde{E}$ is replaced with a tweakable random permutation $\widetilde{\pi}$. Thus we have that

$$\mathbf{Adv}_{\Psi}^{\text{INT-CTXT}}(\mathcal{A}) = \Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] + \Pr[G_1(\mathcal{A})]$$

Similarly, we have that

$$\Pr[G_0(\mathcal{A})] - \Pr[G_1(\mathcal{A})] = \mathbf{Adv}_{\widetilde{E}}^{\text{TPRP}}(\mathcal{B})_1$$

for a TPRP adversary $\mathcal{B}_1$.

We now bound the probability that $\mathcal{A}$ wins in G_1. In the first case, let (A, C) be the result of a previous encryption query. Suppose that $\mathcal{A}$ queries to Dec with an associated data A' that yields the same tweak as the previous encryption query, that is, $H_2(C||A) = H_2(C||A')$. Then $\mathcal{A}$ can reuse C' to create a valid forgery. We can then reduce this case to a collision-resistance adversary $\mathcal{D}$. In the second case, to be a valid repetition signal, R^* must begin with $n - \sigma - \tau$ leading zeros and the last τ bits corresponds to the hash of a ciphertext block. We assume that $\mathcal{A}$ queries in total β n-bit blocks. By Union Bound, this happens with probability at most $\frac{2^\sigma \beta^2}{2^n}$. Finally, we have that

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\tilde{E}}(\mathcal{B}_1) + \mathbf{Adv}^{\mathrm{CR}}_{H_2}(\mathcal{B}_2) + \frac{2^\sigma \beta^2}{2^n}$$

which concludes the proof. □

Chapter 6

Conclusion and Future Work

6.1 Possible Extension and Optimization

The AEAD scheme presented in this work necessitates perfect correctness. It is worth investigating the feasibility of increasing the correctness parameters without compromising security or at the cost of negligibly reduced security. Given that the security of the AE-SCB relies on the combination of a tweak and a message block, it may be possible to decrease the security parameters while maintaining IND-CPA security.

Moreover, a challenging question arises regarding the extension of the AE-SCB scheme to a length-preserving authenticated encryption scheme while eliminating the prerequisite we pose. A robust authenticated encryption (RAE) scheme, initially introduced in [HKR15], offers users the flexibility to choose a parameter $\lambda \geq 0$, allowing encryption of plaintexts of any length into ciphertexts with an expansion length of λ. Currently, the only scheme that achieves strict length preservation is AEZ as presented in [HKR15]. However, AEZ requires multiple rounds of AES and its implementation is overly complex. The same work [HKR15] shows that it is theoretically possible to construct a robust AE scheme from a tweakable block cipher following the Encoding-then-Encipher (EtE) paradigm. In our construction, we view the construction of a repetition signal in SCB as encoding scheme. Designing an encoding scheme that can be applied on the message of any structure, for example, without message repetition, may be the first step towards constructing a *real* length-preserving AEAD scheme based on SCB.

Additionally, further optimization of the SCB mode can be pursued by addressing the issue of state growth. The SCB mode and our modification employs various lookup tables to maintain the states during encryption and decryption. The current implementation of the SCB mode exhibits linear state growth. It is yet to be explored whether there exists an approach to optimize the growth of the state, or even maintaining a constant state.

6.2 Conclusion

SCB mode serves as a concrete example of length-preserving encryption, demonstrating the possibility of achieving semantic security in a LPE scheme while sacrificing some level of correctness. This prompts an investigation into the trade-off between correctness and security in length-preserving encryption schemes.

In our research, we examine how the repetition of message block might affect the security in length-preserving encryption. We also study how to provide better correctness by accounting for the counter in the decryption algorithm of SCB. The sliding window technique we proposed effectively increase the correctness of SCB mode in case blocks are reordered, enabling the selection of a larger security parameter, thereby improving overall security.

Our findings constitute a preliminary step towards achieving an improved balance between correctness and security in length-preserving encryption. Nevertheless, the question of designing a length-preserving encryption scheme with a better trade-off remains an outstanding issue. Several areas of potential improvement for SCB mode include the development of better counter validation schemes in case of block reordering.

Based on SCB mode, we propose a length-preserving AEAD scheme. Our design utilizes a tweakable block cipher under the assumption of block repetition. It should be noted that our scheme is applicable only under these specific assumptions. We leave it for future research to develop an AEAD scheme on the top of SCB mode that imposes little to no assumptions on the structure of the message, and achieves perfect length-preservation.

Bibliography

[Ban22] Fabio Banfi. SCB mode: Semantically secure length-preserving encryption. *IACR Transactions on Symmetric Cryptology*, 2022(4):1–23, 2022.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15, Santa Barbara, CA, USA, August 18–22, 1996. Springer, Heidelberg, Germany.

[BD99] Daniel Bleichenbacher and Anand Desai. A construction of a super-pseudorandom cipher. *Manuscript, February*, 1999.

[BDJR97] Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science*, pages 394–403, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545, Kyoto, Japan, December 3–7, 2000. Springer, Heidelberg, Germany.

[BR99] Mihir Bellare and Phillip Rogaway. On the construction of variable-input-length ciphers. In Lars R. Knudsen, editor, *Fast Software Encryption – FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 231–244, Rome, Italy, March 24–26, 1999. Springer, Heidelberg, Germany.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 317–330, Kyoto, Japan, December 3–7, 2000. Springer, Heidelberg, Germany.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.

[CYK04] Debra L. Cook, Moti Yung, and Angelos D. Keromytis. Elastic block ciphers. Cryptology ePrint Archive, Report 2004/128, 2004. https://eprint.iacr.org/2004/128.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th Annual ACM Symposium on Theory of Computing*, pages 365–377, San Francisco, CA, USA, May 5–7, 1982. ACM Press.

[Hal04] Shai Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004: 5th International Conference in Cryptology in India*, volume 3348 of *Lecture Notes in Computer Science*, pages 315–327, Chennai, India, December 20–22, 2004. Springer, Heidelberg, Germany.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

[HR04] Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In Tatsuaki Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304, San Francisco, CA, USA, February 23–27, 2004. Springer, Heidelberg, Germany.

[LRW11] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. *Journal of Cryptology*, 24(3):588–613, July 2011.

[MF07] David A. McGrew and Scott R. Fluhrer. The security of the extended codebook (xcb) mode of operation. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *SAC 2007: 14th Annual International Workshop on Selected Areas in Cryptography*, volume 4876 of *Lecture Notes in Computer Science*, pages 311–327, Ottawa, Canada, August 16–17, 2007. Springer, Heidelberg, Germany.

[RB18] E. Rescorla and R. Barnes. The JSON Web Encryption (JWE) Unencoded Payload Option. Request for comments, IETF, August 2018.

[RBBK01] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In Michael K. Reiter and Pierangela Samarati, editors, *ACM CCS 2001: 8th Conference on Computer and Communications Security*, pages 196–205, Philadelphia, PA, USA, November 5–8, 2001. ACM Press.

[Rog04] Phillip Rogaway. Nonce-based symmetric encryption. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 348–359, New Delhi, India, February 5–7, 2004. Springer, Heidelberg, Germany.

[Rog06] Phillip Rogaway. Formalizing human ignorance. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06: 1st International Conference on Cryptology in Vietnam*, volume 4341 of *Lecture Notes in Computer Science*, pages 211–228, Hanoi, Vietnam, September 25–28, 2006. Springer, Heidelberg, Germany.

[Ros] Mike Rosulek. The joy of cryptography. https://joyofcryptography.com.

[RWZ12] Phillip Rogaway, Mark Wooding, and Haibin Zhang. The security of ciphertext stealing. In Anne Canteaut, editor, *Fast Software Encryption – FSE 2012*, volume 7549 of *Lecture Notes in Computer Science*, pages 180–195, Washington, DC, USA, March 19–21, 2012. Springer, Heidelberg, Germany.

[ST13] Thomas Shrimpton and R. Seth Terashima. A modular framework for building variable-input-length tweakable ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 405–423, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.