From Single-Bit to Multi-Bit Public-Key Encryption via Non-Malleable Codes

Sandro Coretti[1], Ueli Maurer[1], Björn Tackmann[1], and Daniele Venturi[2]

[1] *ETH Zürich*
[2] *Sapienza University of Rome*

May 14, 2014

Abstract

One approach towards basing public-key encryption schemes on weak and credible assumptions is to build "stronger" or more general schemes generically from "weaker" or more restricted schemes. One particular line of work in this context, which has been initiated by Myers and Shelat (FOCS '09) and continued by Hohenberger, Lewko, and Waters (Eurocrypt '12), is to build a multi-bit chosen-ciphertext (CCA) secure public-key encryption scheme from a single-bit CCA-secure one. While their approaches achieve the desired goal, it is fair to say that the employed techniques are complicated and that the resulting ciphertext lengths are impractical.

We propose a completely different and surprisingly simple approach to solving this problem. While it is well-known that encrypting each bit of a plaintext string independently is insecure—the resulting scheme is *malleable*—we show that applying a suitable non-malleable code (Dziembowski *et al.*, ICS '10) to the plaintext and subsequently encrypting the resulting codeword bit-by-bit results in a secure scheme. Our result is the one of the first applications of non-malleable codes in a context other than memory tampering.

The original notion of non-malleability is, however, not sufficient. We therefore prove that (a simplified version of) the code of Dziembowski *et al.* is actually *continuously non-malleable* (Faust *et al.*, TCC '14). Then, we show that this notion is sufficient for our application. Since continuously non-malleable codes require to keep a single bit of (not necessarily secret) state in the decoding, the decryption of our scheme also has to keep this state. This slight generalization of the traditional formalization of public-key encryption schemes seems appropriate for applications. Compared to the previous approaches, our technique leads to conceptually simpler and more efficient schemes.

Contents

1 Introduction

1.1 Overview

A public-key encryption (PKE) scheme enables a sender A to send messages to a receiver B confidentially if B can send a single message, the public key, to A authentically. A encrypts a message with the public key and sends the ciphertext to B via a channel that could be authenticated or insecure, and B decrypts the received ciphertext using the private key. Following the seminal work of Diffie and Hellman [15], the first formal definition of public-key encryption has been provided by Goldwasser and Micali [25], and to date numerous instantiations of this concept have been proposed, e.g., [42, 18, 12, 22, 26, 28, 43, 41], for different security properties and based on various different computational assumptions.

One natural approach towards developing public-key encryption schemes based on weak and credible assumptions is to build "stronger" or more general schemes generically from "weaker" or less general ones. While the "holy grail"—generically building a chosen-ciphertext secure scheme based on any chosen-plaintext secure one—has so far remained out of reach, and despite negative results [24], various interesting positive results have been shown. For instance, Cramer *et al.* [11] build *bounded-query* chosen-ciphertext secure schemes from chosen-plaintext secure ones, Choi *et al.* [6] *non-malleable* schemes from chosen-plaintext secure ones, and Lin and Tessaro [30] show how the security of weakly chosen-ciphertext secure schemes can be amplified. A line of work started by Myers, Sergi, and shelat [39] and continued by Dachman-Soled [13] shows how to obtain chosen-ciphertext secure schemes from plaintext-aware ones. Most relevant for our work, however, are the results of Myers and Shelat [40] and Hohenberger, Lewko, and Waters [27], which generically build a multi-bit chosen-ciphertext secure scheme from a single-bit chosen-ciphertext secure one.

A naïve approach to solving this problem would be to encrypt each bit $m[i]$ of a plaintext $m = (m[1], \ldots, m[k])$ under an independent public key pk_i of the single-bit scheme. Unfortunately, this simple approach does not yield chosen-ciphertext security. The reason is that the above scheme is *malleable*: Given a ciphertext $e = (e_1, \ldots, e_k)$, where e_i is an encryption of $m[i]$, an attacker can generate a new ciphertext $e' \neq e$ that decrypts to a related message, for instance by copying the first ciphertext component e_1 and replacing the other components by fresh encryptions of, say, 0.

The idea underlying our approach is remarkably simple: As the insufficiency of the naïve scheme stems from its malleability, we first encode the message using a non-malleable[1] code (a concept introduced by Dziembowski *et al.* [17]) to protect its integrity, obtaining an n-bit codeword $c = (c[1], \ldots, c[n])$. Then, we encrypt each bit $c[i]$ of the codeword using public key pk_i as in the naïve protocol from above.

Unfortunately, non-malleable codes as introduced by [17] are not sufficient: Since they are only secure against a single tampering, the security of the resulting scheme would only hold with respect to a single decryption. *Continuously* non-malleable codes (Faust *et al.* [19]) allow us to extend this guarantee to an a priori unbounded number of decryptions. These codes, however, require us to keep one bit of state for the decryption: The code "self-destructs" once an attack has been detected, and, therefore, further decryptions must be prevented. This is a restriction that we prove to be unavoidable.

Overall, we obtain a scheme that achieves chosen-ciphertext security for an *a priori unbounded* number of decryptions (unlike, e.g., [11, 6]) and becomes dysfunctional only in the event of an explicit attack. This restriction is acceptable in the usual scenarios where the attacker can anyway violate the availability by preventing messages from being delivered.

1.2 Outline of the Paper

The above issue of building a multi-bit PKE scheme from a single-bit one and our approach based on non-malleable codes can be rephrased in the framework of constructive cryptography [32, 34]. This permits splitting the security proof of our scheme into two independent steps. For the first step, which includes a reduction from breaking the CCA-security of the 1-bit scheme, we can reuse a previous result [8]. The second step—the main technical contribution of this paper—is purely information-theoretic.

Constructive cryptography. Security statements for cryptographic schemes can be stated as *constructions* of a "stronger" or more useful desired resource from a "weaker" or more restricted assumed one. Two such construction steps can be composed, i.e., if a protocol π constructs a resource S from an assumed resource R, denoted by $R \xmapsto{\pi} S$, and, additionally, a protocol ψ assumes resource S and constructs a resource T, then the composition theorem of constructive cryptography states that the composed protocol, denoted $\psi \circ \pi$, constructs resource T from R. The resources considered in this work are different types of communication channels between two parties A and B; a channel is a resource that involves three entities: the sender, the receiver, and a (potential) attacker E.

[1]Roughly, a code is non-malleable w.r.t. a function class $\mathcal{F}$, if the message obtained by decoding a codeword modified via a function in $\mathcal{F}$ is either the original message or a completely unrelated value.

We use and extend the notation by [36], denoting different types of channels by different arrow symbols. A *confidential* channel (later denoted $-\!\diamond\!\!\twoheadrightarrow\!\bullet$) hides the messages sent by A from the attacker E but potentially allows her to inject *independent* messages; an *authenticated* channel (later denoted $\bullet\!\!-\!\diamond\!\!\twoheadrightarrow$) is dual to the confidential channel in that it potentially leaks the message to the attacker but prevents modifications and injections; an insecure channel (later denoted $-\,-\!\!\twoheadrightarrow$) protects neither the confidentiality nor the authenticity. In all cases, the double arrow head indicates that the channel can be used to transmit multiple messages. A single arrow head, instead, means that channels are single-use.

Warm-up: Dealing with the malleability of the one-time pad. The one-time pad allows to encrypt an n-bit message m using an n-bit shared key κ by computing the ciphertext $e = m \oplus \kappa$. If e is sent via an insecure channel, an attacker can replace it by a different ciphertext e', in which case the receiver will compute $m' = e' \oplus \kappa = m \oplus (e \oplus e')$. This can be seen, as described in [35], as constructing from an insecure channel and a shared secret n-bit key an "XOR-malleable" channel (denoted $-\!\oplus\!\!\rightarrow\!\bullet$), which is confidential but allows the attacker to specify a mask $\delta \in \{0,1\}^n$ ($= e \oplus e'$) to be XORed to the transmitted message.

Non-malleable codes can be used to deal with the XOR-malleability. To transmit a k-bit message m, we encode m with a (k,n)-bit non-malleable code, obtaining an n-bit codeword c, which we transmit via the XOR-malleable channel $-\!\oplus\!\!\rightarrow\!\bullet$. Since by XORing a mask δ to a codeword transmitted via $-\!\oplus\!\!\rightarrow\!\bullet$ the attacker can influence the value of each bit of the codeword only independently, a code C that is non-malleable w.r.t. the function class $\mathcal{F}_{\mathsf{bit}}$, which (in particular) allows to either "keep" or "flip" each bit of a codeword only individually, is sufficient. Indeed, the non-malleability of C implies that the decoded message will be either the original message or a completely unrelated value, which is the same guarantee as formulated by the single-message confidential channel (denoted $\longrightarrow\!\bullet$), and hence using C, one achieves the construction

$$-\!\oplus\!\!\rightarrow\!\bullet \qquad \Longmapsto \qquad \longrightarrow\!\bullet.$$

A more detailed treatment and a formalization of this example appears in Appendix A; suitable non-malleable codes are described in [17, 5].

Dealing with the malleability of multiple single-bit encryptions. Following [8], a PKE scheme is chosen-ciphertext secure if and only if it constructs a confidential channel $-\!\diamond\!\!\twoheadrightarrow\!\bullet$ from A to B from an authenticated channel $\longleftarrow\!\bullet$ from B to A and an insecure channel $-\,-\!\!\twoheadrightarrow$ from A to B [8]. Consequently, a single-bit public-key encryption scheme constructs a single-bit confidential channel, denoted by $\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$. By the composition theorem, n copies of a single-bit encryption scheme construct n instances of the channel $\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$, written $\big[\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}\big]^n$.

Thus, the remaining step is showing how to achieve the construction

$$\big[\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}\big]^n \qquad \Longmapsto \qquad \overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet} \tag{1}$$

for some $k > 1$. Then, by the composition theorem, plugging these two steps together yields a construction of a k-bit confidential channel from an authenticated channel and an insecure channel, and thus, using the result from [8] again, is a chosen-ciphertext secure PKE scheme.

To achieve construction (1), we use non-malleable codes. The fact that the channels are multiple-use leads to two important differences to the one-time-pad example above: First, the attacker can fabricate multiple codewords, which are then decoded. Second, these messages can be created by combining *any* of the bits in each channel. This results in a different class of tampering functions, called $\mathcal{F}_{\mathsf{copy}}$, against which the code needs to be secure.

We build a continuously non-malleable code w.r.t. $\mathcal{F}_{\mathsf{copy}}$; the code consists of a linear error-correcting secret sharing (LECSS) scheme and can be seen as a simplified version of the code in [17]. The security proof of the code proceeds in two steps: First, we prove that it is continuously non-malleable w.r.t. $\mathcal{F}_{\mathsf{copy}}$ against tampering with a single encoding. Then, we show that if a code is continuously non-malleable w.r.t. $\mathcal{F}_{\mathsf{copy}}$ against tampering with a single encoding, then it is also *adaptively* continuously non-malleable w.r.t. $\mathcal{F}_{\mathsf{copy}}$, i.e., against tampering with many encodings simultaneously.[2] These two steps are the technical heart of this work.

On the necessity of "self-destruct". The description of our main protocol above omitted one important detail. The code, to be continuously non-malleable, has to "self-destruct" in the event of a decoding error. For the application in the setting of public-key encryption, this means that the decryption algorithm also has to deny processing any further ciphertext once the code self-destructs, which requires storing a single bit of information. We formalize this as a resource FLAG allowing to store a single (publicly readable) bit. The necessity of self-destruct is not an artifact of our proof technique: We show that without self-destruct no code can be continuously non-malleable with respect to $\mathcal{F}_{\mathsf{copy}}$, which means in particular that no such code is sufficient for the constructive statement we aim for. This proof can be found in Section 5.

For practical applications, instead of registering an encryption public key at a certification authority (CA), the receiver can register a signature verification key and publish a new, signed encryption public key (e.g., on his web page) once the decryption self-destructs. This is different from bounded-query secure schemes, which can be used to process only an *a priori fixed* number of messages. In our scheme, the key only needs to be replaced in the event of an attack, and if no attack occurs, the number of possible decryptions is *a priori unbounded.* This methodology could even lead to stronger security statements in other related constructions.

Game-based security. The security of our scheme can also be captured by a game-based notion. This notion, called self-destruct chosen-ciphertext security (SD-CCA), is a CCA variant that allows the scheme to self-destruct in case it detects an invalid ciphertext. The standard CCA game can easily be extended to include the self-destruct mode of the decryption: The decryption oracle keeps answering decryption queries as long as no invalid ciphertext (i.e., a ciphertext upon which the decryption algorithm outputs an error symbol) is received; after such an event occurs, no further decryption query is answered.

The guarantees of SD-CCA are perhaps best understood if compared to the q-bounded CCA notion by [6]. While q-CCA allows an *a priori determined* number q of decryption queries, SD-CCA allows an *arbitrary* number of *valid* decryption queries and one invalid query. From a practical viewpoint, an attacker can efficiently violate the availability with a scheme of either notion. However, as long as no invalid ciphertexts are received, an SD-CCA scheme can run indefinitely, whereas a q-CCA scheme has to necessarily stop after q decryptions. A formal definition of SD-CCA and further discussion can be found in Appendix C.

One can show that SD-CCA security can in fact be achieved from CPA security only [9], by generalizing the approach of Choi *et al.* [6]. The resulting scheme, however, is considerably less

[2]We remark that all our definitions are based on non-malleability and not on *strong* non-malleability [17].

efficient than the one we provide in this paper.

1.3 More Details on Related Work

The work of Hohenberger *et al.* [27]—building on the work of Myers and Shelat [40]—describes a multi-bit CCA-secure encrytion scheme from a single-bit CCA-secure one, a CPA-secure one, and a 1-query-bounded CCA-secure one. Their scheme is rather sophisticated and has a somewhat circular structure, requiring a complex security proof. The public key is of the form $\mathsf{pk} = (\mathsf{pk}_{in}, \mathsf{pk}_A, \mathsf{pk}_B)$, where the "inner" public key pk_{in} is the public key of a DCCA secure PKE scheme, and the "outer" public keys pk_A and pk_B are, respectively, the public key of a 1-bounded CCA and a CPA secure PKE scheme. To encrypt a k-bit message m one first encrypts a tuple (r_A, r_B, m), using the "inner" public key, obtaining a ciphertext e_{in}, where r_A and r_B are thought as being the randomness for the "outer" encryption scheme. Next, one has to encrypt e_{in} under the "outer" public key pk_A (resp. pk_B) using randomness r_A (resp. r_B) and thus obtaining a ciphertext e_A (resp. e_B). The output ciphertext is $e = (e_A, e_B)$.

To use the above scheme, we have to instantiate the DCCA, 1-bounded CCA and CPA components. As argued in [27], all schemes can be instantiated using a single-bit IND-CCA PKE scheme yielding a fully black-box construction of a multi-bit IND-CCA PKE scheme from a single-bit IND-CCA PKE scheme. Let us denote with γ_p (resp., γ_e) the bit-length of the public key (resp., the ciphertext) for the single-bit IND-CCA PKE scheme. When we refer to the construction of [11] for the 1-bounded CCA component, we get a public key of size roughly $(3 + 16s)\gamma_p$ for the public key and $(k + 2s) \cdot 4s \cdot \gamma_e^2$ for the ciphertext, for security parameter s.[3]

In contrast, our scheme instantiated with the information-theoretic LECSS scheme of [17] has a ciphertext of length $\approx 5k\gamma_e$ and a public key of length $k\gamma_p$. Note that the length of the public key depends on the length of the message, as we need independent public keys for each encrypted bit (whereas the DCCA scheme can use always the same public key). However, we observe that when k is not too large, e.g. in case the PKE scheme is used as a key encapsulation mechanism, we would have $k \approx s$ yielding public keys of comparable size. On the negative side, recall that our construction needs one bit of (potentially public) storage to self-destruct in case an invalid ciphertext is processed, which is not required in [27].

As shown in [8], the constructive security statement for public-key encryption corresponds to RCCA-security, a notion proposed by Canetti *et al.* [3]. Hence, our scheme actually achieves self-destruct RCCA-security. We remark, however, that if one is interested in CCA-security, this can be achieved generically from RCCA-security [3]. Moreover, we conjecture that when instantiated with a *strong* adaptively continuously non-malleable code w.r.t. $\mathcal{F}_{\mathsf{copy}}$, our approach actually yields a scheme that is CCA-secure.

Non-malleable codes. Beyond the constructions of [17, 5, 19], non-malleable codes exists against block-wise tampering [7], against split-state tampering—both information-theoretic [16, 1] and computational [31]—and in a setting where the computational complexity of the tampering functions is somewhat limited [4, 21]. We stress that the typical application of non-malleable codes is to protect cryptographic schemes against memory tampering (see, e.g., [17, 14]). A further application of non-malleable codes has been shown by Agrawal et al. [2] (in concurrent and independent

[3]For simplicity, we assumed that the random strings r_A, r_B are computed by stretching the seed (of length s) of a pseudo-random generator.

work). They show that one can obtain a non-malleable multi-bit commitment scheme from a non-malleable single-bit commitment scheme by encoding the value with a (specific) non-malleable code and then committing to the codeword bits. Despite the similarity of the approaches, the techniques applied in their paper differ heavily from ours. The class of tampering functions the code has to protect against is different, and we additionally need continuous non-malleability to handle multiple decryption queries (this is not required for the commitment case).

2 Preliminaries

2.1 Random Systems

Resources and converters. We use the concepts and terminology of abstract [34] and constructive cryptography [32]. The *resources* we consider are different types of communication channels, which are systems with three interfaces labeled by A, B, and E. A *converter* is a two-interface system which is directed in that it has an *inside* and an *outside* interface. Converters model protocol engines that are used by the parties, and using a protocol is modeled by connecting the party's interface of the resource to the inside interface of the converter (which hides those two interfaces) and using the outside interface of the converter instead. We generally use upper-case, bold-face letters (e.g., $\mathbf{R}$, $\mathbf{S}$) or channel symbols (e.g., $\bullet\!\!-\!\!\diamond\!\!\twoheadrightarrow$) to denote resources or single-interface systems and lower-case Greek letters (e.g., α, β) or sans-serif fonts (e.g., enc, dec) for converters. We denote by Φ the set of all resources and by Σ the set of all converters.

For $I \in \{A, B, E\}$, a resource $\mathbf{R} \in \Phi$, and a converter $\alpha \in \Sigma$, the expression $\alpha^I \mathbf{R}$ denotes the composite system obtained by connecting the inside interface of α to interface I of $\mathbf{R}$; the outside interface of α becomes the I-interface of the composite system. The system $\alpha^I \mathbf{R}$ is again a resource (cf. Figure 5 on page 13). For two resources $\mathbf{R}$ and $\mathbf{S}$, $[\mathbf{R}, \mathbf{S}]$ denotes the parallel composition of $\mathbf{R}$ and $\mathbf{S}$. For each $I \in \{A, B, E\}$, the I-interfaces of $\mathbf{R}$ and $\mathbf{S}$ are merged and become the *sub-interfaces* of the I-interface of $[\mathbf{R}, \mathbf{S}]$.

Distinguishers. A *distinguisher* $\mathbf{D}$ connects to all interfaces of a resource $\mathbf{U}$ and outputs a single bit at the end of its interaction with $\mathbf{U}$. The expression $\mathbf{DU}$ defines a binary random variable, and the *distinguishing advantage of a distinguisher* $\mathbf{D}$ *on two systems* $\mathbf{U}$ *and* $\mathbf{V}$ is defined as

$$\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) := |\mathsf{P}[\mathbf{DU} = 1] - \mathsf{P}[\mathbf{DV} = 1]|.$$

The distinguishing advantage measures how much the output distribution of $\mathbf{D}$ differs when it is connected to either $\mathbf{U}$ or $\mathbf{V}$. Note that the distinguishing advantage is a pseudo-metric.[4]

Reductions. When relating two distinguishing problems, it is convenient to use a special type of system $\mathbf{C}$ that translates one setting into the other. Formally, $\mathbf{C}$ is a converter that has an *inside* and an *outside* interface. When it is connected to a system $\mathbf{S}$, which is denoted by $\mathbf{CS}$, the inside interface of $\mathbf{C}$ connects to the (merged) interface(s) of $\mathbf{S}$ and the outside interface of $\mathbf{C}$ is the interface of the composed system. $\mathbf{C}$ is called a *reduction system* (or simply *reduction*).

To reduce distinguishing two systems $\mathbf{S}, \mathbf{T}$ to distinguishing two systems $\mathbf{U}, \mathbf{V}$, one exhibits a reduction $\mathbf{C}$ such that $\mathbf{CS} \equiv \mathbf{U}$ and $\mathbf{CT} \equiv \mathbf{V}$. Then, for all distinguishers $\mathbf{D}$, we have $\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) =$

[4] That is, it is symmetric, satisfies the triangle inequality, and $\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{R}) = 0$ for all $\mathbf{D}$ and $\mathbf{R}$.

$\Delta^{\mathbf{D}}(\mathbf{CS}, \mathbf{CT}) = \Delta^{\mathbf{DC}}(\mathbf{S}, \mathbf{T})$. The last equality follows from the fact that $\mathbf{C}$ can also be thought of as being part of the distinguisher, which follows from composition-order independence [34].

Discrete systems. The behavior of systems can be formalized by random systems as in [38, 33]: A random system $\mathbf{S}$ is a sequence $(\mathsf{p}^{\mathbf{S}}_{Y^i|X^i})_{i\geq 1}$, where $\mathsf{p}^{\mathbf{S}}_{Y^i|X^i}(y^i, x^i)$ is the probability of observing the outputs $y^i = (y_1, \dots, y_i)$ given the inputs $x^i = (x_1, \dots, x_i)$. If for two systems $\mathbf{R}$ and $\mathbf{S}$,

$$\mathsf{p}^{\mathbf{R}}_{Y^i|X^i} = \mathsf{p}^{\mathbf{S}}_{Y^i|X^i}$$

for all i and for all parameters where both are defined, they are called *equivalent*, denoted by $\mathbf{R} \equiv \mathbf{S}$. In that case, $\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) = 0$ for all distinguishers $\mathbf{D}$.

A system $\mathbf{S}$ can be extended by a so-called *monotone binary output* (or *MBO*) $\mathcal{B}$, which is an additional one-bit output $B_1, B_2, \dots$ with the property that $B_i = 1$ implies $B_{i+1} = 1$ for all i.[5] The enhanced system is denoted by $\hat{\mathbf{S}}$, and its behavior is described by the sequence $(\mathsf{p}^{\hat{\mathbf{S}}}_{Y^i,B_i|X^i})_{i\geq 1}$. If for two systems $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ with MBOs,

$$\mathsf{p}^{\hat{\mathbf{R}}}_{Y^i,B_i=0|X^i} = \mathsf{p}^{\hat{\mathbf{S}}}_{Y^i,B_i=0|X^i}$$

for all i, they are called *game equivalent*, which is denoted by $\hat{\mathbf{R}} \overset{g}{\equiv} \hat{\mathbf{S}}$. In such a case, $\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) \leq \Gamma^{\mathbf{D}}(\hat{\mathbf{R}}) = \Gamma^{\mathbf{D}}(\hat{\mathbf{S}})$, where $\Gamma^{\mathbf{D}}(\hat{\mathbf{R}})$ denotes the probability that $\mathbf{D}$ provokes the MBO. For more details and a proof of this fact, consult [33].

The notion of construction. We formalize the security of protocols via the notion of *construction*, introduced in [32]:

Definition 1. Let Φ and Σ be as above, and let ε_1 and ε_2 be two functions mapping each distinguisher $\mathbf{D}$ to a real number in $[0, 1]$. A protocol $\pi = (\pi_1, \pi_2) \in \Sigma^2$ *constructs resource* $\mathbf{S} \in \Phi$ *from resource* $\mathbf{R} \in \Phi$ *with distance* $(\varepsilon_1, \varepsilon_2)$ *and with respect the simulator* $\sigma \in \Sigma$, denoted $\mathbf{R} \overset{\pi,\sigma,(\varepsilon_1,\varepsilon_2)}{\Longmapsto} \mathbf{S}$, if for all distinguishers $\mathbf{D}$,

$$\begin{cases} \Delta^{\mathbf{D}}({\pi_1}^A {\pi_2}^B \perp^E \mathbf{R}, \perp^E \mathbf{S}) & \leq \varepsilon_1(\mathbf{D}) & \qquad \text{(availability)} \\ \Delta^{\mathbf{D}}({\pi_1}^A {\pi_2}^B \mathbf{R}, \sigma^E \mathbf{S}) & \leq \varepsilon_2(\mathbf{D}) & \qquad \text{(security).} \end{cases}$$

The availability condition captures that a protocol must correctly implement the functionality of the constructed resource in the absence of the attacker. The security condition models the requirement that everything the attacker can achieve in the setting with the assumed resource and the protocol, she can also accomplish in the setting with the constructed resource (using the simulator to translate the behavior).

2.2 Channel Resources

From the perspective of constructive cryptography, the purpose of a public-key encryption scheme is to construct a confidential channel from non-confidential channels. Here, a channel is a resource that involves a sender, a receiver, and—to model channels with different levels of security—an attacker. The main type of channels relevant to this work are defined below.

[5] In other words, once the MBO is 1, it cannot return to 0.

Insecure multiple-use channel. We specify the insecure channel with respect to a set $\{A, B, E\}$ of interfaces, and parametrize the channel by a message space $\{M\} \subseteq \{0,1\}^*$. The insecure channel $-\twoheadrightarrow$ transmits multiple messages and corresponds to, for instance, communication via the Internet. If no attacker is present (i.e., in case $\perp^E -\twoheadrightarrow$), then all messages are transmitted from A to B faithfully. Otherwise (for $-\twoheadrightarrow$), the communication can be controlled via the E-interface, i.e., the attacker learns all messages input at the A-interface and chooses the messages to be output at the B-interface. The channel is described in more detail in Figure 1.

The channel $\perp^E -\twoheadrightarrow$

(Repeatedly) Upon input a message m at the A-interface, output m at the B-interface.

The channel $-\twoheadrightarrow$ with accessible E-interface

- (Repeatedly) Upon input a message m at the A-interface, output m at the E-interface.
- (Repeatedly) Upon input a message m at the E-interface, output m at the B-interface.

Figure 1: Insecure, multiple-use communication channel from A to B, denoted $-\twoheadrightarrow$.

Authenticated (unreliable) single-use channel. The (single-use) authenticated channel $\bullet\!\!\longrightarrow$, described in Figure 2, is also formulated in the $\{A, B, E\}$-setting and allows the sender A to transmit a single message to the receiver B authentically. That means, while the attacker (at the E-interface) can still read the transmitted message, the only influence allowed is delaying the message (arbitrarily, i.e., there is no guarantee that the message will ever be delivered). The channel guarantees that *if* a message is delivered to B, *then* this message was input by A before. There are different constructions that result in the channel $\longleftarrow\!\!\bullet$, based on, for instance, MACs or signature schemes.

The channel $\perp^E \bullet\!\!\longrightarrow$

Upon input a message m at the A-interface, output m at the B-interface.

The channel $\bullet\!\!\longrightarrow$ with accessible E-interface

- Upon input a message m at the A-interface, output m at the E-interface.
- Accept at the E-interface a bit $d \in \{0,1\}$, on input $d = 0$, output m at the B-interface.

Figure 2: Authenticated, single-use communication channel from A to B, denoted $\bullet\!\!\longrightarrow$.

Confidential multiple-use channel. The k-bit confidential channel is also specified with interfaces in $\{A, B, E\}$. The channel $\overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$ transmits multiple messages. If no attacker is present (i.e., in case $\perp^E \overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$), then all messages are transmitted from A to B faithfully. Otherwise (for $\overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$), on input a message $m \in \{0,1\}^k$ at the A-interface, the message m is stored in a buffer $\mathcal{B}$. The attacker can then choose messages from the buffer $\mathcal{B}$ (by using an index, since it might not know the messages) to be delivered at the B-interface, or inject "fresh" messages from $\{0,1\}^k$ which are then also output at the B-interface. The channel is described in more detail in Figure 3.

The channel $\perp^E \overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$

- On input $m \in \{0,1\}^k$ at interface A, output m at interface B.

The channel $\overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$ with accessible E-interface

- On the i^{th} input $m_i \in \{0,1\}^k$ at interface A, output i at the E-interface and store (i, m_i) in buffer $\mathcal{B}$.
- On input $i \in \mathbb{N}$ at interface E, if $(i, m) \in \mathcal{B}$ for some $m \in \{0,1\}^k$, output m at interface B.
- On input $m \in \{0,1\}^k$ at interface E, output m at the interface B.

Figure 3: Confidential, multiple-use k-bit channel from A to B; denoted $\overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$.

2.3 Public-Key Encryption Schemes

A public-key encryption (PKE) scheme with message space $\mathcal{M} \subseteq \{0,1\}^*$ and ciphertext space $\mathcal{E}$ is defined as three algorithms $\Pi = (K, E, D)$, where the key-generation algorithm K outputs a key pair $(\mathsf{pk}, \mathsf{sk})$, the (probabilistic) encryption algorithm E takes a message $m \in \mathcal{M}$ and a public key pk and outputs a ciphertext $e \leftarrow E_{\mathsf{pk}}(m)$, and the decryption algorithm takes a ciphertext $e \in \mathcal{E}$ and a secret key sk and outputs a plaintext $m \leftarrow D_{\mathsf{sk}}(e)$. The output of the decryption algorithm can be the special symbol $\diamond$, indicating an invalid ciphertext. A PKE scheme is correct if $m = D_{\mathsf{sk}}(E_{\mathsf{pk}}(m))$ (with probability 1 over the randomness in the encryption algorithm) for all messages m and all key pairs $(\mathsf{pk}, \mathsf{sk})$ generated by K.

Chosen-ciphertext security. The standard bit-guessing game used to define security against chosen-ciphertext attacks (CCA) is phrased as a distinguishing problem between two game systems $\mathbf{G}_0^{\mathsf{cca}}$ and $\mathbf{G}_1^{\mathsf{cca}}$ (cf. Section 2.1), defined as follows: For a PKE scheme Π, both initially run the key-generation algorithm to obtain $(\mathsf{pk}, \mathsf{sk})$ and output pk. Upon (the first) query (chall, m), $\mathbf{G}_0^{\mathsf{cca}}$ outputs an encryption $e \leftarrow E_{\mathsf{pk}}(m)$ of m and $\mathbf{G}_1^{\mathsf{cca}}$ an encryption $e \leftarrow E_{\mathsf{pk}}(\bar{m})$, called the *challenge*, of a randomly chosen message $\bar{m}$ of length $|m|$. Both systems answer decryption queries (dec, e') by returning $m' \leftarrow D_{\mathsf{sk}}(e')$ at any time unless e' equals the challenge e (if defined), in which case the answer is test.

2.4 Continuously Non-Malleable Codes

Non-malleable codes, introduced in [17], are coding schemes that protect the encoded messages against certain classes of adversarially chosen modifications, in the sense that the decoding will result either in the original message or in an unrelated value.

Definition 2 (Coding scheme)**.** A (k, n)-coding scheme $(\mathrm{Enc}, \mathrm{Dec})$ consists of a randomized encoding function $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ and a deterministic decoding function $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{\diamond\}$ such that $\mathrm{Dec}(\mathrm{Enc}(x)) = x$ (with probability 1 over the randomness of the encoding function) for each $x \in \{0,1\}^k$. The special symbol $\diamond$ indicates an invalid codeword.

In the original definition, the adversary is allowed to modify the codeword via a function of the specified class $\mathcal{F}$ only once. Continuous non-malleability, introduced in [19], extends this guarantee to the case where the adversary is allowed to perform multiple such modifications for a fixed target codeword. The notion of *adaptive* continuous non-malleability considered here is an extension of

System $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$

init
- $\beta \leftarrow 0$
- $i \leftarrow 0$

on (encode, x)
- $i \leftarrow i + 1$
- $c^{(i)} \leftarrow\!\!\$\ \mathrm{Enc}(x)$

on (tamper, f) *with* $f \in \mathcal{F}^{(i)}$
- **if** $\beta = 1$
  - **output** $\diamond$
- **else**
  - $c' \leftarrow f(c^{(1)}, \ldots, c^{(i)})$
  - $x' \leftarrow \mathrm{Dec}(c')$
  - **if** $x' = \diamond$
    - $\beta \leftarrow 1$
  - **output** x'

System $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$

init
- $\beta \leftarrow 0$
- $i \leftarrow 0$

on (encode, x)
- $i \leftarrow i + 1$
- $x^{(i)} \leftarrow\!\!\$\ x$

on (tamper, f) *with* $f \in \mathcal{F}^{(i)}$
- **if** $\beta = 1$
  - **output** $\diamond$
- **else**
  - $x' \leftarrow\!\!\$\ \tau(i, f)$
  - **if** $x' = \diamond$
    - $\beta \leftarrow 1$
  - **if** $x' = (\mathsf{same}, j)$
    - $x' \leftarrow x^{(j)}$
  - **output** x'

Figure 4: Systems $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ defining adaptive continuous non-malleability of $(\mathrm{Enc}, \mathrm{Dec})$.

the one in [19] in that the adversary is allowed to adaptively specify messages and the functions may depend on multiple codewords. That is, the class $\mathcal{F}$ is actually a sequence $(\mathcal{F}^{(i)})_{i\geq 1}$ of function families with $\mathcal{F}^{(i)} \subseteq \{f \mid f : (\{0,1\}^n)^i \to \{0,1\}^n\}$, and after encoding i messages, the adversary chooses functions from $\mathcal{F}^{(i)}$. A similar adaptive notion has been already considered for continuous strong non-malleability in the split-state model [20].

Formally, adaptive continuous non-malleability w.r.t. $\mathcal{F}$ is defined by comparing the two random systems $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ defined in Figure 4. Both systems expect to interact with a distinguisher $\mathbf{D}$, whose objective is to tell the two systems apart. System $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ produces a random encoding $c^{(i)}$ of each message $x^{(i)}$ specified by $\mathbf{D}$ and allows $\mathbf{D}$ to repeatedly issue tampering functions $f \in \mathcal{F}^{(i)}$. For each such query, $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ computes the modified codeword $c' = f(c^{(1)}, \ldots, c^{(i)})$ and outputs $\mathrm{Dec}(c')$. Whenever $\mathrm{Dec}(c') = \diamond$, the system enters a "self-destruct" mode, in which all further queries are replied with $\diamond$.

The second random system, $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$, features a simulator τ, which is allowed to keep state. The simulator repeatedly takes a tampering function and outputs either a message x', (same, i), or $\diamond$, where (same, i) is used by τ to indicate that (it believes that) the tampering function has copied the i^{th} encoding. System $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ outputs whatever τ outputs, except that (same, i) is replaced by the i^{th} message $x^{(i)}$ specified by $\mathbf{D}$. Moreover, in case of $\diamond$, $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ "self-destructs".

For $\ell, q \in \mathbb{N}$, $\mathbf{S}^{\mathsf{real}}_{\mathcal{F},\ell,q}$ is the system that behaves as $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ except that only the first ℓ encode-queries and the first q tamper-queries are handled (and similarly for $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau,\ell,q}$ and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$). Note that by setting $\ell = 1$, one recovers continuous non-malleability,[6] and by additionally setting $q = 1$ the original definition of non-malleability.

Definition 3 (Adaptive continuous non-malleability)**.** Consider a sequence $\mathcal{F} = (\mathcal{F}^{(i)})_{i\geq 1}$ of function families $\mathcal{F}^{(i)} \subseteq \{f \mid f : (\{0,1\}^n)^i \to \{0,1\}^n\}$ and let $\ell, q \in \mathbb{N}$. A coding scheme $(\mathrm{Enc}, \mathrm{Dec})$ is *adaptively continuously* $(\mathcal{F}, \varepsilon, \ell, q)$*-non-malleable* (or simply $(\mathcal{F}, \varepsilon, \ell, q)$*-non-malleable*) if there exists a simulator τ such that $\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F},\ell,q}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau,\ell,q}) \leq \varepsilon$ for all distinguishers $\mathbf{D}$.

3 From Single-Bit to Multi-Bit Channels

In this section we show how to combine a single-bit chosen-ciphertext secure (CCA) PKE scheme with an adaptively continuously non-malleable code to achieve a multi-bit chosen-ciphertext secure

[6] Being based on strong non-malleability, the notion of [19] is actually stronger than ours.

scheme (see Section 2.3 for a definition of CCA security). All channel resources that appear in this section are formally defined in Section 2.2.

Let $k > 1$. As shown in [8], in constructive terms obtaining a k-bit CCA-secure scheme means achieving the construction

$$[\longleftarrow\!\!\bullet, -\twoheadrightarrow] \quad \Longmapsto \quad \overset{k\text{-bit}}{-\!\!\diamond\!\!\twoheadrightarrow\!\!\bullet}, \tag{2}$$

where the (single-use) authenticated channel $\longleftarrow\!\!\bullet$ can be used for transmitting the public key and the insecure channel $-\twoheadrightarrow$ for sending ciphertexts.[7] Our approach to achieve construction (2) can be modularly divided into two main constructive steps, as explained in the following subsections.

3.1 Single-bit Channels from Single-bit PKE

Given a 1-bit CCA-secure PKE scheme Π, one can build a protocol $\mathsf{pke} = (\mathsf{encrypt}, \mathsf{decrypt})$ that achieves the construction

$$[\longleftarrow\!\!\bullet, -\twoheadrightarrow] \quad \overset{\mathsf{pke}}{\Longmapsto} \quad [\overset{1\text{-bit}}{-\!\!\diamond\!\!\twoheadrightarrow\!\!\bullet}]^n \tag{3}$$

for any $n \in \mathbb{N}$. More precisely, following [8, Theorem 2], a 1-bit CCA-secure PKE scheme can be seen as a protocol $\mathsf{pke}_1 = (\mathsf{encrypt}_1, \mathsf{decrypt}_1)$ that achieves the construction

$$[\longleftarrow\!\!\bullet, -\twoheadrightarrow] \quad \overset{(\mathsf{encrypt}_1,\mathsf{decrypt}_1)}{\Longmapsto} \quad \overset{1\text{-bit}}{-\!\!\diamond\!\!\twoheadrightarrow\!\!\bullet}, \tag{4}$$

where, in a nutshell, $\mathsf{decrypt}_1$ is responsible for key generation as well as decryption and $\mathsf{encrypt}_1$ for encryption.

Using the composition theorem (see Appendix B), one obtains

$$[\longleftarrow\!\!\bullet, -\twoheadrightarrow]^n \quad \overset{(\mathsf{encrypt}'_1,\mathsf{decrypt}'_1)}{\Longmapsto} \quad [\overset{1\text{-bit}}{-\!\!\diamond\!\!\twoheadrightarrow\!\!\bullet}]^n, \tag{5}$$

where $\mathsf{encrypt}'_1$ and $\mathsf{decrypt}'_1$ are the n-fold parallel composition of $\mathsf{encrypt}_1$ and $\mathsf{decrypt}_1$, respectively. A slight modification pke''_1 of protocol $\mathsf{pke}'_1 = (\mathsf{encrypt}'_1, \mathsf{decrypt}'_1)$ allows to use $[\longleftarrow\!\!\bullet, [-\twoheadrightarrow]^n]$ as the assumed resource. Essentially, all public keys are concatenated and sent via a single $\longleftarrow\!\!\bullet$. A proof of security is straight-forward. Moreover, there is a simple protocol s that constructs $[-\twoheadrightarrow]^n$ from $-\twoheadrightarrow$. Essentially, it appends i to a message when it is to be sent over the i^{th} channel. Thus, using the composition theorem again, the concatenation $\mathsf{pke} := \mathsf{pke}''_1 \circ \mathsf{s}$ achieves construction (3).

3.2 Tying the Channels Together

To achieve construction (2), it remains to construct a k-bit confidential channel from the n single-bit confidential channels. This is achieved by having the sender encode the message with a (k, n)-non-malleable code and sending the resulting codeword over the 1-bit channels, while the receiver decodes all n-bit strings received on these channels.

Due to the self-destruct property of continuously non-malleable codes, the receiver must stop decoding once an invalid codeword has been received. This requires keeping a single bit of state, which we formalize by the additional resource FLAG: Initially, it internally sets $\beta \leftarrow 0$. When read is input at interface B, FLAG outputs β at B. When B inputs set, FLAG sets $\beta \leftarrow 1$ and outputs β at E.

[7] According to [8], a scheme that achieves (2) is in fact only guaranteed to be RCCA-secure [3], a notion sufficient for most applications. Note, however, that full CCA security can be achieved generically from RCCA security [3].

Summarizing, the goal is to develop a protocol $\mathsf{nmc} = (\mathsf{enc}, \mathsf{dec})$ that achieves

$$[\mathsf{FLAG}, [\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n] \quad \overset{\mathsf{nmc}}{\Longmapsto} \quad \overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}\,. \tag{6}$$

Note that the need for FLAG is not an artifact of our proof technique: In Section 5 we show that $\overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$ cannot be constructed from $[\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n$ by a stateless protocol.

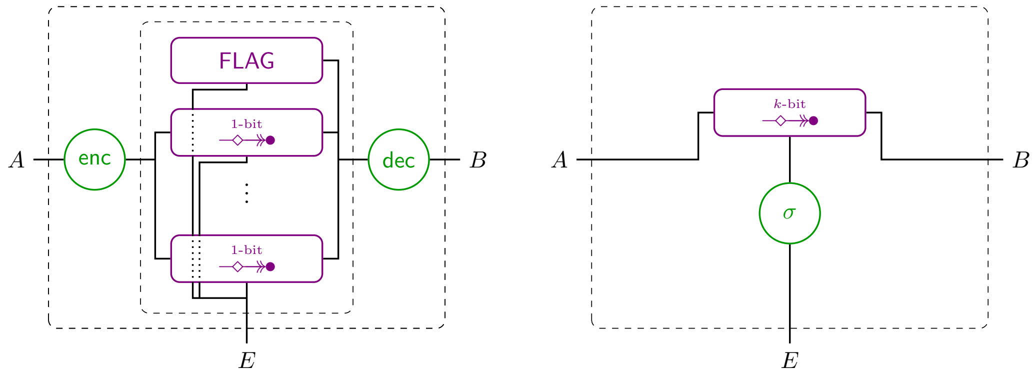

Figure 5: Left: The assumed resource $[\mathsf{FLAG}, [\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n]$ with protocol converters enc and dec attached to interfaces A and B, denoted $\mathsf{enc}^A\mathsf{dec}^B[\mathsf{FLAG}, [\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n]$. Right: The constructed resource $\overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$ with simulator σ attached to the E-interface, denoted $\sigma^E \overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$. In particular, σ must simulate the E-interfaces of FLAG and $[\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n$. The protocol is secure if the two systems are indistinguishable.

Let $(\mathrm{Enc}, \mathrm{Dec})$ be a (k, n)-coding scheme and consider the following protocol $\mathsf{nmc} = (\mathsf{enc}, \mathsf{dec})$ (cf. Figure 4): Converter enc encodes every message $m \in \{0,1\}^k$ input at its outside interface with fresh randomness, resulting in an encoding $c = (c[1], \ldots, c[n]) \leftarrow \mathrm{Enc}(m)$. Then, for $i = 1, \ldots, n$, it outputs bit $c[i]$ to the i^{th} channel at the inside interface. Converter dec, whenever it receives an n-bit string $c' = (c'[1], \ldots, c'[n])$ (where the i^{th} bit $c'[i]$ was sent via the i^{th} channel), it outputs read at the inside sub-interface corresponding to resource FLAG. If the value subsequently received at the inside interface is $\beta = 1$, dec outputs $\diamond$ at its outside interface. Otherwise, it computes $m' \leftarrow \mathrm{Dec}(c')$ and outputs m' at the outside interface. If $m' = \diamond$, it also outputs set at the inside interface.

The required non-malleability. Since each of the channels $\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$ allows the attacker to either forward one of the bits in the channel or to inject a fresh bit which is either 0 or 1, this results in the following class $\mathcal{F}_{\mathsf{copy}}$ of tampering functions against which the code needs to be secure: Let $\mathcal{F}_{\mathsf{copy}} := (\mathcal{F}^{(i)}_{\mathsf{copy}})_{i \geq 1}$ where $\mathcal{F}^{(i)}_{\mathsf{copy}} \subseteq \{f \mid f : (\{0,1\}^n)^i \to \{0,1\}^n\}$ and each function $f \in \mathcal{F}^{(i)}_{\mathsf{copy}}$ is characterized by a vector $\chi(f) = (f_1, \ldots, f_n)$ where $f_i \in \{\mathsf{zero}, \mathsf{one}, \mathsf{copy}_1, \ldots, \mathsf{copy}_i\}$, with the meaning that f takes as input i codewords $(c^{(1)}, \ldots, c^{(i)})$ and outputs a codeword $c' = (c'[1], \ldots, c'[n])$ in which each bit is either set to 0 (zero), set to 1 (one), or copied from the *corresponding* bit in a codeword $c^{(j)}$ (copy_j).

Theorem 1 (see below) implies that nmc achieves construction (6) if $(\mathrm{Enc}, \mathrm{Dec})$ is adaptively continuously non-malleable w.r.t. $\mathcal{F}_{\mathsf{copy}}$. We construct such a code in Section 4.

Theorem 1. *For any* $\ell, q \in \mathbb{N}$*, if* $(\mathrm{Enc}, \mathrm{Dec})$ *is* $(\mathcal{F}_{\mathsf{copy}}, \varepsilon, \ell, q)$*-continuously non-malleable, there exists a simulator* σ *such that*

$$[\mathsf{FLAG}, [\overset{\text{1-bit},\ell,q}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n] \overset{(\mathsf{enc},\mathsf{dec}),\sigma,(0,\varepsilon)}{\Longmapsto} \overset{k\text{-bit},\ell,q}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}, \tag{7}$$

where the additional superscripts ℓ, q *on a channel mean that it only processes the first* ℓ *queries at the* A*-interface and only the first* q *queries at the* E*-interface.*

Proof. The availability condition (7) holds by the correctness of the code.

Let $\mathcal{F} := \mathcal{F}_{\mathsf{copy}}$, $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} := \mathbf{S}^{\mathsf{real}}_{\mathcal{F},\ell,q}$, and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau} := \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau,\ell,q}$ where τ is the simulator guaranteed to exist by Definition 3. Consider the following simulator σ (based on τ), which simulates the E-sub-interfaces of FLAG and the 1-bit confidential channels at its outside interface: Initially it sets $\beta \leftarrow 0$. When i is received at the inside interface, it outputs i at each outside sub-interface corresponding to a 1-bit confidential channel. Whenever σ receives one instruction to either deliver of inject one bit[8] at each outside sub-interface corresponding to one of the confidential channels, it assemble these to a function f with $\chi(f) = (f_1, \ldots, f_n)$ as follows: For all $j = 1, \ldots, n$,

$$f_j := \begin{cases} \mathsf{zero} & \text{if the instruction on the } j^{\text{th}} \text{ sub-interface is } 0, \\ \mathsf{one} & \text{if the instruction on the } j^{\text{th}} \text{ sub-interface is } 1, \\ \mathsf{copy}_v & \text{if the instruction on the } j^{\text{th}} \text{ sub-interface is } v. \end{cases}$$

Then, σ invokes τ to obtain $x' \leftarrow\!\!\$\, \tau(i, f)$, where i is the number of instructions i received at the inside interface so far. If $\beta = 1$, σ outputs $\diamond$ at the inside interface. Otherwise, if $x' = \diamond$, σ sets $\beta \leftarrow 1$ and outputs it at the outside sub-interface corresponding to FLAG. If $x' = (\mathsf{same}, j)$, σ outputs j at the inside interface. Otherwise, it outputs x'.

Consider the following reduction $\mathbf{C}$, which provides interfaces A, B, and E on the outside and expects to connect to either $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ or $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ on the inside. When a message m is input at the A-interface, $\mathbf{C}$ outputs (encode, m) on the inside. Similarly to σ, it repeatedly collects instructions input at the E-sub-interfaces and uses them to form a tamper function f, which it outputs on the inside as (tamper, f). Then, it outputs the answer x' received on the inside at the B-interface. Additionally, if $x' = \diamond$, $\mathbf{C}$ outputs 1 at the E-sub-interface corresponding to FLAG and subsequently only outputs $\diamond$ at interface B.

One observes that

$$\mathbf{C}\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} \equiv \mathsf{enc}^A \mathsf{dec}^B [\mathsf{FLAG}, [\overset{\text{1-bit},\ell,q}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n] \qquad \text{and} \qquad \mathbf{C}\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau} \equiv \sigma^E \overset{k\text{-bit},\ell,q}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}.$$

Thus, for all distinguishers $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathsf{enc}^A \mathsf{dec}^B [\mathsf{FLAG}, [\overset{\text{1-bit},\ell,q}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n], \sigma^E \overset{k\text{-bit},\ell,q}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}) = \Delta^{\mathbf{D}}(\mathbf{C}\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{C}\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) = \Delta^{\mathbf{DC}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) \leq \varepsilon.$$

□

[8] For simplicity, assume that no deliver instruction for some v greater than the number of instructions i received at the inside interface so far is input.

4 Continuous Non-Malleability against Bit-Wise Tampering

In this section, we describe a code based on a *linear error-correcting secret-sharing code (LECSS)* and prove it adaptively continuously non-malleable w.r.t. $\mathcal{F}_{\mathsf{copy}}$. As we argue below, it is actually sufficient to prove that the code is continuously non-malleable for a single encoding, which is formalized by the following (generic) theorem. The proof appears in Section 4.2.

Let $\ell \in \mathbb{N}$. Consider the sequence $\mathcal{F}_{\mathsf{copy}} = (\mathcal{F}_{\mathsf{copy}}^{(i)})_{i\in[\ell]}$ (as introduced in Section 3). The transition from $(\mathcal{F}_{\mathsf{copy}}^{(1)}, \cdot, 1, \cdot)$- to $(\mathcal{F}_{\mathsf{copy}}, \cdot, \ell, \cdot)$-non-malleability is achieved generically for an arbitrary (k, n)-coding scheme $(\mathrm{Enc}, \mathrm{Dec})$. In particular, we prove the following theorem:

Theorem 2. *If* $(\mathrm{Enc}, \mathrm{Dec})$ *is continuously* $(\mathcal{F}_{\mathsf{copy}}, \varepsilon, 1, q)$*-non-malleable, it is also continuously* $(\mathcal{F}_{\mathsf{copy}}, 2\ell\varepsilon + \frac{q\ell}{2^k}, \ell, q)$*-non-malleable, for all* $\ell \in \mathbb{N}$.

The use of a LECSS is inspired by the work of [17], who proposed a (single-shot) non-malleable code against bit-wise tampering based on a LECSS and one other code. As we do not need to provide non-malleability against "bit-flips", using only the LECSS is sufficient for our purposes. The following definition is taken from [17]:

Definition 4 (LECSS code)**.** A (k, n)-coding scheme $(\mathrm{Enc}, \mathrm{Dec})$ is a (d, t)-linear error-correcting secret-sharing (LECSS) code if the following properties hold:

- Linearity: For all $c \in \{0, 1\}^n$ such that $\mathrm{Dec}(c) \neq \perp$, all $\delta \in \{0, 1\}^n$, we have

$$\mathrm{Dec}(c + \delta) = \begin{cases} \perp & \text{if } \mathrm{Dec}(\delta) = \perp \\ \mathrm{Dec}(c) + \mathrm{Dec}(\delta) & \text{otherwise.} \end{cases}$$

- Distance d: For all non-zero $c' \in \{0, 1\}^n$ with Hamming weight $w_H(c') < d$, we have $\mathrm{Dec}(c') = \perp$.
- Secrecy t: For any fixed $x \in \{0, 1\}^k$, the bits of $\mathrm{Enc}(x)$ are individually uniform and t-wise independent (over the randomness in the encoding).

It turns out that a LECSS code is already continuously non-malleable with respect to $\mathcal{F}_{\mathsf{copy}}$:

Theorem 3. *Assume that* $(\mathrm{Enc}, \mathrm{Dec})$ *is a* (t, d)*-LECSS* (k, n)*-code for* $d > n/4$ *and* $d > t$*. Then* $(\mathrm{Enc}, \mathrm{Dec})$ *is* $(\mathcal{F}_{\mathsf{copy}}, \varepsilon, 1, q)$*-continuously non-malleable for all* $q \in \mathbb{N}$ *and*

$$\varepsilon = 3 \cdot 2^{-t} + \left(\frac{t}{n(d/n - 1/4)^2}\right)^{t/2}.$$

4.1 Proof of Theorem 3

For brevity, we write $\mathcal{F}_{\mathsf{set}}$ for $\mathcal{F}_{\mathsf{copy}}^{(1)}$ below, with the idea that the tampering functions in $\mathcal{F}_{\mathsf{copy}}^{(1)}$ only allow to keep a bit or to set it to 0 or to 1. More formally, a function $f \in \mathcal{F}_{\mathsf{set}}$ can be characterized by a vector $\chi(f) = (f_1, \dots, f_n)$ where $f_i \in \{\mathsf{zero}, \mathsf{one}, \mathsf{keep}\}$, with the meaning that f takes as input a codeword c and outputs a codeword $c' = (c'[1], \dots, c'[n])$ in which each bit is either set to 0 (zero), set to 1 (one), or left unchanged (keep).

For the proof of Theorem 3, fix $q \in \mathbb{N}$ and some distinguisher $\mathbf{D}$. For the remainder of this section, let $\mathcal{F} := \mathcal{F}_{\mathsf{set}}$, $\mathbf{S}_{\mathcal{F}}^{\mathsf{real}} := \mathbf{S}_{\mathcal{F},1,q}^{\mathsf{real}}$ and $\mathbf{S}_{\mathcal{F},\tau}^{\mathsf{simu}} := \mathbf{S}_{\mathcal{F},\tau,1,q}^{\mathsf{simu}}$ (for a simulator τ to be determined).

For a tamper query $f \in \mathcal{F}$ with $\chi(f) = (f_1, \ldots, f_n)$ issued by $\mathbf{D}$, let $A(f) := \{i \mid f_i \in \{\mathsf{zero}, \mathsf{one}\}\}$, $B(f) := \{i \mid f_i \in \{\mathsf{keep}\}\}$, and $a(f) := |A(f)|$. Moreover, let $\mathsf{val}(\mathsf{zero}) := 0$ and $\mathsf{val}(\mathsf{one}) := 1$. Queries f with $0 \le a(f) \le t$, $t < a(f) < n - t$, and $n - t \le a(f) \le n$ are called *low queries*, *middle queries*, and *high queries*, respectively.

Handling Middle Queries. Consider the hybrid system $\mathbf{H}$ that proceeds as $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$, except that as soon as $\mathbf{D}$ specifies a middle query f, $\mathbf{H}$ self-destructs, i.e., answers f and all subsequent queries with $\diamond$.

Lemma 4. $\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{H}) \le \frac{1}{2^t} + \left(\frac{t}{n(d/n-1/4)^2}\right)^{t/2}$.

Proof. Define a *successful* middle query to be a middle query that does not decode to $\diamond$. On both systems $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ and $\mathbf{H}$, one can define an MBO $\mathcal{B}$ (cf. Section 2.1) that is provoked if and only if the *first* middle query is successful.

Clearly, $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ and $\mathbf{H}$ behave identically until MBO $\mathcal{B}$ is provoked, thus $\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}} \overset{g}{\equiv} \hat{\mathbf{H}}$, and

$$\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{H}) \le \Gamma^{\mathbf{D}}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}).$$

Towards bounding $\Gamma^{\mathbf{D}}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}})$, note first that adaptivity does not help in provoking $\mathcal{B}$: For any distinguisher $\mathbf{D}$, there exists a *non-adaptive* distinguisher $\mathbf{D}'$ with

$$\Gamma^{\mathbf{D}}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}) \le \Gamma^{\mathbf{D}'}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}). \tag{8}$$

$\mathbf{D}'$ proceeds as follows: First, it (internally) interacts with $\mathbf{D}$ only. Initially, it stores the message x output by $\mathbf{D}$ internally. Whenever $\mathbf{D}$ outputs a low query, $\mathbf{D}'$ answers with x. Whenever $\mathbf{D}$ outputs a high query $f = (f_1, \ldots, f_n)$, $\mathbf{D}'$ checks whether there exists a codeword c^* that agrees with f in positions i where $f_i \in \{\mathsf{zero}, \mathsf{one}\}$. If it exists, it answers with $\mathrm{Dec}(c^*)$, otherwise with $\diamond$. As soon as $\mathbf{D}$ specifies a middle query, $\mathbf{D}'$ stops its interaction with $\mathbf{D}$ and sends x and all the queries to $\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}$.

To prove (8), fix all randomness in experiment $\mathbf{D}'\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$, i.e., the coins of $\mathbf{D}$ (inside $\mathbf{D}'$) and the randomness of the encoding (inside $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$). Suppose $\mathbf{D}$ would provoke $\mathcal{B}$ in the direct interaction with $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$. In that case all the answers by $\mathbf{D}'$ are equal to the answers by $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$. This is due to the fact that the distance of the LECSS is $d > t$; a successful low query must therefore result in the original message x and a successful high query in $\mathrm{Dec}(c^*)$. Thus, whenever $\mathbf{D}$ provokes $\mathcal{B}$, $\mathbf{D}'$ provokes it as well.

It remains to analyze the success probability of non-adaptive distinguishers $\mathbf{D}'$. Fix the coins of $\mathbf{D}'$; this determines the tamper queries. Suppose there is at least one middle case, as otherwise $\mathcal{B}$ is trivially not provoked. The middle case's success probability can be analyzed as in [17], which leads to $\Gamma^{\mathbf{D}'}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}) \le \frac{1}{2^t} + \left(\frac{t}{n(d/n-1/4)^2}\right)^{t/2}$ (recall that the MBO cannot be provoked after an unsuccessful first middle query). $\square$

Simulator. The final step of the proof consists of exhibiting a simulator τ such that $\Delta^{\mathbf{D}}(\mathbf{H}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau})$ is small. The indistinguishability proof is facilitated by defining two hardly distinguishable systems $\mathbf{B}$ and $\mathbf{B}'$ and a wrapper system $\mathbf{W}$ such that $\mathbf{W}\mathbf{B} \equiv \mathbf{H}$ and $\mathbf{W}\mathbf{B}' \equiv \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$.

System $\mathbf{B}$ works as follows: Initially, it takes a value $x \in \{0,1\}^k$, computes an encoding $(c[1], \ldots, c[n]) \leftarrow\!\!\$\ \mathrm{Enc}(x)$ of it, and outputs λ (where the symbol λ indicates an empty output).

Then, it repeatedly accepts guesses $g_i = (j, b)$, where (j, b) is a guess b for $c[j]$. If a guess g_i is correct, $\mathbf{B}$ returns $a_i = 1$. Otherwise, it outputs $a_i = \diamond$ and self-destructs (i.e., all future answers are $\diamond$). The system $\mathbf{B}'$ behaves as $\mathbf{B}$ except that the initial input x is ignored and the $c[1], \ldots, c[n]$ are chosen uniformly at random and independently.

The behavior of $\mathbf{B}$ (and similarly the one of $\mathbf{B}'$) is described by a sequence $(\mathsf{p}^{\mathbf{B}}_{A^i|G^i})_{i\geq 0}$ of conditional probability distributions, where $\mathsf{p}^{\mathbf{B}}_{A^i|G^i}(a^i, g^i)$ is the probability of observing the outputs $a^i = (\lambda, a_1, \ldots, a_i)$ given the inputs $g^i = (x, g_1, \ldots, g_i)$. For simplicity, assume below that g^i is such that no position is guessed twice (a generalization is straight-forward) and that a^i is of the form $\{\lambda\}\{1\}^*\{\diamond\}^*$ (as otherwise it has probability 0 anyway).

For system $\mathbf{B}$, all i, and any g^i, $\mathsf{p}^{\mathbf{B}}_{A^i|G^i}(a^i, g^i) = 2^{-(s+1)}$ if a^i has $s < \min(i, t)$ leading 1's; this follows from the t-wise independence of the bits of $\mathrm{Enc}(x)$. All remaining output vectors a^i, i.e., those with at least $\min(i, t)$ preceding 1's, share a probability mass of $2^{-\min(i,t)}$, in a way that depends on the code in use and on x. (It is easily verified that this yields a valid probability distribution.) The behavior of $\mathbf{B}'$ is obvious given the above (simply replace "t" by "n" in the above description).

Lemma 5. $\Delta^{\mathbf{D}}(\mathbf{B}, \mathbf{B}') \leq 2^{-(t-1)}$.

Proof. On both systems $\mathbf{B}$ and $\mathbf{B}'$, one can define an MBO $\mathcal{B}$ that is zero as long as *less* than t positions have been guessed correctly. In the following, $\hat{\mathbf{B}}$ and $\hat{\mathbf{B}}'$ denote $\mathbf{B}$ and $\mathbf{B}'$ with the MBO, respectively.

Analogously to the above, the behavior of $\hat{\mathbf{B}}$ (and similarly the one of $\hat{\mathbf{B}}'$) is described by a sequence $(\mathsf{p}^{\hat{\mathbf{B}}}_{A^i,B_i=0|G^i})_{i\geq 0}$ of conditional probability distributions, where $\mathsf{p}^{\hat{\mathbf{B}}}_{A^i,B_i=0|G^i}(a^i, g^i)$ is the probability of observing the outputs $a^i = (\lambda, a_1, \ldots, a_i)$ and $b_0 = b_1 = \ldots = b_i = 0$ given the inputs $g^i = (x, g_1, \ldots, g_i)$. One observes that due to the t-wise independence of $\mathrm{Enc}(x)$'s bits, for $i < t$,

$$\mathsf{p}^{\hat{\mathbf{B}}}_{A^i,B_i=0|G^i}(a^i, g^i) = \mathsf{p}^{\hat{\mathbf{B}}'}_{A^i,B_i=0|G^i}(a^i, g^i) = \begin{cases} 2^{-(s+1)} & \text{if } a^i \text{ has } s < i \text{ leading 1's,} \\ 2^{-i} & \text{if } a^i \text{ has } i \text{ leading 1's, and} \\ 0 & \text{otherwise,} \end{cases}$$

and for $i \geq t$,

$$\mathsf{p}^{\hat{\mathbf{B}}}_{A^i,B_i=0|G^i}(a^i, g^i) = \mathsf{p}^{\hat{\mathbf{B}}'}_{A^i,B_i=0|G^i}(a^i, g^i) = \begin{cases} 2^{-(s+1)} & \text{if } a^i \text{ has } s < t \text{ leading 1's,} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, $\hat{\mathbf{B}} \overset{g}{\equiv} \hat{\mathbf{B}}'$ and $\Delta^{\mathbf{D}}(\mathbf{B}, \mathbf{B}') \leq \Gamma^{\mathbf{D}}(\hat{\mathbf{B}}')$. Observe that by an argument similar to the one above, adaptivity does not help in provoking the MBO of $\hat{\mathbf{B}}'$. Thus, $\Gamma^{\mathbf{D}}(\hat{\mathbf{B}}') \leq 2^{-(t-1)}$, since an optimal non-adaptive strategy simply tries to guess distinct positions. $\square$

Recall that the purpose of the wrapper system $\mathbf{W}$ is to emulate $\mathbf{H}$ using $\mathbf{B}$. The key point is to note that low queries f can be answered knowing only the positions $A(f)$ of $\mathrm{Enc}(x)$, high queries knowing only the positions in $B(f)$, and middle queries can always be rejected. A full description of $\mathbf{W}$ can be found in Figure 6. It has an outside interface `o` and an inside interface `i`; at the latter interface, $\mathbf{W}$ expects to be connected to either $\mathbf{B}$ or $\mathbf{B}'$. For notational convenience, let $\mathsf{val}(\mathsf{zero}) := 0$ and $\mathsf{val}(\mathsf{one}) := 1$.

System $\mathbf{W}$

```
init
    ∀i ∈ [n] : c[i] ← λ

on first (encode, x) at o
    output x at i

on (tamper, f) with 0 ≤ a(f) ≤ t at o
    for i where f_i ∈ A(f)
        g ← val(f_i)
        if c[i] = λ
            output (i, g) at i
            get a ∈ {◇, 1} at i
            if a = ◇
                self-destruct
            c[i] ← g
        else
            if c[i] ≠ g
                self-destruct
    output x at out

on (tamper, f) with t < a(f) < n − t at o
    self-destruct

on (tamper, f) with n − t ≤ a(f) ≤ n at o
    for i where f_i ∈ A(f)
        c'[i] ← val(f_i)
    if ∃codeword c* : ∀i ∈ A(f) : c'[i] = c*[i]
        for i where f_i ∈ B(f)
            g ← c*[i]
            if c[i] = λ
                output (i, g) at i
                get a ∈ {◇, 1} at i
                if a = ◇
                    self-destruct
                c[i] ← g
            else
                if c[i] ≠ g
                    self-destruct
    else
        self-destruct
    output Dec(c*) at out
```

Figure 6: The wrapper system $\mathbf{W}$. The command **self-destruct** causes $\mathbf{W}$ to output $\diamond$ at o and to answer all future queries by $\diamond$.

Lemma 6. $\mathbf{WB} \equiv \mathbf{H}$.

Proof. Since the distance of the LECSS is $d > t$, the following holds: A low query results in same if all injected positions match the corresponding bits of the encoding, and in $\diamond$ otherwise. Similarly, for a high query, there can be at most one codeword that matches the injected positions. If such a codeword c^* exists, the outcome is $\mathrm{Dec}(c^*)$ if the bits in the keep-positions match c^*, and otherwise $\diamond$. By inspection, it can be seen that $\mathbf{W}$ acts accordingly. $\square$

Consider now the system $\mathbf{WB}'$. Due to the nature of $\mathbf{B}'$, the behavior of $\mathbf{WB}'$ is independent of the value x that is initially encoded. This allows to easily design a simulator τ as required by Definition 3. A full description of τ can be found in Figure 7.

Lemma 7. *The simulator τ of Figure 7 satisfies* $\mathbf{WB}' \equiv \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$.

Proof. Consider the systems $\mathbf{WB}'$ and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$. Both internally choose uniform and independent bits $c[1], \ldots, c[n]$. System $\mathbf{WB}'$ answers low queries with the value x initially encoded if all injected positions match the corresponding random bits and with $\diamond$ otherwise. Simulator τ returns same in the former case, which $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ replaces by x, and $\diamond$ in the latter case.

Simulator τ

```
init
  ∀i ∈ [n] : c[i] ←$ {0,1}

on (tamper, f) with 0 ≤ a(f) ≤ t
  if ∀i ∈ A(f) : val(f_i) = c[i]
    return same
  else
    return ◇

on (tamper, f) with t < a(f) < n − t
  return ◇

on (tamper, f) with n − t ≤ a(f) ≤ n
  for i where f_i ∈ A(f)
    c'[i] ← val(f_i)
  for i where f_i ∈ B(f)
    c'[i] ← c[i]
  c' ← (c'[1] ··· c'[n])
  return Dec(c')
```

Figure 7: The simulator τ.

Observe that the answer by $\mathbf{WB}'$ to a high query f always matches $\mathrm{Dec}(c'[1], \ldots, c'[n])$, where for $i \in A(f)$, $c'[i] = \mathsf{val}(f_i)$, and for $i \in B(f)$, $c'[i] = c[i]$: If no codeword c^* matching the injected positions exists, then $\mathrm{Dec}(c'[1], \ldots, c'[n]) = \diamond$, which is also what $\mathbf{WB}'$ outputs. If such c^* exists and $c^*[i] = c[i]$ for all $i \in B(f)$, the output of $\mathbf{WB}'$ is $\mathrm{Dec}(c'[1], \ldots, c'[n])$. If there exists an $i \in B(f)$ with $c^*[i] \neq c[i]$, $\mathbf{WB}'$ outputs $\diamond$, and in this case $\mathrm{Dec}(c'[1], \ldots, c'[n]) = \diamond$ since the distance of the LECSS is $d > t$. $\square$

The proof of Theorem 3 now follows from a simple triangle inequality.

Proof (of Theorem 3). From Lemmas 4, 5, 6, and 7, one obtains that for all distinguishers $\mathbf{D}$,

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) &\leq \Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{H}) + \underbrace{\Delta^{\mathbf{D}}(\mathbf{H}, \mathbf{WB})}_{=0} + \underbrace{\Delta^{\mathbf{D}}(\mathbf{WB}, \mathbf{WB}')}_{=\Delta^{\mathbf{DW}}(\mathbf{B},\mathbf{B}')} + \underbrace{\Delta^{\mathbf{D}}(\mathbf{WB}', \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau})}_{=0} \\ &\leq 2^{-t} + \left(\frac{t}{n(d/n - 1/4)^2}\right)^{t/2} + 2^{-(t-1)} \leq 3 \cdot 2^{-t} + \left(\frac{t}{n(d/n - 1/4)^2}\right)^{t/2}.\end{aligned}$$

$\square$

4.2 Proof of Theorem 2

Theorem 2. *If* $(\mathrm{Enc}, \mathrm{Dec})$ *is continuously* $(\mathcal{F}_{\mathsf{copy}}, \varepsilon, 1, q)$*-non-malleable, it is also continuously* $(\mathcal{F}_{\mathsf{copy}}, 2\ell\varepsilon + \frac{q\ell}{2^k}, \ell, q)$*-non-malleable, for all* $\ell \in \mathbb{N}$.

Left-or-right non-malleability. The proof of Theorem 2, which uses a hybrid argument, is facilitated by introducing a left-or-right (LOR) variant of non-malleability. The two definitions are equivalent, as shown by Lemmas 8 and 9

System $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},b}$

```
init
  β ← 0
  i ← 0

on (encode, x_0, x_1)
  i ← i + 1
  x_0^(i) ←$ x_0
  x_1^(i) ←$ x_1
  c^(i) ←$ Enc(x_b)

on (tamper, f) with f ∈ F^(i)
  if β = 1
    output ◇
  else
    c' ← f(c^(1), ..., c^(i))
    x' ←$ Dec(f)
    if x' = ◇
      β ← 1
    if ∃j : x' ∈ {x_0^(j), x_1^(j)}
      x' ← (same, j)
    output x'
```

Figure 8: Systems $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}$ and $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1}$ defining lor-non-malleability of $(\mathrm{Enc}, \mathrm{Dec})$.

below. In the LOR variant,[9] the encode-oracle takes as input pairs of messages and encodes either always the first or always the second message. The goal of the attacker is to find out which is the case. Formally, LOR-non-malleability is defined using the two random systems $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}$ and $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1}$, shown in Figure 8.[10]

When processing a tamper query, if there are multiple indices j for which (same, j) could be output, $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},b}$ outputs the largest such j. As before, for $b \in \{0,1\}$ and $\ell, q \in \mathbb{N}$, $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},b,\ell,q}$ is the system that behaves as $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},b}$ except that only the first ℓ encode-queries and the first q tamper-queries are handled.

Definition 5 (Adaptive continuous left-or-right non-malleability)**.** Let $\mathcal{F} = (\mathcal{F}^{(i)})_{i\geq 1}$ be a sequence of function families $\mathcal{F}^{(i)} \subseteq \{f \mid f : (\{0,1\}^n)^i \to \{0,1\}^n\}$ and let $\ell, q \in \mathbb{N}$. A coding scheme (Enc, Dec) is *adaptively continuously* $(\mathcal{F}, \varepsilon, \ell, q)$*-LOR-non-malleable* (or simply $(\mathcal{F}, \varepsilon, \ell, q)$-*LOR-non-malleable*) if there exists a simulator τ such that $\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0,\ell,q}, \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1,\ell,q}) \leq \varepsilon$ for all distinguishers $\mathbf{D}$.

Lemma 8. *If* (Enc, Dec) *is* $(\mathcal{F}, \varepsilon, \ell, q)$*-non-malleable, it is also* $(\mathcal{F}, 2\varepsilon, \ell, q)$*-LOR-non-malleable.*

Proof. Fix ℓ, q, and a simulator τ, and let $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} := \mathbf{S}^{\mathsf{real}}_{\mathcal{F},\ell,q}$, $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau} := \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau,\ell,q}$, $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0} := \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0,\ell,q}$, and $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1} := \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1,\ell,q}$. For $b \in \{0,1\}$, consider the following reduction $\mathbf{C}_b$: Upon the i^{th} query $(\mathsf{encode}, x_0, x_1)$ at the outside interface, it stores $x_0^{(i)} := x_0$ and $x_1^{(i)} := x_1$ internally and outputs (encode, x_b) at the inside interface. Upon a query (tamper, f) at the outside interface, $\mathbf{C}_b$ outputs (tamper, f) at the inside interface and subsequently receives a value x' at the inside interface. If there exist indices i' such that $x' \in \{x_0^{(i')}, x_1^{(i')}\}$, $\mathbf{C}_b$ outputs (same, i') for the largest such index at the outside interface. Otherwise, it outputs x'.

One observers that

$$\mathbf{C}_0\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} \equiv \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0} \qquad \text{and} \qquad \mathbf{C}_1\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} \equiv \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1} \qquad \text{and} \qquad \mathbf{C}_0\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau} \equiv \mathbf{C}_1\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau},$$

where the third equivalence follows from the fact that the observable behavior of $\mathbf{C}_b\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ is independent of the messages $\mathbf{C}_b$ outputs to $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$. Hence, for all attackers $\mathbf{A}$,

$$\begin{aligned}
\Delta^{\mathbf{A}}(\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}, \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1}) &= \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{C}_1\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}) \\
&\leq \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{C}_0\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) + \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}, \mathbf{C}_1\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) + \Delta^{\mathbf{A}}(\mathbf{C}_1\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}, \mathbf{C}_1\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}) \\
&\leq \Delta^{\mathbf{A}\mathbf{C}_0}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) + \Delta^{\mathbf{A}\mathbf{C}_1}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) \\
&\leq 2\varepsilon.
\end{aligned}$$

□

[9] One should not confuse the above LOR variant with *strong* non-malleability, the difference being that for strong non-malleability $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},b}$ would output (same, j) iff $c' = c^{(j)}$. In fact, being equivalent to non-malleability, our LOR variant is strictly weaker.

[10] The same LOR variant was already considered in [17, Definition A.1] (and referred to as "alternative" non-malleability). In this sense Lemma 8 and 9 below are a generalization of [17, Theorem A.1] to the adaptive and continuous case.

Lemma 9. *If* (Enc, Dec) *is* $(\mathcal{F}, \varepsilon, \ell, q)$*-LOR-non-malleable, it is also* $(\mathcal{F}, \varepsilon + \frac{q\ell}{2^k}, \ell, q)$*-non-malleable.*

Proof. Fix ℓ and q, and let $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} := \mathbf{S}^{\mathsf{real}}_{\mathcal{F},\ell,q}$, $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau} := \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau,\ell,q}$ (for a simulator τ to be defined next), $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0} := \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0,\ell,q}$, and $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1} := \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1,\ell,q}$. Consider the following simulator τ: It internally keeps a counter $i \leftarrow 0$. When invoked on (i', f) with $f \in \mathcal{F}^{(i')}$, if $i' > i$, it samples $x_1^{(j)} \leftarrow\!\!\$\ \{0,1\}^k \setminus \{x_1^{(1)}, \ldots, x_1^{(j-1)}\}$ and computes $c_1^{(j)} \leftarrow\!\!\$\ \mathrm{Enc}(x_1^{(j)})$ for all $i < j \leq i'$ and sets $i \leftarrow i'$. Then, it computes the tampered codeword $c' \leftarrow \mathrm{Dec}(f(c_1^{(1)}, \ldots, c_1^{(i)}))$ and decodes it to $x' \leftarrow \mathrm{Dec}(c')$. If $x' = x_1^{(j)}$ for some indices j, τ returns (same, j) for the largest such j. Otherwise, it returns x'.

Consider the following reduction $\mathbf{C}$: Upon the i^{th} query (encode, x) at the outside interface, it chooses $x_1^{(i)} \leftarrow\!\!\$\ \{0,1\}^k \setminus \{x_1^{(1)}, \ldots, x_1^{(i-1)}\}$, stores $x_0^{(i)} := x$ internally, and outputs $(\mathsf{encode}, x_0^{(i)}, x_1^{(i)})$ at the inside interface. Upon a query (tamper, f) at the outside interface, $\mathbf{C}$ outputs (tamper, f) at the inside interface and subsequently receives a value x' at the inside interface. If $x' = (\mathsf{same}, j)$ for some j, $\mathbf{C}$ outputs $x_0^{(j)}$ at the outside interface. Otherwise, it outputs x'.

Observe that $\mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1} \equiv \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$. In both cases, the i^{th} query of the type (encode, x) is treated by sampling fresh values $x_1^{(i)}$ distinct from all $x_1^{(1)}, \ldots, x_1^{(i-1)}$ and computing $c_1^{(i)}$ as an encoding of $x_1^{(i)}$. (This is delayed in $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$, but that does not change the distribution.) A query (tamper, f) with some function $f \in \mathcal{F}^{(i)}$ is answered by evaluating $f(c_1^{(1)}, \ldots, c_1^{(i)})$, decoding the resulting codeword to obtain a message x', and if $x' = x_1^{(j)}$ for some $j \in \{1, \ldots, i\}$, returning $x_0^{(j)}$ and x' otherwise.

The systems $\mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}$ and $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ are, however, not equivalent. The reason is that if, in $\mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}$, $\mathrm{Dec}(f(c_0^{(1)}, \ldots, c_0^{(i)})) = x_1^{(j)}$ for some $j \in \{1, \ldots, i\}$, then $\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}$ returns (same, j), which $\mathbf{C}$ replaces by $x_0^{(j)}$. There is no comparable behavior in $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$. Provoking this event, however, corresponds to "non-adaptively guessing" one of the values $x_1^{(j)}$, which occurs with probability at most $\frac{i}{2^k}$ in each query.

Formally, one can define a monotone binary output (MBO, see Section 2.1) on $\mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}$; $\widehat{\mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}}$ (the system extended by this additional output) and $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ are now conditionally equivalent, and by [38, Theorem 1], the distinguishing advantage $\Delta^{\mathbf{A}}(\mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}, \mathbf{S}^{\mathsf{real}}_{\mathcal{F}})$ is upper-bounded by the probability of provoking this event, which for at most ℓ encode- and at most q tamper-queries can be bounded by $\frac{q\ell}{2^k}$.

Hence, for all attackers $\mathbf{A}$,

$$\begin{aligned}
\Delta^{\mathbf{A}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) &= \Delta^{\mathbf{A}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1}) \\
&\leq \Delta^{\mathbf{A}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}) + \Delta^{\mathbf{A}}(\mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}, \mathbf{C}\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1}) \\
&\leq \tfrac{q\ell}{2^k} + \Delta^{\mathbf{AC}}(\mathbf{S}^{\mathsf{lor}}_{\mathcal{F},0}, \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},1}) \\
&\leq \tfrac{q\ell}{2^k} + \varepsilon.
\end{aligned}$$

□

Lemma 10. *If* (Enc, Dec) *is continuously* $(\mathcal{F}_{\mathsf{copy}}, \varepsilon, 1, q)$*-LOR-non-malleable, it is also continuously* $(\mathcal{F}_{\mathsf{copy}}, \ell \cdot \varepsilon, \ell, q)$*-LOR-non-malleable, for all* $\ell \in \mathbb{N}$.

Proof. Fix ℓ and q, let $\mathcal{F} := \mathcal{F}_{\mathsf{copy}}$, and set $\mathbf{S}'_b := \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},b,\ell,q}$ and $\mathbf{S}_b := \mathbf{S}^{\mathsf{lor}}_{\mathcal{F},b,1,q}$ for $b \in \{0,1\}$.

The distinguishing advantage between $\mathbf{S}'_0$ and $\mathbf{S}'_1$ is bounded via a hybrid argument, where the i^{th} hybrid $\mathbf{H}^{(i)}$ picks x_0 when processing the first i encode queries $(\mathsf{encode}, x_0, x_1)$ and x_1 afterwards.

For each i, the distinguishing advantage between successive hybrids $\mathbf{H}^{(i-1)}$ and $\mathbf{H}^{(i)}$ is bounded by exhibiting a system $\mathbf{C}_i$ that reduces distinguishing $\mathbf{S}_0$ and $\mathbf{S}_1$ to distinguishing the hybrids.

For $i = 0, 1, \ldots, \ell$, hybrid $\mathbf{H}^{(i)}$ works as follows: Initialization and (tamper, f) are defined as with $\mathbf{S}_0'$ and $\mathbf{S}_1'$. The first i queries $(\mathsf{encode}, x_0, x_1)$ are handled by encoding x_0, i.e., $c^{(j)} \leftarrow \mathrm{Enc}(x_0)$ for the j^{th} encoding. For all later queries, x_1 is encoded, i.e., $c^{(j)} \leftarrow \mathrm{Enc}(x_1)$.

One observes that

$$\mathbf{H}^{(\ell)} \equiv \mathbf{S}_0' \qquad \text{and} \qquad \mathbf{H}^{(0)} \equiv \mathbf{S}_1'.$$

For $i = 1, \ldots, n$, reduction $\mathbf{C}_i$ works as follows: For the first $i-1$ encode queries $(\mathsf{encode}, x_0, x_1)$ (at the outside interface), it computes and stores an encoding of x_0, i.e., $c^{(j)} \leftarrow \mathrm{Enc}(x_0)$ for the j^{th} encoding. Upon the i^{th} query $(\mathsf{encode}, x_0, x_1)$, it outputs $(\mathsf{encode}, x_0, x_1)$ at the inside interface. (Note that as a consequence, a target encoding $c \leftarrow\!\!\$\ \mathrm{Enc}(x_b)$ is generated, depending on whether $\mathbf{C}_i$ is connected to $\mathbf{S}_0$ or $\mathbf{S}_1$.) The remaining encode queries are handled by encoding the second message x_1, i.e., $c^{(j)} \leftarrow \mathrm{Enc}(x_1)$.

System $\mathbf{C}_i$ maintains a counter j that keeps track of the number of encode queries it has encountered. When a tamper query (tamper, f) with $f \in \mathcal{F}_{\mathsf{copy}}^{(j)}$ and $\chi(f) = (f_1, \ldots, f_n)$ is received at the outside interface, it computes $f_1', \ldots, f_n'$, where

$$f_v' := \begin{cases} f_v & \text{if } f_v \in \{\mathsf{zero}, \mathsf{one}\}, \\ \mathsf{zero} & \text{if } f_v = \mathsf{copy}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 0, \\ \mathsf{one} & \text{if } f_v = \mathsf{copy}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 1, \\ \mathsf{copy}_1 & \text{if } f_v = \mathsf{copy}_i. \end{cases}$$

Then, it outputs (tamper, f') at the inside interface, where f' is the function in $\mathcal{F}_{\mathsf{copy}}^{(1)}$ with $\chi(f') = (f_1', \ldots, f_n')$.[11] Let x' be the answer to the tamper query at the inside interface. $\mathbf{C}_i$ computes the set of indices j for which x' matches one of the two messages of the j^{th} encode query. Moreover, if $x' = \mathsf{same}$, index i is added to that set as well. Then, it outputs (same, j) for the largest index j in the set. If the set is empty, x' is output.

One observes that

$$\mathbf{C}_i\mathbf{S}_0 = \mathbf{H}^{(i)} \qquad \text{and} \qquad \mathbf{C}_i\mathbf{S}_1 = \mathbf{H}^{(i-1)}.$$

Thus, for all adversaries $\mathbf{A}$,

$$\begin{aligned} \Delta^{\mathbf{A}}(\mathbf{S}_0', \mathbf{S}_1') = \Delta^{\mathbf{A}}(\mathbf{H}^{(\ell)}, \mathbf{H}^{(0)}) &\leq \sum_{i=1}^{\ell} \Delta^{\mathbf{A}}(\mathbf{H}^{(i)}, \mathbf{H}^{(i-1)}) \\ &\leq \sum_{i=1}^{\ell} \Delta^{\mathbf{A}}(\mathbf{C}_i\mathbf{S}_0, \mathbf{C}_i\mathbf{S}_1) \leq \sum_{i=1}^{\ell} \Delta^{\mathbf{A}\mathbf{C}_i}(\mathbf{S}_0, \mathbf{S}_1) \leq \ell \cdot \varepsilon. \end{aligned}$$

□

Proof (of Theorem 2). Follows immediately from Lemmas 8, 9, and 10. □

[11] For simplicity, we assume here that $\mathbf{S}_0$ and $\mathbf{S}_1$ answer tamper queries consisting of zero and one instructions only even before a message has been encoded.

5 On the Necessity of Self-Destruct

In this section we show that no (k, n)-coding scheme $(\mathrm{Enc}, \mathrm{Dec})$ can achieve (even non-adaptive) continuous non-malleability against $\mathcal{F}_{\mathsf{copy}}$ without self-destruct. This fact is reminiscent of the negative result by Gennaro *et al.* [23]. The impossibility proof in this section assumes that Dec is deterministic and that $\mathrm{Dec}(\mathrm{Enc}(x)) = x$ with probability 1 for all $x \in \{0,1\}^k$ (cf. Definition 2). The distinguisher $\mathbf{D}$ provided by Theorem 11 is universal, i.e., it breaks any coding scheme (if given oracle access to its decoding algorithm).

For the remainder of this section, let $\mathcal{F} := \mathcal{F}_{\mathsf{set}}$ (as defined in Section 4), $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} := \mathbf{S}^{\mathsf{real}}_{\mathcal{F},1,n}$, and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau} := \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau,1,n}$ (with some simulator τ). Moreover, both $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ are stripped of the self-destruct mode.

Theorem 11. *There exists a distinguisher* $\mathbf{D}$ *such that for all coding schemes* $(\mathrm{Enc}, \mathrm{Dec})$ *and all simulators* τ*,*

$$\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) \geq 1 - \frac{n+1}{2^k}.$$

The corollary below states no pair of converters $(\mathsf{enc}, \mathsf{dec})$ can achieve the constructive statement corresponding to Theorem 1 without relying on the self-destruct feature.

Corollary 12. *For any protocol* $\mathsf{nmc} := (\mathsf{enc}, \mathsf{dec})$ *and all simulators* σ*, if both converters are stateless and*

$$[\overset{\text{1-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n \quad \overset{(\mathsf{enc},\mathsf{dec}),\sigma,(0,\varepsilon)}{\Longmapsto} \quad \overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet},$$

then,

$$\varepsilon \geq 1 - \frac{n+1}{2^k}.$$

Proof. Note that the protocol achieves perfect availability and thus constitutes a perfectly correct (k, n)-coding scheme (since the converters are stateless and with perfect correctness, dec can w.l.o.g. be assumed to be deterministic). Consider an arbitrary simulator σ. It can be converted into a simulator τ as required by Definition 3 in a straight-forward manner. Similarly, there exists a straight-forward reduction $\mathbf{C}$ such that

$$\mathbf{C}(\mathsf{enc}^A \mathsf{dec}^B [\overset{\text{1-bit},1,n}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}]^n) \equiv \mathbf{S}^{\mathsf{real}}_{\mathcal{F}} \qquad \text{and} \qquad \mathbf{C}(\sigma^E \overset{k\text{-bit},1,n}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}) \equiv \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}.$$

Thus, $\mathbf{DC}$ achieves advantage $1 - \frac{n+1}{2^k}$. □

5.1 Proof of Theorem 11

Distinguisher $\mathbf{D} := \mathbf{D}_{\mathsf{Ext}}$ uses an algorithm Ext that always extracts the encoded message when interacting with system $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ and does so with small probability only when interacting with system $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ (for any simulator).

The Extraction Algorithm. Consider the following algorithm Ext, which repeatedly issues tamper queries (tamper, f) with $f \in \mathcal{F}_{\mathsf{set}}$, expects an answer in $\{0,1\}^k \cup \{\diamond, \mathsf{same}\}$, and eventually outputs a value $x' \in \{0,1\}^k$: Initially, it initializes variables $f_1, \ldots, f_n \leftarrow \lambda$ (where the value λ stands for "undefined"). Then, for $i = 1, \ldots, n$ it proceeds as follows: It queries (tamper, f) with $\chi(f) = (f_1, \ldots, f_{i-1}, \mathsf{zero}, \mathsf{keep}, \ldots, \mathsf{keep})$. If the answer is same, it sets $f_i \leftarrow \mathsf{zero}$ and otherwise $f_i \leftarrow \mathsf{one}$. In the end Ext outputs $x' \leftarrow \mathrm{Dec}(\mathsf{val}(f_1) \cdots \mathsf{val}(f_n))$.

The Distinguisher. Consider the following distinguisher $\mathbf{D}_{\mathsf{Ext}}$: Initially, it chooses $x \leftarrow \{0,1\}^k$ and outputs (encode, x) to the system it is connected to. Then, it lets Ext interact with that system, replacing an answer by same whenever it is x. When Ext terminates and outputs a value x', $\mathbf{D}_{\mathsf{Ext}}$ outputs 1 if $x' = x$ and 0 otherwise.

Lemma 13. $\mathsf{P}[\mathbf{D}_{\mathsf{Ext}}\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} = 1] = 1$.

Proof. Assume that before the i^{th} iteration of Ext, asking the query (tamper, f) with $\chi(f) = (f_1, \ldots, f_{i-1}, \mathsf{keep}, \mathsf{keep}, \ldots, \mathsf{keep})$ to $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ yields the answer x. From this it follows that either $(f_1, \ldots, f_{i-1}, \mathsf{zero}, \mathsf{keep}, \ldots, \mathsf{keep})$ or $(f_1, \ldots, f_{i-1}, \mathsf{one}, \mathsf{keep}, \ldots, \mathsf{keep})$ leads to the answer x; Ext sets f_i appropriately (the fact that the answer x is replaced by same plays no role here). Thus, in the end, computing $\mathrm{Dec}(\mathsf{val}(f_1) \cdots \mathsf{val}(f_n))$ yields x. □

In other words, Lemma 13 means that Ext always succeeds at recovering the value x chosen by $\mathbf{D}$. Showing that this happens only with small probability when $\mathbf{D}_{\mathsf{Ext}}$ interacts with $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ completes the proof.

Lemma 14. $\mathsf{P}[\mathbf{D}_{\mathsf{Ext}}\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau} = 1] \leq \frac{n+1}{2^k}$.

Proof. Consider the following modified distinguisher $\hat{\mathbf{D}}_{\mathsf{Ext}}$ that works as $\mathbf{D}_{\mathsf{Ext}}$ except that it does *not* modify the answers received by the system it is connected to. Moreover, let $\hat{\mathbf{S}}^{\mathrm{simu}}_{\mathcal{F},\tau}$ be the the system that ignores all encode-queries and handles queries (tamper, f) by invoking $\tau(1, f)$ and outputting τ's answer.

Note that in both experiments, Ext's view is identical unless it causes τ to output x (the value encoded by $\mathbf{D}$), which happens with probability at most $\frac{n}{2^k}$. Thus,

$$|\mathsf{P}^{\mathbf{D}_{\mathsf{Ext}}\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}}[\mathsf{Ext} \text{ outputs } x] - \mathsf{P}^{\hat{\mathbf{D}}_{\mathsf{Ext}}\hat{\mathbf{S}}^{\mathrm{simu}}_{\mathcal{F},\tau}}[\mathsf{Ext} \text{ outputs } x]| \leq \frac{n}{2^k}.$$

Furthermore, in experiment $\hat{\mathbf{D}}_{\mathsf{Ext}}\hat{\mathbf{S}}^{\mathrm{simu}}_{\mathcal{F},\tau}$, Ext's view is independent of x, and therefore, x is output by Ext with probability $\frac{1}{2^k}$. The claim follows. □

6 Conclusions

We have shown how non-malleable codes can be used to obtain a construction of a multi-bit chosen-ciphertext secure PKE scheme from a single-bit chosen-ciphertext secure one, which is one of the first applications of non-malleable codes outside the area of tamper resilience. Our construction is quite efficient and very intuitive. Its decryption algorithm needs to keep a single bit of state, which is acceptable for practical applications. In general, this suggests that dropping the usual requirement that the decryption be stateless may lead to the discovery of better schemes.

The formalization in constructive cryptography allowed us to focus on the technically most challenging part—proving that our code satisfies an extension of the original non-malleability requirement—and to keep this proof purely information-theoretical. The reduction from breaking the security of the single-bit scheme to breaking the security of our construction we then obtain, using the composition theorem of constructive cryptography, as a corollary from our results.

Acknowledgments. We thank Joël Alwen and Daniel Tschudi for helpful discussions, in particular on the impossibility proof in Section 5. The work was supported by the Swiss National Science Foundation (SNF), project no. 200020-132794.

References

[1] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:81, 2013. To appear in STOC 2014.

[2] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes resistant to permutations. Cryptology ePrint Archive, Report 2014/316, May 2014.

[3] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing chosen-ciphertext security. In *CRYPTO*, pages 565–582, 2003.

[4] Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In *ITCS*, pages 155–168, 2014.

[5] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *TCC*, pages 440–464, 2014.

[6] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In *TCC*, pages 427–444, 2008.

[7] Seung Geol Choi, Aggelos Kiayias, and Tal Malkin. BiTR: Built-in tamper resilience. In *ASIACRYPT*, pages 740–758, 2011.

[8] Sandro Coretti, Ueli Maurer, and Björn Tackmann. Constructing confidential channels from authenticated channels - public-key encryption revisited. In *ASIACRYPT (1)*, pages 134–153, 2013.

[9] Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi. Self-destruct chosen-ciphertext security from semantic security, 2014. Manuscript.

[10] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *EUROCRYPT*, pages 471–488, 2008.

[11] Ronald Cramer, Goichiro Hanaoka, Dennis Hofheinz, Hideki Imai, Eike Kiltz, Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Bounded CCA2-secure encryption. In *ASIACRYPT*, pages 502–518, 2007.

[12] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25, Heidelberg, 1998. Springer.

[13] Dana Dachman-Soled. A black-box construction of a CCA2 encryption scheme from a plaintext aware encryption scheme. In Hugo Krawczyk, editor, *PKC*, LNCS. Springer, 2014.

[14] Ivan Damgård, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. In *ASIACRYPT (2)*, pages 140–160, 2013.

[15] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[16] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In *CRYPTO (2)*, pages 239–257, 2013.

[17] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *ICS*, pages 434–452, 2010.

[18] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, 1984.

[19] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *TCC*, pages 465–488, 2014.

[20] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. A leakage and tamper resilient random access machine, 2014. Manuscript.

[21] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *EUROCRYPT*, pages 111–128, 2014.

[22] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP is secure under the RSA assumption. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 260–274, Heidelberg, 2001. Springer.

[23] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In *TCC*, pages 258–277, 2004.

[24] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and CCA security for public key encryption. In *TCC*, pages 434–455, 2007.

[25] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[26] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332, Heidelberg, 2009. Springer.

[27] Susan Hohenberger, Allison B. Lewko, and Brent Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. In *EUROCRYPT*, pages 663–681, 2012.

[28] Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 590–609, Heidelberg, 2009. Springer.

[29] Markulf Kohlweiss, Ueli Maurer, Cristina Onete, Björn Tackmann, and Daniele Venturi. Anonymity-preserving public-key encryption: A constructive approach. In *Privacy Enhancing Technologies*, pages 19–39, 2013.

[30] Huijia Lin and Stefano Tessaro. Amplification of chosen-ciphertext security. In *EUROCRYPT*, pages 503–519, 2013.

[31] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *CRYPTO*, pages 517–532, 2012.

[32] Ueli Maurer. Constructive cryptography - a new paradigm for security definitions and proofs. In *TOSCA*, pages 33–56, 2011.

[33] Ueli Maurer. Conditional equivalence of random systems and indistinguishability proofs. In *2013 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 3150–3154, 2013.

[34] Ueli Maurer and Renato Renner. Abstract cryptography. In *ICS*, pages 1–21, 2011.

[35] Ueli Maurer, Andreas Rüedlinger, and Björn Tackmann. Confidentiality and integrity: A constructive perspective. In *TCC*, pages 209–229, 2012.

[36] Ueli Maurer and Pierre Schmid. A calculus for security bootstrapping in distributed systems. *Journal of Computer Security*, 4(1):55–80, 1996.

[37] Ueli Maurer and Björn Tackmann. On the soundness of authenticate-then-encrypt: formalizing the malleability of symmetric encryption. In *ACM Conference on Computer and Communications Security*, pages 505–515, 2010.

[38] Ueli M. Maurer. Indistinguishability of random systems. In *EUROCRYPT*, pages 110–132, 2002.

[39] Steven Myers, Mona Sergi, and Abhi Shelat. Blackbox construction of a more than non-malleable CCA1 encryption scheme from plaintext awareness. In Ivan Visconti and Roberto De Prisco, editors, *Security and Cryptography for Networks*, volume 7485 of *LNCS*, pages 149–165. Springer, 2012.

[40] Steven Myers and Abhi Shelat. Bit encryption is complete. In *FOCS*, pages 607–616, 2009.

[41] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *SIAM J. Comput.*, 40(6):1803–1844, 2011.

[42] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems (reprint). *Commun. ACM*, 26(1):96–99, 1983.

[43] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. *SIAM J. Comput.*, 39(7):3058–3088, 2010.

A Non-Malleable Codes and the One-Time Pad

The one-time pad encryption scheme is strongly malleable: if a transmitted ciphertext $e \in \{0,1\}^n$ (corresponding to some message $m \in \{0,1\}^n$) is replaced by a different ciphertext $e' \in \{0,1\}^n$, then the decryption of e' will result in $m \oplus (e \oplus e')$. From the attacker's perspective, the one-time pad is *XOR-malleable*: by replacing the ciphertext e by $e \oplus \delta$ for some $\delta \in \{0,1\}^n$, he can maul the plaintext from m into $m \oplus \delta$. If one encodes the message with a non-malleable code prior to encrypting with the one-time pad, however, the malleability disappears. Let us stress that there are other, more efficient, ways of achieving the same effect; we analyze this scheme here to prepare for the analysis of our main scheme in Section 3, which follows the same approach.

We first explicitly describe the channel that is constructed by the one-time pad from an insecure channel and an (n-bit) shared secret key, namely the *XOR-malleable confidential channel* $-\!\!\oplus\!\!\rightarrow\!\!\bullet$ as described in [37]. This channel, which exactly formalizes that the attacker can specify a bit mask $\delta \in \{0,1\}^n$ (but not more), is described as follows. (The proof is a restricted case of [37, Lemma 2].)

XOR-Malleable Confidential n-bit Channel $\overset{n\text{-bit}}{-\!\!\oplus\!\!\rightarrow}\!\!\bullet$

Initially take a bit $b \in \{0,1\}$ at the E-interface. If $b = 0$ then:

1. On input $m \in \{0,1\}^n$ at the A-interface, output m at the B-interface.

Otherwise, if $b = 1$, then:

1. On input $m \in \{0,1\}^n$ at the A-interface, output $|m|$ at the E-interface.
2. On input $\delta \in \{0,1\}^n$ at the E-interface, output $m \oplus \delta$ at the B-interface.

We then assume the existence of a (k,n)-coding scheme $(\mathrm{Enc}, \mathrm{Dec})$ which is $(\mathcal{F}_{\mathsf{bit}}, \varepsilon)$-non-malleable (this corresponds to adaptive continuous $(\mathcal{F}_{\mathsf{bit}}, 1, 1, \varepsilon)$-non-malleability according to Definition 3), and describe converters enc and dec as follows:

- The converter enc, obtaining a message $m \in \{0,1\}^k$ at its outside interface, computes $c \leftarrow\!\!\$\ \mathrm{Enc}(m)$ and outputs c at its inside interface.
- The converter dec, obtaining a message $c' \in \{0,1\}^n$ at its inside interface, computes $m' \leftarrow \mathrm{Dec}(c')$ and outputs m' at its outside interface.

We claim that the protocol $(\mathsf{enc}, \mathsf{dec})$ constructs, from the XOR-malleable n-bit channel $\overset{n\text{-bit}}{-\!\!\oplus\!\!\rightarrow}\!\!\bullet$, the non-malleable k-bit channel $\xrightarrow{k\text{-bit}}\!\!\bullet$ described below. Intuitively, a non-malleable channel allows the attacker to inject any *fixed* message of its choice, in the sense that the message transmitted to the receiver does not depend on the originally sent message.

(Non-Malleable) Confidential n-bit Channel $\xrightarrow{n\text{-bit}}\!\!\bullet$

Initially take a bit $b \in \{0,1\}$ at the E-interface. If $b = 0$ then:

1. On input $m \in \{0,1\}^n$ at the A-interface, output m at the B-interface.

Otherwise, if $b = 1$, then:

1. On input $m \in \{0,1\}^n$ at the A-interface, output $|m|$ at the E-interface.
2. On input $1 \in \mathbb{N}$ at the E-interface, output m at the B-interface and halt.
3. On input $m' \in \{0,1\}^n$ at the E-interface, output m' at the B-interface and halt.

The formal construction statement is as follows.

Lemma 15. *Assume that* $(\mathrm{Enc}, \mathrm{Dec})$ *is a* (k, n)*-coding scheme that is* $(\mathcal{F}_{\mathsf{bit}}, \varepsilon)$*-non-malleable. Then*

$$\mathsf{enc}^A\mathsf{dec}^B\bot^E \overset{n\text{-bit}}{-\!\oplus\!\!\!\rightarrow}\bullet \quad\equiv\quad \bot^E \overset{k\text{-bit}}{\longrightarrow}\bullet \tag{9}$$

and there is a simulator σ_{xor} *such that for all distinguishers* $\mathbf{D}$*,*

$$\Delta^{\mathbf{D}}\left(\mathsf{enc}^A\mathsf{dec}^B \overset{n\text{-bit}}{-\!\oplus\!\!\!\rightarrow}\bullet, \sigma_{\mathrm{xor}}^E \overset{k\text{-bit}}{\longrightarrow}\bullet\right) \leq \varepsilon. \tag{10}$$

Proof. Condition (9) follows from the correctness of the scheme, i.e., Definition 2. Let $\mathcal{F} := \mathcal{F}_{\mathsf{bit}}$, $\mathbf{S}_{\mathcal{F}}^{\mathsf{real}} := \mathbf{S}_{\mathcal{F},1,1}^{\mathsf{real}}$ and $\mathbf{S}_{\mathcal{F},\tau}^{\mathsf{simu}} := \mathbf{S}_{\mathcal{F},\tau,1,1}^{\mathsf{simu}}$, where τ is the simulator guaranteed to exist by Definition 3. For condition (10), we describe a simulator σ_{xor} as follows:

- On input the message length k at the inside interface, output n at the outside interface.
- On input the mask $\delta \in \{0,1\}^n$ at the outside interface, invoke τ as $m' \leftarrow\!\!\$\; \tau(f_\delta)$ with $\chi(f_\delta) = (f_1, \ldots, f_n)$ and $f_i = \mathsf{keep}$ (resp., $f_i = \mathsf{flip}$) iff $\delta[i] = 0$ (resp., $\delta[i] = 1$). If $m' = \mathsf{same}$ then output $1 \in \mathbb{N}$ at the inside interface, otherwise input m'.

To conclude the proof, we describe a reduction $\mathbf{C}$ that, once connected with its inside interface to $\mathbf{S}_{\mathcal{F}}^{\mathsf{real}}$ (resp. $\mathbf{S}_{\mathcal{F},\tau}^{\mathsf{simu}}$), behaves as $\mathsf{enc}^A\mathsf{dec}^B \overset{n\text{-bit}}{-\!\oplus\!\!\!\rightarrow}\bullet$ (resp. $\sigma_{\mathrm{xor}}^E \overset{k\text{-bit}}{\longrightarrow}\bullet$). This converter provides at the outside interface three sub-interfaces (labeled A, B, and E), and behaves as follows:

- Upon input a value $m \in \{0,1\}^k$ at the outside A-sub-interface, output n at the outside E-sub-interface.
- Upon input a value $\delta \in \{0,1\}^n$ at the outside E-sub-interface, define $f_\delta \in \mathcal{F}$ such that $\chi(f_\delta) = (f_1, \ldots, f_n)$ with $f_i = \mathsf{keep}$ (resp., $f_i = \mathsf{flip}$) iff $\delta[i] = 0$ (resp., $\delta[i] = 1$), and output $(\mathsf{tamper}, f_\delta)$ at the inside interface.
- Obtaining the response $m' \in \{0,1\}^k$ at the inside interface, output m' at the outside B-sub-interface.

The output at the E-sub-interface upon input $m \in \{0,1\}^k$ at the A-sub-interface is always consistent (namely, n). For $\mathbf{CS}_{\mathcal{F}}^{\mathsf{real}}$, the output at the B-interface on input $\delta \in \{0,1\}^n$ at the E-interface is computed by applying the tampering function f_δ to the encoding of the value m; exactly as in $\mathsf{enc}^A\mathsf{dec}^B \overset{n\text{-bit}}{-\!\oplus\!\!\!\rightarrow}\bullet$. Analogously, in $\mathbf{CS}_{\mathcal{F},\tau}^{\mathsf{simu}}$, the output at the B-interface on input $\delta \in \{0,1\}^n$ at the E-interface is computed by applying invoking the simulator τ on the tampering function f_δ; exactly as in $\sigma_{\mathrm{xor}}^E \overset{n\text{-bit}}{\longrightarrow}\bullet$. As a result, we obtain

$$\Delta^{\mathbf{D}}\left(\mathsf{enc}^A\mathsf{dec}^B \overset{n\text{-bit}}{-\!\oplus\!\!\!\rightarrow}\bullet, \sigma_{\mathrm{xor}}^E \overset{n\text{-bit}}{\longrightarrow}\bullet\right) = \Delta^{\mathbf{D}}\left(\mathbf{CS}_{\mathcal{F}}^{\mathsf{real}}, \mathbf{CS}_{\mathcal{F},\tau}^{\mathsf{simu}}\right) = \Delta^{\mathbf{DC}}\left(\mathbf{S}_{\mathcal{F}}^{\mathsf{real}}, \mathbf{S}_{\mathcal{F},\tau}^{\mathsf{simu}}\right) \leq \varepsilon,$$

where $\mathbf{DC}$ makes at most one tamper-query. This concludes the proof. □

A non-malleable confidential channel still allows an attacker to inject messages. A fully secure channel, denoted as $\bullet\!\!\longrightarrow\!\!\bullet$, in contrast, allows the attacker only to delay or drop messages. More formally, the secure channel is described as follows.

Secure n-bit Channel $\bullet\!\xrightarrow{n\text{-bit}}\!\bullet$

Initially take a bit $b \in \{0,1\}$ at the E-interface. If $b = 0$ then:

1. On input $m \in \{0,1\}^n$ at the A-interface, output m at the B-interface.

Otherwise, if $b = 1$, then:

1. On input $m \in \{0,1\}^n$ at the A-interface, output $|m|$ at the E-interface.
2. On input $1 \in \mathbb{N}$ at the E-interface, output m at the B-interface and halt.

If we assume the availability of some shared secret key $\bullet\!\overset{\ell\text{-bit}}{=\!=\!=}\!\bullet$ for some $\ell \leq k$ in parallel to a confidential k-bit channel $\xrightarrow{k\text{-bit}}\!\bullet$, we can securely transmit a $(k-\ell)$-bit message by simply appending the key. The key is also specified as a resource as follows.

ℓ-bit Secret Key $\bullet\!\overset{\ell\text{-bit}}{=\!=\!=}\!\bullet$

Choose $\kappa \in \{0,1\}^\ell$ uniformly at random, output κ at the A- and B-interfaces.

Let $(\mathsf{app}, \mathsf{chk})$ be the pair of converters that appends the key to the transmitted message, and in more detail works as follows:

- The converter app, upon obtaining a message $m \in \{0,1\}^{k-\ell}$ at the outside and a key $\kappa \in \{0,1\}^\ell$ at the (first sub-interface of the) inside interface, outputs $m|\kappa$ at the (second sub-interface of the) inside interface.
- The converter chk, upon obtaining a message $x \in \{0,1\}^k$ and a key $\kappa \in \{0,1\}^\ell$, checks whether $x = m|\kappa$ for some $m \in \{0,1\}^{k-\ell}$, and in that case, outputs m at the outside interface. (Otherwise nothing.)

This protocol constructs from the ℓ-bit key and the k-bit confidential channel a $(k-\ell)$-bit secure channel.

Lemma 16. *Let* $(\mathsf{app}, \mathsf{chk})$ *be the protocol described above, then:*

$$\mathsf{app}^A \mathsf{chk}^B \bot^E \left[\bullet\!\overset{\ell\text{-bit}}{=\!=\!=}\!\bullet, \xrightarrow{k\text{-bit}}\!\bullet\right] \quad \equiv \quad \bot^E \ \bullet\!\xrightarrow{(k-\ell)\text{-bit}}\!\bullet \tag{11}$$

and there is a simulator σ such that for all distinguishers $\mathbf{D}$*,*

$$\Delta^{\mathbf{D}}\left(\mathsf{app}^A \mathsf{chk}^B \left[\bullet\!\overset{\ell\text{-bit}}{=\!=\!=}\!\bullet, \xrightarrow{k\text{-bit}}\!\bullet\right], \sigma^E \ \bullet\!\xrightarrow{(k-\ell)\text{-bit}}\!\bullet\right) \leq 2^{-\ell}. \tag{12}$$

Proof sketch. Equation (11) is again easy to verify. To conclude the correctness of equation (12), we use the simulator σ that upon input $(k-\ell)$ at the inside interface outputs k at the outside interface. In case of an input $1 \in \mathbb{N}$ at the outside interface, σ also outputs $1 \in \mathbb{N}$ at the inside interface; and in case of an input $m' \in \{0,1\}^k$ at the outside interface, σ simply halts.

We first see that inputting $1 \in \mathbb{N}$ does not benefit the distinguisher, as the output at the B-interface is exactly the message input at the A-interface. Then we see that the only possibility to input a value $m' \in \{0,1\}^k$ and obtain *some* output in $\mathsf{app}^A \mathsf{chk}^B \left[\bullet\!\overset{\ell\text{-bit}}{=\!=\!=}\!\bullet, \xrightarrow{k\text{-bit}}\!\bullet\right]$ (note that $\sigma^E \ \bullet\!\xrightarrow{(k-\ell)\text{-bit}}\!\bullet$ will never give any output) is to guess the ℓ-bit secret key, which happens with probability at most $2^{-\ell}$. This concludes the proof. $\square$

B The Composition Theorem of Constructive Cryptography

The main statement we prove in the main paper shows the security of one protocol step in isolation, i.e. we show for the non-malleable code that it constructs the multi-bit confidential channel from multiple assumed single-bit confidential channels. The composition theorem now states that two such construction steps can be composed: if one (lower-level) protocol constructs the resource that is assumed by the other (higher-level) protocol, then the composition of those two protocols constructs the same resource as the higher-level protocol, but from the resources assumed by the lower-level protocol, under the assumptions that occur in (at least) one of the individual security statements. To state the theorem, we make use of a special converter id that behaves transparently (i.e., allows access to the underlying interface of the resource).

The composition theorem was first explicitly stated in [37], but the statement there was restricted to asymptotic settings. Later, in [29], the theorem was stated in a way that also allows to capture concrete security statements. The proof, however, still follows the same steps as the one in [37]. For the statement of the theorem we assume the operation $[\cdot, \dots, \cdot]$ to be left-associative; in this way we can simply express multiple resources using the single variable $\mathbf{U}$.

Theorem 17. *Let $\mathbf{R}, \mathbf{S}, \mathbf{T}, \mathbf{U} \in \Phi$ be resources. Let $\pi = (\pi_1, \pi_2)$ and $\psi = (\psi_1, \psi_2)$ be protocols, σ_π and σ_ψ be simulators, and $(\varepsilon_\pi^1, \varepsilon_\pi^2)$, $(\varepsilon_\psi^1, \varepsilon_\psi^2)$ such that*

$$\mathbf{R} \quad \overset{\pi, \sigma_\pi, (\varepsilon_\pi^1, \varepsilon_\pi^2)}{\Longmapsto} \quad \mathbf{S} \qquad \text{and} \qquad \mathbf{S} \quad \overset{\psi, \sigma_\psi, (\varepsilon_\psi^1, \varepsilon_\psi^2)}{\Longmapsto} \quad \mathbf{T}.$$

Then

$$\mathbf{R} \quad \overset{\alpha, \sigma_\alpha, (\varepsilon_\alpha^1, \varepsilon_\alpha^2)}{\Longmapsto} \quad \mathbf{T}$$

with $\alpha = (\psi_1 \circ \pi_1, \psi_2 \circ \pi_2)$, $\sigma_\alpha = \sigma_\pi \circ \sigma_\psi$, and $\varepsilon_\alpha^i(\mathbf{D}) = \varepsilon_\pi^i(\mathbf{D}\sigma_\psi^E) + \varepsilon_\psi^i(\mathbf{D}\pi_1^A \pi_2^B)$, where $\mathbf{D}\sigma_\psi^E$ and $\mathbf{D}\pi_1^A \pi_2^B$ mean that $\mathbf{D}$ applies the converters at the respective interfaces. Moreover

$$[\mathbf{R}, \mathbf{U}] \quad \overset{[\pi, (\mathsf{id}, \mathsf{id})], [\sigma_\pi, \mathsf{id}], (\bar{\varepsilon}_\pi^1, \bar{\varepsilon}_\pi^2)}{\Longmapsto} \quad [\mathbf{S}, \mathbf{U}],$$

with $\bar{\varepsilon}_\pi^i(\mathbf{D}) = \varepsilon_\pi^i(\mathbf{D}[\cdot, \mathbf{U}])$, where $\mathbf{D}[\cdot, \mathbf{U}]$ means that the distinguisher emulates $\mathbf{U}$ in parallel. (The analogous statement holds with respect to $[\mathbf{U}, \mathbf{R}]$ and $[\mathbf{U}, \mathbf{S}]$.)

C (Replayable) Self-Destruct Chosen Ciphertext Security

In Section 3, based on a 1-bit CCA-secure PKE scheme, we provide a protocol (a pair of converters) $\mathsf{pke} = (\mathsf{encrypt}, \mathsf{decrypt})$ that achieves transformation

$$[\longleftarrow\!\!\bullet, - \twoheadrightarrow, \mathsf{FLAG}] \quad \overset{\mathsf{pke}}{\Longmapsto} \quad \overset{k\text{-bit}}{-\!\!\diamond\!\!\twoheadrightarrow\!\!\bullet}\ . \tag{13}$$

An alternative view is that we in fact implicitly provide a PKE scheme $\Pi = (K, E, D)$. In rough terms, key generation algorithm K, generates n independent key pairs of the 1-bit scheme. Encryption algorithm E first encodes a message using a non-malleable code and then encrypts each bit of the resulting encoding independently and outputs the n resulting ciphertexts. Decryption algorithm D first decrypts the n ciphertexts, decodes the resulting bitstring, and outputs the decoded message or the symbol $\diamond$, indicating an invalid ciphertext, if any of these steps fails.

System $\mathbf{G}_b^{\textsf{sd-rcca}}$

init
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow K$
  output pk

on (chall, m_0)
  $m_1 \leftarrow\!\!\$\; \mathcal{M}$ s.t. $|m_1| = |m_0|$
  $c \leftarrow E_{\mathsf{pk}}(m_b)$
  output c

on (dec, c')
  $m' \leftarrow D_{\mathsf{sk}}(c')$
  if $m' = \diamond$
    self-destruct
  else if $m' \in \{m_0, m_1\}$
    output test
  else
    output m'

Figure 9: System $\mathbf{G}_b^{\textsf{sd-rcca}}$, where $b \in \{0,1\}$, defining SD-RCCA security of a PKE scheme $\Pi = (K, E, D)$. The command **self-destruct** causes the system to output $\diamond$ and to answer all future decryption queries by $\diamond$.

From scheme Π, converters $\mathsf{encrypt}$ and $\mathsf{decrypt}$ are recovered as follows: Converter $\mathsf{encrypt}$ initially expects a public key pk at the inside interface. When a message m is input at the outside interface, $\mathsf{encrypt}$ outputs $c \leftarrow\!\!\$\; E_{\mathsf{pk}}(m)$ at the inside interface. Converter $\mathsf{decrypt}$ initially generates a key pair $(\mathsf{pk}, \mathsf{sk})$ using K and outputs pk at the inside interface. When $\mathsf{decrypt}$ receives a ciphertext c' at the inside interface, it first outputs read at the inside interface of FLAG to obtain a bit β. In case $\beta = 0$, $\mathsf{decrypt}$ computes $m' \leftarrow D_{\mathsf{sk}}(c')$ and outputs m' at the outside interface. In case $m' = \diamond$, $\mathsf{decrypt}$ also outputs set at the inside interface of FLAG. In case $\beta = 1$, $\mathsf{decrypt}$ outputs $\diamond$ at its outside interface.

In the remainder of this section, we show that our scheme achieves *replayable self-destruct chosen-ciphertext security (SD-RCCA)*,[12] a CCA variant in which the decryption oracle stops working after receiving an invalid ciphertext.

C.1 Formal Definition

The only difference between the SD-RCCA game and the standard game used to define RCCA is that the decryption oracle self-destructs, i.e., it stops processing further queries once an invalid ciphertext is ever queried. Note that the self-destruct feature only affects the decryption oracle; the adversary is still allowed to get the challenge ciphertext after provoking a self-destruct. For convenience, the game is phrased as a distinguishing problem between the two systems $\mathbf{G}_0^{\textsf{sd-rcca}}$ and $\mathbf{G}_1^{\textsf{sd-rcca}}$ described in Figure 9.

C.2 Security Proof

It remains to prove that our PKE scheme is indeed SD-RCCA secure. This is achieved by showing that whenever *any* protocol $\mathsf{pke} = (\mathsf{encrypt}, \mathsf{decrypt})$ built from a PKE scheme Π as above achieves construction (13), then Π is SD-RCCA secure.

In the following, let

$$\mathbf{U} := \mathsf{encrypt}^A \mathsf{decrypt}^B [\longleftarrow\!\!\bullet, -\twoheadrightarrow, \mathsf{FLAG}] \qquad \text{and} \qquad \mathbf{V} := \sigma^E \overset{k\text{-bit}}{\multimap\!\!\twoheadrightarrow\!\!\bullet},$$

[12]The notion of *replayable* CCA security was introduced by [3] to deal with the artificial strictness of full CCA security.

where σ is an *arbitrary* simulator.

Theorem 18. *There exist efficient reductions* $\mathbf{C}_0$ *and* $\mathbf{C}_1$ *such that, for all adversaries* $\mathbf{A}$*,*

$$\Delta^{\mathbf{A}}(\mathbf{G}_0^{\textsf{sd-rcca}}, \mathbf{G}_1^{\textsf{sd-rcca}}) \leq \Delta^{\mathbf{A}\mathbf{C}_0}(\mathbf{U}, \mathbf{V}) + \Delta^{\mathbf{A}\mathbf{C}_1}(\mathbf{U}, \mathbf{V}).$$

Proof. Consider the following reductions $\mathbf{C}_0$ and $\mathbf{C}_1$. Both connect to an $\{A, B, E\}$-resource on the inside and provide a single interface on the outside: Initially, both obtain pk at the inside E-interface and output pk at the outside interface. When (chall, m_0) is received on the outside, *both* systems choose a random message m_1. $\mathbf{C}_0$ outputs m_0 at the inside A-interface and $\mathbf{C}_1$ outputs m_1. Subsequently, c is received at the inside E-interface, and c is output on the outside by both systems. When a decryption query (dec, c') is received on the outside, both systems output c' at the inside E-interface. A subsequently received message m' at B is output on the outside by both systems (as answer to the decryption query) unless $m' \in \{m_0, m_1\}$, in which case test is returned. Moreover, if $m' = \diamond$, both reduction systems self-destruct, i.e., they answer all future decryption queries by $\diamond$. We have

$$\mathbf{C}_0\mathbf{U} \equiv \mathbf{G}_0^{\textsf{sd-rcca}} \qquad \text{and} \qquad \mathbf{C}_1\mathbf{U} \equiv \mathbf{G}_1^{\textsf{sd-rcca}} \qquad \text{and} \qquad \mathbf{C}_0\mathbf{V} \equiv \mathbf{C}_1\mathbf{V},$$

where the last equivalence follows from the fact that, in $\mathbf{V}$, the input from $\overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$ to σ is the same in both systems (the length of the message input at the A-interface of $\overset{k\text{-bit}}{-\!\diamond\!\!\twoheadrightarrow\!\bullet}$) and that decryption queries causing m_0 or m_1 to be output at the B-interface are answered by test. Hence,

$$\begin{aligned}\Delta^{\mathbf{A}}(\mathbf{G}_0^{\textsf{sd-rcca}}, \mathbf{G}_1^{\textsf{sd-rcca}}) &= \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{U}, \mathbf{C}_1\mathbf{U}) \leq \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{U}, \mathbf{C}_0\mathbf{V}) + \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{V}, \mathbf{C}_1\mathbf{V}) + \Delta^{\mathbf{A}}(\mathbf{C}_1\mathbf{V}, \mathbf{C}_1\mathbf{U}) \\ &= \Delta^{\mathbf{A}\mathbf{C}_0}(\mathbf{U}, \mathbf{V}) + \Delta^{\mathbf{A}\mathbf{C}_1}(\mathbf{U}, \mathbf{V}).\end{aligned}$$

□

D Continuous Non-Malleability against Full Bit-Wise Tampering

In this section we show that the coding scheme by [17] is continuously non-malleable against $\mathcal{F}_{\mathsf{copy}}$ extended with bit flips. The scheme relies on a LECSS (E, D) (cf. Definition 4 in Section 4) and a so-called AMD code (A, V); the latter concept was introduced by [10].

Definition 6 (AMD code)**.** A (k, n)-coding scheme (A, V) is a ρ-secure algebraic manipulation detection (AMD) code if for all $x \in \{0,1\}^n$ and non-zero $\Delta \in \{0,1\}^n$, $\mathsf{P}[\mathsf{V}(\mathsf{A}(x) + \Delta) \neq \diamond] \leq \rho$.

The scheme $(\mathrm{Enc}, \mathrm{Dec})$ by [17] is the concatenation of an AMD code and a LECSS, i.e., $\mathrm{Enc} := \mathsf{E} \circ \mathsf{A}$ and $\mathrm{Dec} := \mathsf{V} \circ \mathsf{D}$, where $\mathsf{V}(\diamond) = \diamond$.

The tampering class $\mathcal{F}_{\mathsf{copy}}$ can be extended to account for bit flips: Let $\mathcal{F}'_{\mathsf{copy}} := (\mathcal{F}'^{(i)}_{\mathsf{copy}})_{i \geq 1}$ where $\mathcal{F}'^{(i)}_{\mathsf{copy}} \subseteq \{f \mid f : (\{0,1\}^n)^i \to \{0,1\}^n\}$ and each function $f \in \mathcal{F}'^{(i)}_{\mathsf{copy}}$ is characterized by a vector $\chi(f) = (f_1, \ldots, f_n)$ where $f_i \in \{\mathsf{zero}, \mathsf{one}, \mathsf{copy}_1, \ldots, \mathsf{copy}_i, \mathsf{flip}_1, \ldots, \mathsf{flip}_i\}$, with the meaning that f takes as input i codewords $(c^{(1)}, \ldots, c^{(i)})$ and outputs a codeword $c' = (c'[1], \ldots, c'[n])$ in which each bit is either set to 0 (zero), set to 1 (one), copied from the *corresponding* bit in a codeword $c^{(j)}$ (copy_j), or copied and flipped from the corresponding bit in a codeword $c^{(j)}$ (flip_j).

Theorem 19. *Let* $(\mathrm{Enc}, \mathrm{Dec})$ *as defined above with a* (t,d)*-LECSS* (k,n)*-code for* $d > n/4$ *and* $d > t$ *and a* ρ*-secure AMD code. Then* $(\mathrm{Enc}, \mathrm{Dec})$ *is* $(\mathcal{F}_{\mathsf{copy}}, \varepsilon, 1, q)$*-continuously non-malleable for all* $q \in \mathbb{N}$ *and*

$$\varepsilon = 3 \cdot 2^{-t} + \left(\frac{t}{n(d/n - 1/4)^2}\right)^{t/2} + \rho.$$

For brevity, we write $\mathcal{F}_{\mathsf{bit}}$ for $\mathcal{F}'^{(1)}_{\mathsf{copy}}$ below, with the idea that the tampering functions in $\mathcal{F}'^{(1)}_{\mathsf{copy}}$ only allow to keep or flip a bit or to set it to 0 or to 1. More formally, a function $f \in \mathcal{F}_{\mathsf{bit}}$ can be characterized by a vector $\chi(f) = (f_1, \ldots, f_n)$ where $f_i \in \{\mathsf{zero}, \mathsf{one}, \mathsf{keep}, \mathsf{flip}\}$, with the meaning that f takes as input a codeword c and outputs a codeword $c' = (c'[1], \ldots, c'[n])$ in which each bit is either set to 0 (zero), set to 1 (one), left unchanged (keep), or flipped (flip).

For the proof of Theorem 19, fix $q \in \mathbb{N}$ and some distinguisher $\mathbf{D}$. For the remainder of this section, let $\mathcal{F} := \mathcal{F}_{\mathsf{bit}}$, $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}} := \mathbf{S}^{\mathsf{real}}_{\mathcal{F},1,q}$ and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau} := \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau,1,q}$ (for a simulator τ to be determined). For a tamper query $f \in \mathcal{F}$ with $\chi(f) = (f_1, \ldots, f_n)$ issued by $\mathbf{D}$, let $A(f) := \{i \mid f_i \in \{\mathsf{zero}, \mathsf{one}\}\}$, $B(f) := \{i \mid f_i \in \{\mathsf{keep}, \mathsf{flip}\}\}$, and $a(f) := |A(f)|$. Moreover, let $\mathsf{val}(\mathsf{zero}) := \mathsf{val}(\mathsf{keep}) := 0$ and $\mathsf{val}(\mathsf{one}) := \mathsf{val}(\mathsf{flip}) := 1$. Queries f with $0 \le a(f) \le t$, $t < a(f) < n - t$, and $n - t \le a(f) \le n$ are called *low queries*, *middle queries*, and *high queries*, respectively.

Dangerous queries. A tamper query is *dangerous* if it is

- a middle query or
- a low query such that there exists a codeword δ^* of the LECSS with $\forall i \in B(f) : \delta^*[i] = \mathsf{val}(f_i)$ and $\mathsf{D}(\delta^*) \neq 0$.

Consider the hybrid system $\mathbf{H}$ that proceeds as $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$, except that as soon as $\mathbf{D}$ specifies a dangerous query f, $\mathbf{H}$ self-destructs, i.e., answers f and all subsequent queries with $\diamond$.

Lemma 20. $\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{H}) \le \frac{1}{2^t} + \left(\frac{t}{n(d/n-1/4)^2}\right)^{t/2} + \rho.$

Proof. Define a *successful* dangerous query to be a dangerous query that does not decode to $\diamond$. On both systems $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ and $\mathbf{H}$, one can define an MBO $\mathcal{B}$ (cf. Section 2.1) that is provoked if and only if the *first* dangerous query is successful.

Clearly, $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$ and $\mathbf{H}$ behave identically until MBO $\mathcal{B}$ is provoked, thus $\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}} \overset{g}{\equiv} \hat{\mathbf{H}}$, and

$$\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{H}) \le \Gamma^{\mathbf{D}}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}).$$

Towards bounding $\Gamma^{\mathbf{D}}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}})$, note first that adaptivity does not help in provoking $\mathcal{B}$: For any distinguisher $\mathbf{D}$, there exists a *non-adaptive* distinguisher $\mathbf{D}'$ with

$$\Gamma^{\mathbf{D}}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}) \le \Gamma^{\mathbf{D}'}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}). \tag{14}$$

$\mathbf{D}'$ proceeds as follows: First, it (internally) interacts with $\mathbf{D}$ only. Initially, it stores the message x output by $\mathbf{D}$ internally. Then, it handles the tamper queries f by $\mathbf{D}$ as follows:

- *Low query:* If there exists a codeword δ^* of the LECSS with $\forall i \in B(f) : \delta^*[i] = \mathsf{val}(f_i)$ and $\mathsf{D}(\delta^*) = 0$, $\mathbf{D}'$ answers with x. Otherwise, $\mathbf{D}'$ stops its interaction with $\mathbf{D}$ and sends x and all the queries to $\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}$.
- *Middle query:* $\mathbf{D}'$ stops its interaction with $\mathbf{D}$ and sends x and all the queries to $\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}$.

- *High query:* If there exists a codeword c^* that agrees with f in positions i where $f_i \in \{\mathsf{zero}, \mathsf{one}\}$, $\mathbf{D}'$ answers with $\mathrm{Dec}(c^*)$. Otherwise, $\mathbf{D}'$ stops its interaction with $\mathbf{D}$ and sends x and all the queries to $\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}$.

To prove (8), fix all randomness in experiment $\mathbf{D}'\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$, i.e., the coins of $\mathbf{D}$ (inside $\mathbf{D}'$) and the randomness of the encoding (inside $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$). Suppose $\mathbf{D}$ would provoke $\mathcal{B}$ in the direct interaction with $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$. In that case all the answers by $\mathbf{D}'$ are equal to the answers by $\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}$. This is due to the fact that the distance of the LECSS is $d > t$; a successful non-dangerous low query must result in the original message x and a successful high query in $\mathrm{Dec}(c^*)$. Thus, whenever $\mathbf{D}$ provokes $\mathcal{B}$, $\mathbf{D}'$ provokes it as well.

It remains to analyze the success probability of non-adaptive distinguishers $\mathbf{D}'$. Fix the coins of $\mathbf{D}'$; this determines the tamper queries. Suppose there is at least one dangerous query, as otherwise $\mathcal{B}$ is trivially not provoked. The query's success probability can be analyzed as in [17], depending on whether it is a low or a high query, which leads to $\Gamma^{\mathbf{D}'}(\hat{\mathbf{S}}^{\mathsf{real}}_{\mathcal{F}}) \leq \frac{1}{2^t} + \left(\frac{t}{n(d/n-1/4)^2}\right)^{t/2} + \rho$ (recall that the MBO cannot be provoked after an unsuccessful first dangerous query). □

Simulator. The final step of the proof consists of exhibiting a simulator τ such that $\Delta^{\mathbf{D}}(\mathbf{H}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau})$ is small. The indistinguishability proof is facilitated by reusing the two (hardly distinguishable) systems $\mathbf{B}$ and $\mathbf{B}'$ from Section 4 and the wrapper system $\mathbf{W}$ defined in Figure 10, such that $\mathbf{WB} \equiv \mathbf{H}$ and $\mathbf{WB}' \equiv \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$. System $\mathbf{W}$ has an outside interface $\mathtt{o}$ and an inside interface $\mathtt{i}$; at the latter interface, $\mathbf{W}$ expects to be connected to either $\mathbf{B}$ or $\mathbf{B}'$.

Lemma 21. $\mathbf{WB} \equiv \mathbf{H}$.

Proof. Fix a message x. Consider a low query $f = (f_1, \ldots, f_n)$. Let $c = \mathsf{E}(\mathsf{A}(x))$ be an encoding of x, set $c' := f(c)$, and let $\delta' := c + c'$. Using the linearity of the LECSS,

$$\mathsf{D}(c') = \mathsf{D}(\mathsf{E}(\mathsf{A}(x)) + \delta') = \mathsf{A}(x) + \mathsf{D}(\delta').$$

Therefore, $\mathbf{H}$ answers tamper query f by x if $\mathsf{D}(\delta') = 0$ and by $\diamond$ otherwise. In order for δ' to be equal to some codeword δ^* of the LECSS, it is necessary that $\mathsf{val}(f_i) = \delta^*[i]$ for all $i \in B(f)$ and that

$$c[i] + \underbrace{c'[i]}_{\mathsf{val}(f_i)} = \delta^*[i]$$

for all $i \in A(f)$. Note that δ^*, if existent, is unique due to the fact that f is a low query and that the distance of the LECSS is $d > t$.

Similarly, for a high query f, there can be at most one codeword that matches the injected positions. If such a codeword c^* exists, the outcome is $\mathrm{Dec}(c^*)$ if the bits in the keep-positions match c^*, and otherwise $\diamond$.

By inspection, it can be seen that $\mathbf{W}$ acts accordingly. □

Consider now the system $\mathbf{WB}'$. Due to the nature of $\mathbf{B}'$, the behavior of $\mathbf{WB}'$ is independent of the value x that is initially encoded. This allows to easily design a simulator τ as required by Definition 3. The description of τ is given in Figure 11.

Lemma 22. *The simulator τ of Figure 11 satisfies* $\mathbf{WB}' \equiv \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$.

System $\mathbf{W}$

```
init
  ∀i ∈ [n] : c[i] ← λ

on first (encode, x) at o
  output x at i

on (tamper, f) with 0 ≤ a(f) ≤ t at o
  for i where f_i ∈ B(f)
    δ'[i] ← val(f_i)
  if ∃codeword δ*: ∀i ∈ B(f) : δ'[i] = δ*[i]
    for i where f_i ∈ A(f)
      g ← val(f_i) ⊕ δ*[i]
      if c[i] = λ
        output (i, g) at i
        get a ∈ {◇, 1} at i
        if a = ◇
          self-destruct
        c[i] ← g
      else
        if c[i] ≠ g
          self-destruct
    if D(δ*) ≠ 0
      self-destruct
    else
      output x at o
  else
    self-destruct

on (tamper, f) with t < a(f) < n − t at o
  self-destruct

on (tamper, f) with n − t ≤ a(f) ≤ n at o
  for i where f_i ∈ A(f)
    c'[i] ← val(f_i)
  if ∃codeword c* : ∀i ∈ A(f) : c'[i] = c*[i]
    for i where f_i ∈ B(f)
      g ← c*[i] ⊕ val(f_i)
      if c[i] = λ
        output (i, g) at i
        get a ∈ {◇, 1} at i
        if a = ◇
          self-destruct
        c[i] ← g
      else
        if c[i] ≠ g
          self-destruct
    if Dec(c*) = ◇
      self-destruct
    else
      output Dec(c*) at o
  else
    self-destruct
```

Figure 10: The wrapper system $\mathbf{W}$. The command **self-destruct** causes $\mathbf{W}$ to output $\diamond$ at o and to answer all future queries by $\diamond$.

Proof. Consider the systems $\mathbf{WB}'$ and $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$. Both internally choose a vector of n uniform and independent bits $c = (c[1], \ldots, c[n])$. Set $c' := f(c)$, and let $\delta' := c + c'$. System $\mathbf{WB}'$ answers low queries with the value x initially encoded if and only if $\mathsf{D}(\delta') = 0$ and with $\diamond$ otherwise. Simulator τ returns same in the former case, which $\mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}$ replaces by x, and $\diamond$ in the latter case.

Observe that the answer by $\mathbf{WB}'$ to a high query f always matches $\mathrm{Dec}(c'[1], \ldots, c'[n])$, where for $i \in A(f)$, $c'[i] = \mathsf{val}(f_i)$, and for $i \in B(f)$, $c'[i] = c[i] \oplus \mathsf{val}(f_i)$: If no codeword c^* matching the injected positions exists, then $\mathrm{Dec}(c'[1], \ldots, c'[n]) = \diamond$, which is also what $\mathbf{WB}'$ outputs. If such c^* exists and $c^*[i] = c[i] \oplus \mathsf{val}(f_i)$ for all $i \in B(f)$, the output of $\mathbf{WB}'$ is $\mathrm{Dec}(c'[1], \ldots, c'[n])$. If there exists an $i \in B(f)$ with $c^*[i] \neq c[i] \oplus \mathsf{val}(f_i)$, $\mathbf{WB}'$ outputs $\diamond$, and in this case $\mathrm{Dec}(c'[1], \ldots, c'[n]) = \diamond$ since the distance of the LECSS is $d > t$. $\square$

The proof of Theorem 19 now follows from a simple triangle inequality.

Simulator τ

init
  $\forall i \in [n] : c[i] \leftarrow\!\!\$ \{0,1\}$

on (tamper, f) *with* $0 \le a(f) \le t$
  for i *where* $f_i \in A(f)$
    $\delta'[i] \leftarrow \mathsf{val}(f_i) \oplus c[i]$
  for i *where* $f_i \in B(f)$
    $\delta'[i] \leftarrow \mathsf{val}(f_i)$
  $\delta' \leftarrow \delta'[1] \cdots \delta'[n]$
  if $\mathsf{D}(\delta') \neq 0$
    return $\diamond$
  else
    return same

on (tamper, f) *with* $t < a(f) < n - t$
  return $\diamond$

on (tamper, f) *with* $n - t \le a(f) \le n$
  for i *where* $f_i \in A(f)$
    $c'[i] \leftarrow \mathsf{val}(f_i)$
  for i *where* $f_i \in B(f)$
    $c'[i] \leftarrow c[i] \oplus \mathsf{val}(f_i)$
  $c' \leftarrow c'[1] \cdots c'[n]$
  return $\mathrm{Dec}(c')$

Figure 11: Simulator τ.

Proof (of Theorem 19). From Lemmas 20, 5, 21, and 22, one obtains that for all distinguishers $\mathbf{D}$,

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau}) &\le \Delta^{\mathbf{D}}(\mathbf{S}^{\mathsf{real}}_{\mathcal{F}}, \mathbf{H}) + \underbrace{\Delta^{\mathbf{D}}(\mathbf{H}, \mathbf{WB})}_{=0} + \underbrace{\Delta^{\mathbf{D}}(\mathbf{WB}, \mathbf{WB}')}_{=\Delta^{\mathbf{DW}}(\mathbf{B},\mathbf{B}')} + \underbrace{\Delta^{\mathbf{D}}(\mathbf{WB}', \mathbf{S}^{\mathsf{simu}}_{\mathcal{F},\tau})}_{=0} \\ &\le 2^{-t} + \left(\frac{t}{n(d/n - 1/4)^2}\right)^{t/2} + \rho + 2^{-(t-1)} \\ &\le 3 \cdot 2^{-t} + \left(\frac{t}{n(d/n - 1/4)^2}\right)^{t/2} + \rho.\end{aligned}$$

□

Lemma 23. *If* $(\mathrm{Enc}, \mathrm{Dec})$ *is continuously* $(\mathcal{F}'_{\mathsf{copy}}, \varepsilon, 1, q)$*-LOR-non-malleable, it is also continuously* $(\mathcal{F}'_{\mathsf{copy}}, \ell \cdot \varepsilon, \ell, q)$*-LOR-non-malleable, for all* $\ell \in \mathbb{N}$*.*

Proof. The proof is analogous to the proof of Lemma 10, except that the reduction system $\mathbf{C}_i$ computes f'_v as follows:

$$f'_v := \begin{cases} f_v & \text{if } f_v \in \{\mathsf{zero}, \mathsf{one}\}, \\ \mathsf{zero} & \text{if } f_v = \mathsf{copy}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 0, \\ \mathsf{one} & \text{if } f_v = \mathsf{copy}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 1, \\ \mathsf{copy}_1 & \text{if } f_v = \mathsf{copy}_i, \\ \mathsf{one} & \text{if } f_v = \mathsf{flip}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 0, \\ \mathsf{zero} & \text{if } f_v = \mathsf{flip}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 1, \\ \mathsf{flip}_1 & \text{if } f_v = \mathsf{flip}_i. \end{cases}$$

□