On Symmetric Encryption with Distinguishable Decryption Failures*

Alexandra Boldyreva[1], Jean Paul Degabriele[2], Kenneth G. Paterson[2], and Martijn Stam[3]

[1] Georgia Institute of Technology
[2] Royal Holloway, University of London
[3] University of Bristol

Abstract. We propose to relax the assumption that decryption failures are indistinguishable in security models for symmetric encryption. Our main purpose is to build models that better reflect the reality of cryptographic implementations, and to surface the security issues that arise from doing so. We systematically explore the consequences of this relaxation, with some surprising consequences for our understanding of this basic cryptographic primitive. Our results should be useful to practitioners who wish to build accurate models of their implementations and then analyse them. They should also be of value to more theoretical cryptographers proposing new encryption schemes, who, in an ideal world, would be compelled by this work to consider the possibility that their schemes might leak more than simple decryption failures.

1 Introduction

ATTACKS BASED ON DECRYPTION FAILURES. Encryption schemes meeting strong notions of security typically introduce redundancy into their ciphertexts, and as a consequence ciphertexts may be deemed invalid during decryption. A scheme's correctness ensures that honestly generated ciphertexts will always decrypt correctly, hence we expect decryption to 'fail' only for ciphertexts that are corrupted during transmission or are adversarially generated. Typically, protocols making use of an encryption scheme report decryption failures to the sender through error messages, and thus the fact that a decryption failure has occurred becomes known to the adversary. After Bleichenbacher's attack on RSA PKCS#1 [8], it became recognised in the academic community that these decryption failures (and the attendant error messages) may leak significant information to an adversary, undermining schemes' confidentiality properties. Other examples in the asymmetric setting were subsequently discovered [15, 20] and called *reaction attacks*. Vaudenay then showed that similar issues can arise in the symmetric setting [26], and his ideas were extended to produce significant attacks against (among others) SSL/TLS [10, 22], IPsec [11, 12], ASP.NET [13], XML encryption [18] and DTLS [2]. Analysis of error messages in the symmetric setting was also crucial to the success of attacks against the SSH Binary Packet Protocol [1].

THE RELATION BETWEEN ATTACKS AND SECURITY DEFINITIONS. At a very high level the above-mentioned attacks on symmetric schemes have the common feature that during decryption some information about the plaintext is leaked, due to error messages, their timing, or some other aspect of the implementation. The leaked information is normally quite small, and the power of these attacks really comes from the adversary's ability to amplify this leakage through iteration. That is, given a target ciphertext, an adversary is able to produce a sequence of related ciphertexts which when decrypted will leak more information about the target plaintext. If we now compare this to the IND-CCA security model, it appears that such attacks should be fully accounted for and prevented, given the very conservative approach adopted in this model. Indeed, in the IND-CCA model, the adversary is given full access to a decryption oracle for any ciphertext except the target ciphertext, from which he learns either the corresponding plaintext or the fact that decryption fails; and yet this should not leak any information about the target plaintext. Furthermore, several of the attacks above do not even make full use of the decryption oracle, but only consider ciphertexts which result in decryption failures.

Why then are the attacks possible at all? Are the underlying encryption schemes actually IND-CCA secure? Is the IND-CCA model the right one for capturing these classes of attack?

* A short version of this paper was published at FSE 2013. This is the full version.

SSL/TLS makes an instructive case study for answering these questions. At a high level, SSL/TLS most commonly uses a Mac-then-Encrypt (MtE) construction, with either a stream cipher or CBC-mode encryption of a block cipher as the encryption scheme. Thus SSL/TLS is covered by Krawczyk's result [19], and one might reasonably conclude that its symmetric encryption scheme is IND-CCA secure. Yet Canvel et al. [10] presented plaintext-recovering attacks against the OpenSSL implementation of SSL/TLS when CBC-mode is used, in which the attacker does nothing other than submit certain ciphertexts for decryption and analyse the results (i.e. the attacker ostensibly operates within the IND-CCA model). The key point, however, is that at the time of Canvel et al.'s attacks in 2003, it was possible to infer more from SSL/TLS decryption failures than the simple fact that decryption had failed: decryption could fail either because either the underlying padding needed by CBC-mode was incorrectly formatted or because of a MAC failure, and it was possible to tell these conditions apart (either because they were indicated by different error messages or because the error messages were produced at different times during decryption processing). This additional information was sufficient to realise a padding oracle attack, in the style of [26]. Furthermore, this attack is technically *outside* the IND-CCA security model, because this model only ever provides a single decryption failure symbol $\perp$ to the adversary. Thus, while SSL/TLS may be provably IND-CCA secure in theory, it turned out not to be in practice. Suitable countermeasures involve making it hard for an attacker to learn the cause of decryption failures and were incorporated into the TLS specification from version 1.1 onwards. Meanwhile, building an accurate model of SSL/TLS's symmetric encryption scheme and proving its security has turned out to be a complex task that was only recently completed in [22]. Even there, however, it was necessary to assume that all decryption failures are indistinguishable (since, otherwise, attacks like those of [26, 10, 2, 3] are possible). A similar story could be told for MAC-then-encryption configurations of IPsec, to which the theory in [19] and the attacks of [12] both apply.

So the answers to our questions above are, respectively, yes and no. Yes, the underlying encryption schemes *are* provably IND-CCA secure. However, this is for some description of the schemes that may not accurately reflect how they are actually implemented. And no, the standard model for IND-CCA security is not the right one for capturing these attacks: in the current formalism, more specifically the basic syntax adopted for encryption schemes, it is assumed that decryption failures are indistinguishable and that each decryption failure will return the same error symbol $\perp$. This creates a *gap* in the effective power conferred by a decryption oracle between the IND-CCA model and practical attack scenarios (where decryption failures are often distinguishable). In short, knowing *why* decryption failed may be more informative to the adversary than the mere fact that decryption has failed.

OUR CONTRIBUTIONS. We propose to strengthen the existing security definitions for symmetric encryption by letting the adversary distinguish various possible decryption errors. Our main purpose is to build models that better reflect the reality of cryptographic implementations, and to surface the security issues that arise from doing so. We are not the first to make this relaxation (see, for example, [21, 23]), but we are the first to systematically explore its consequences, with some surprising consequences for our understanding of this basic cryptographic primitive. Our results should be useful to practitioners who wish to build accurate models of their implementations and then analyse them. They should also be of value to more theoretical cryptographers proposing new encryption schemes, who, in an ideal world, would be compelled by this work to consider the possibility that their schemes might leak more than simple decryption failures. (Of course, an alternative reaction by the latter group would be to cast this as an implementation issue and simply assume indistinguishable errors as usual; however, the history of attacks tells us that this is hard to guarantee in practice and therefore a dangerous assumption to make.)

Our approach requires the adoption of a slightly different syntax for encryption schemes to the standard one. Now, our decryption algorithm will either return a message from the message space, or an error message from a predetermined finite set of values which we refer to as the *error space*. Technically, then, encryption schemes with multiple errors are a slightly different object from single-error schemes. This approach allows us to handle schemes that can fail in a finite number of distinguishable ways that will be indicated in practice by different error messages. It also enables us to treat attacks in which indistinguishable error messages are returned (perhaps because they are all encrypted, as is the case in

SSL/TLS), but in which the errors are returned at a discrete set of times. We note that our approach is equally applicable to the asymmetric setting; here we will restrict our scope to the symmetric setting only.

With this new syntax in hand, we re-examine the statement due to Bellare and Namprempre [9] that semantic (IND-CPA) security in combination with integrity of ciphertexts (INT-CTXT) is sufficient to imply chosen ciphertext (IND-CCA) security. One consequence of their results is that 'IND-CPA + INT-CTXT' has come to be seen as the 'right' security notion to aim for in the symmetric case, with this combined notion now being referred to as *authenticated-encryption security*. This seems to be mostly because it implies IND-CCA security, and because that is by now the accepted notion in the asymmetric setting. We show, through separations, that this important relation no longer holds for multiple error symmetric encryption schemes. Indeed, it is easy to see where the proof of this relation in [9] breaks down: in the passage from the INT-CTXT security game to the IND-CPA security game, the simulation in [9] simply replies to all decryption queries with the error message $\perp$; only if an adversary forges a ciphertext does this simulation go awry. But this is not an accurate response in the multiple error setting, since one of several possible error messages should be returned, and the simulation does not necessarily know which.

We then go on to establish relations that are similar in spirit to the classic relations, in that they combine a weak form of confidentiality with some form of ciphertext integrity to obtain strong confidentiality. An interesting aspect that emerges in our analysis is that it is not at all obvious how the notion of ciphertext integrity should be extended to the multiple-error setting. We identify two candidate definitions for ciphertext integrity, one being strictly stronger than the other. We compare and contrast the two, and provide evidence (by means of a rather non-trivial counterexample) for requiring the stronger variant in our relations.

We also provide a natural extension of the IND-CCA3 security notion to the multiple-error setting. This notion, due to Rogaway and Shrimpton [25], is an elegant combination of semantic security and ciphertext integrity into a single equivalent security notion. We show that it serves as a good security notion for symmetric encryption with multiple errors. More specifically we show that our extension to IND-CCA3 security does imply chosen-ciphertext security in the multiple error setting.

We conclude by showing that the encode-then-encrypt-then-MAC (EEM) construction is IND-CCA secure for any encoding scheme, any IND-CPA secure encryption scheme with arbitrary error messages, and any SUF-CMA MAC. Following the works of Bellare and Namprempre [9] and Krawczyk [19], this result provides further formal grounds for preferring the EEM composition over other generic constructions, for example MAC-then-encrypt.

In addition to the standard symmetric encryption notions, we provide equivalent results for security definitions involving indistinguishability from random bits introduced by Rogaway [24], and for the stateful setting introduced by Bellare, Kohno, and Namprempre [7]. Many of these additional results follow rather straightforwardly, but we consider it valuable to include them for completeness.

2 Preliminaries

2.1 Notation

Unless otherwise stated, an algorithm may be randomized. An adversary is an algorithm. For any algorithm $\mathcal{A}$ we use $y \leftarrow \mathcal{A}(x_1, x_2, \dots)$ to denote executing $\mathcal{A}$ with fresh coins on inputs $x_1, x_2, \dots$ and assigning its output to y. If $\mathcal{S}$ is a set then $|\mathcal{S}|$ denotes its size, and $y \leftarrow\!\!\$\ \mathcal{S}$ denotes the process of selecting an element from $\mathcal{S}$ uniformly at random and assigning it to y. The set of all finite binary strings is denoted by $\{0,1\}^*$, for any positive integer n and bit b, we denote by b^n the string of n consecutive b's and $\{0,1\}^n$ represents the set of all binary strings of length n. The empty string is represented by ε. For any two strings w and z and a positive integer i, $w \parallel z$ denotes their concatenation, $w \oplus z$ denotes their bitwise XOR, $|w|$ denotes the length of w, and $w[i]$ denotes the i^{th} bit of w. If j is a non-negative integer, then $\langle j \rangle_\ell$ denotes the unsigned ℓ-bit binary representation of j. Accordingly $\langle \cdot \rangle^{-1}$ represents the inverse mapping which maps strings of any length to $\mathbb{N}$. If w is an ℓ-bit string and i is an integer we use $w + i$ as

shorthand for $\langle\langle w\rangle^{-1} + i \bmod 2^\ell\rangle_\ell$. We use $\mathrm{Func}(\mathcal{X}, \mathcal{Y})$ to denote the set of all functions with domain $\mathcal{X}$ and codomain $\mathcal{Y}$. We will often have that $\mathcal{X} = \{0,1\}^\ell$ or $\mathcal{X} = \{0,1\}^*$, and $\mathcal{Y} = \{0,1\}^n$ for some positive integers ℓ and n. Accordingly we abbreviate notation for the corresponding sets of functions to $\mathrm{Func}(\ell, n)$ and $\mathrm{Func}(*, n)$ respectively.

2.2 Building Blocks

PSEUDORANDOM FUNCTIONS. A *function family* is a map $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We refer to $\mathcal{K}$ as the key space of F, $\mathcal{X}$ as the domain of F, and $\mathcal{Y}$ as the codomain of F. In this paper $\mathcal{K}$, $\mathcal{X}$, and $\mathcal{Y}$ will be sets of bit-strings. For eack $K \in \mathcal{K}$ we define the map $F_K : \mathcal{X} \to \mathcal{Y}$ by $F_K(x) = F(K, x)$ for all $x \in \mathcal{X}$. Thus F can be seen as a collection of maps from $\mathcal{X}$ to $\mathcal{Y}$, each identified by some key in $\mathcal{K}$. We will refer to F_K as an instance of F. We will often make use of function families that are *pseudorandom*.

Definition 1 (Pseudorandom functions). *Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a function family. Consider an adversary $\mathcal{A}$ with oracle access to some function with domain $\mathcal{X}$ and codomain $\mathcal{Y}$, that returns a single bit as its output. We define the prf-advantage of adversary $\mathcal{A}$ with respect to the function family F as:*

$$\mathbf{Adv}^{\mathrm{prf}}_F(\mathcal{A}) = \mathbf{Pr}\left[K \leftarrow\!\!\$\, \mathcal{K} : \mathcal{A}^{F_K(\cdot)} = 1\right] - \mathbf{Pr}\left[f \leftarrow\!\!\$\, \mathrm{Func}(\mathcal{X}, \mathcal{Y}) : \mathcal{A}^{f(\cdot)} = 1\right].$$

F is said to be a pseudorandom function (PRF), if for every adversary $\mathcal{A}$ with reasonable resources its prf-advantage $\mathbf{Adv}^{\mathrm{prf}}_F(\mathcal{A})$ is small.

MACS. A *message authentication code (MAC)* $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ with associated error space $\mathcal{Q}_\perp$ consists of three algorithms. The randomized *key-generation* algorithm $\mathcal{K}$ takes no input and returns a secret key K. We will sometimes abuse notation and regard $\mathcal{K}$ as a set of keys. The *tagging* algorithm $\mathcal{T}$ may be randomized or stateful. It takes as input the secret key K and a message $m \in \{0,1\}^*$ to return a tag τ. The *verification* algorithm $\mathcal{V}$ is deterministic and stateless. It takes the secret key K, a message $m \in \{0,1\}^*$ and a candidate tag τ, and returns either 1 or an error message in $\mathcal{Q}_\perp$. We require that for all K that can be output by $\mathcal{K}$ and all $m \in \{0,1\}^*$, it hold (with probability 1) that if $\tau \leftarrow \mathcal{T}_K(m)$ then $\mathcal{V}_K(m, \tau) = 1$. Here, we allow multiple possible error messages for $\mathcal{MA}$ in order to be able to model certain types of attack, e.g. that in [3].

The standard security notion for MACs is existential unforgeability under chosen message attacks (UF-CMA). We will however require a stronger variant of this notion (SUF-CMA) which is defined below.

Definition 2 (SUF-CMA). *Let $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ be a message authentication code with associated error space $\mathcal{Q}_\perp$. For an adversary $\mathcal{A}$, define experiment $\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{MA}}(\mathcal{A})$ as shown in Figure 1. A key K is first generated by calling $\mathcal{K}$. The adversary $\mathcal{A}$ is then given access to a tagging oracle $\mathsf{Tag}(\cdot)$ and a verification oracle $\mathsf{Ver}(\cdot,\cdot)$. The adversary wins if it queries a valid message-tag pair that was not previously returned by the tagging oracle. We define the adversary's advantage as:*

$$\mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(\mathcal{A}) = \Pr\left[\,\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{MA}}(\mathcal{A})\,\right].$$

The scheme $\mathcal{MA}$ is said to be SUF-CMA *secure if, for every adversary $\mathcal{A}$ consuming reasonable resources its advantage $\mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(\mathcal{A})$ is small.*

The standard UF-CMA notion is defined analogously but the adversary is only granted a win if it forges a tag for a message that was not previously queried to the tagging oracle.

ENCODING SCHEMES. When constructing symmetric encryption schemes from other components it is common to perform some form of preprocessing on the message. Its purpose may be to map messages to the message space of the encryption scheme, or as an attempt to extend the scheme's functionality, such as masking the message length. Generally such transformations are unkeyed, but may be randomized. We model such transformations by encoding schemes.

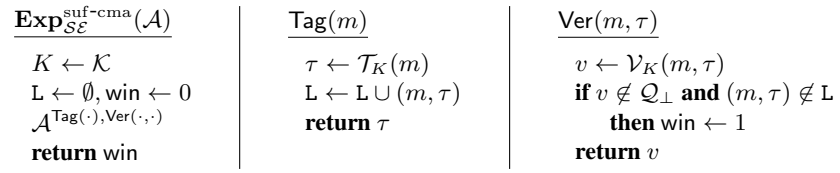


Fig. 1. SUF-CMA experiment for message authentication codes.

An *encoding scheme* $\mathcal{ES} = (\mathcal{EC}, \mathcal{DC})$ consists of two algorithms and associated domain, codomain, and an error space. The *encoding* algorithm $\mathcal{EC}$ which may be randomized, takes as input a string from its domain and maps it to some string in its codomain. The *decoding* algorithm $\mathcal{DC}$ is deterministic and takes a string from its codomain and returns either a string in its domain or an error symbol from its error space. The scheme must be correct, i.e. for every string m in its domain it holds that $\mathcal{DC}(\mathcal{EC}(m)) = m$. An encoding scheme is *length-regular* if for any two strings m and m' in its domain, it holds that if $|m| = |m'|$ then $|\mathcal{EC}(m)| = |\mathcal{EC}(m')|$.

3 Symmetric Encryption with Multiple Errors: Definitions

SYNTAX. A *symmetric encryption scheme* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with associated message space $\mathcal{M} \subseteq \{0,1\}^*$, ciphertext space $\mathcal{C} \subseteq \{0,1\}^*$, and error space $\mathcal{S}_\perp$ consists of three algorithms. The randomized *key-generation* algorithm $\mathcal{K}$ takes no input and returns a secret key K, an initial encryption state σ_0, and an initial decryption state ϱ_0. We will sometimes abuse notation and regard $\mathcal{K}$ as a set of keys. The randomized and stateful *encryption* algorithm $\mathcal{E} : \mathcal{K} \times \mathcal{M} \times \Sigma \to \mathcal{C} \times \Sigma$ takes as input the secret key $K \in \mathcal{K}$, a plaintext $m \in \mathcal{M}$, and the current encryption state $\sigma \in \Sigma$, and returns a ciphertext in $\mathcal{C}$ together with an updated state. The deterministic and stateful *decryption* algorithm $\mathcal{D} : \mathcal{K} \times \mathcal{C} \times \Sigma \to (\mathcal{M} \cup \mathcal{S}_\perp) \times \Sigma$ takes as input the secret key K, a ciphertext $c \in \mathcal{C}$, and the current decryption state ϱ to return the corresponding plaintext $m \in \mathcal{M}$ or a special symbol from $\mathcal{S}_\perp$ (indicating that the ciphertext is invalid) and an updated state.

Our syntax of symmetric encryption schemes differs in two main ways from the more conventional way of modelling symmetric encryption schemes. Firstly it allows the decryption algorithm to indicate invalid ciphertexts with distinct error messages within the error space. We will assume the error space be a set of symbols $\{\perp_1, \perp_2, \ldots, \perp_n\}$ for some positive integer n. The symbol $\perp$ will be used interchangeably to denote a specific error symbol or a variable assuming values from the error space. We will use the term *multiple-error encryption scheme* to indicate schemes with an error space of size strictly greater than one. Secondly we adopt a stateful syntax for both encryption and decryption. This is without loss of generality. Both encryption and decryption can be made stateless by defining $\mathcal{K}$ to always return the empty string for the corresponding initial state, and having $\mathcal{E}, \mathcal{D}$ ignore (i.e. never update) the state.

For any $\ell \in \mathbb{N}$ and any $\mathbf{m} = [m_1, \ldots, m_\ell] \in \mathcal{M}^\ell$, we write $(\mathbf{c}, \sigma) \leftarrow \mathcal{E}_K(\mathbf{m}, \sigma_0)$ as shorthand for $(c_1, \sigma_1) \leftarrow \mathcal{E}_K(m_1, \sigma_0)$, $(c_2, \sigma_2) \leftarrow \mathcal{E}_K(m_2, \sigma_1), \ldots (c_\ell, \sigma_\ell) \leftarrow \mathcal{E}_K(m_\ell, \sigma_{\ell-1})$, where $\mathbf{c} = [c_1, \ldots, c_\ell]$ and $\sigma = \sigma_\ell$. Similarly we use $(\mathbf{m}', \varrho) \leftarrow \mathcal{D}_K(\mathbf{c}, \varrho_0)$ to denote the analogous process for decryption. Finally, we require that a symmetric encryption scheme satisfy *correctness* which is defined as follows:

Definition 3 (Correctness of $\mathcal{SE}$). *For all (K, σ_0, ϱ_0) that can be output by $\mathcal{K}$, all $\ell \in \mathbb{N}$, and all $\mathbf{m} \in \mathcal{M}^\ell$, it holds (with probability 1) that if $(\mathbf{c}, \sigma) \leftarrow \mathcal{E}_K(\mathbf{m}, \sigma_0)$ and $(\mathbf{m}', \varrho) \leftarrow \mathcal{D}_K(\mathbf{c}, \varrho_0)$, then $\mathbf{m}' = \mathbf{m}$.*

INDISTINGUISHABILITY NOTIONS. We adopt the 'left-or-right' model of indistinguishability from Bellare et al. [5] to define three notions of confidentiality for symmetric encryption. Indistinguishability under chosen-plaintext attack (IND-CPA), and indistinguishability under chosen-ciphertext attack (IND-CCA) are fairly standard, except for the fact that for multiple-error schemes the decryption oracle will now return one of many possible error messages. We introduce the notion of indistinguishability under ciphertext-validity attack (IND-CVA), which can be seen as a *strengthened* adaption of a similar notion

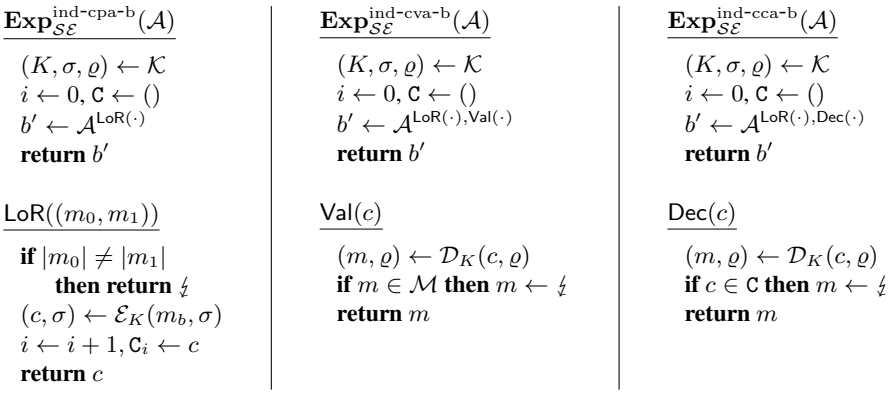


Fig. 2. IND-ATK experiments for symmetric encryption schemes.

defined by Bauer et al. [4] to the symmetric setting. Here, in addition to an encryption oracle the adversary is given access to a *ciphertext-validity* oracle which indicates whether a ciphertext is valid or not, and if not, *returns the exact error message* output by the decryption algorithm.

Definition 4 (IND-ATK security). *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. For an adversary $\mathcal{A}$ and a bit b, define the experiments $\mathbf{Exp}^{\text{ind-atk-b}}_{\mathcal{SE}}(\mathcal{A})$ where $\text{atk} \in \{\text{cpa}, \text{cva}, \text{cca}\}$ as shown in Figure 2. In all three experiments, a key K is first generated by calling $\mathcal{K}$. The adversary $\mathcal{A}$ is then given access to a left-or-right encryption oracle $\mathsf{LoR}(\cdot)$, and possibly a ciphertext-validity oracle $\mathsf{Val}(\cdot)$ or a decryption oracle $\mathsf{Dec}(\cdot)$. No restriction is imposed on the adversary's queries, rather if it queries a pair of messages of unequal length to $\mathsf{LoR}(\cdot)$, or if it queries a ciphertext to $\mathsf{Dec}(\cdot)$ previously returned by $\mathsf{LoR}(\cdot)$, the $\lightning$ symbol is returned. In the $\mathsf{Val}(\cdot)$ oracle the $\lightning$ symbol indicates that the queried ciphertext was valid.*

The adversary's goal is to output a bit b', as its guess of the challenge bit b, and the experiment returns b' as well. For each of these three experiments we define the corresponding advantages of an adversary $\mathcal{A}$ as:

$$\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{SE}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{ind-atk-1}}_{\mathcal{SE}}(\mathcal{A}) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-atk-0}}_{\mathcal{SE}}(\mathcal{A}) = 1\right].$$

The scheme $\mathcal{SE}$ is said to be IND-ATK *secure, if for every adversary $\mathcal{A}$ with reasonable resources its advantage $\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{SE}}(\mathcal{A})$ is small.*

INDISTINGUISHABILITY FROM RANDOM BITS. We can recast the above three security notions in terms of indistinguishability from random bits as introduced by Rogaway [24]. Here the adversarial goal is to distinguish encrypted messages from random bit-strings of the same length.

Definition 5 (IND$-ATK security). *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. For an adversary $\mathcal{A}$ and a bit b, define the experiments $\mathbf{Exp}^{\text{ind\$-atk-b}}_{\mathcal{SE}}(\mathcal{A})$ where $\text{atk} \in \{\text{cpa}, \text{cva}, \text{cca}\}$ as shown in Figure 3. In all three experiments, a key K is first generated by calling $\mathcal{K}$. The adversary $\mathcal{A}$ is then given access to a special encryption oracle $\mathsf{Enc\$}(\cdot)$, if $b = 1$ the oracle returns the encrypted message, otherwise it returns a uniformly-random bit-string of the same length. In the* ind$-cva *and* ind$-cca *experiments, the adversary is additionally given access to a ciphertext-validity oracle $\mathsf{Val}(\cdot)$ and a decryption oracle $\mathsf{Dec}(\cdot)$ respectively. Trivial-win conditions are avoided by having the decryption oracle return $\lightning$ in response to any ciphertext that was previously output by the encryption oracle. The ciphertext-validity oracle uses $\lightning$ to indicate that the queried ciphertext was valid or has been previously output by the encryption oracle.*

The adversary's goal is to output a bit b', as its guess of the challenge bit b, and the experiment returns b' as well. For each of these three experiments we define the corresponding advantages of an adversary $\mathcal{A}$ as:

$$\mathbf{Adv}^{\text{ind\$-atk}}_{\mathcal{SE}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{ind\$-atk-1}}_{\mathcal{SE}}(\mathcal{A}) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind\$-atk-0}}_{\mathcal{SE}}(\mathcal{A}) = 1\right].$$

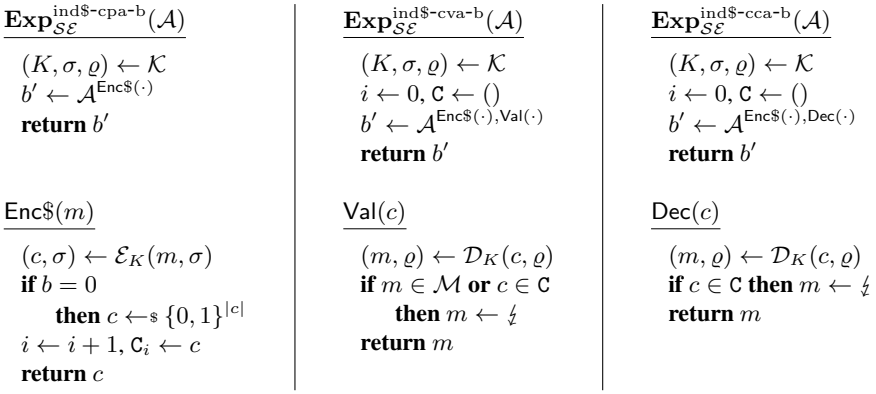


Fig. 3. IND$-ATK experiments for symmetric encryption schemes.

The scheme $\mathcal{SE}$ is said to be IND$-ATK *secure, if for every adversary $\mathcal{A}$ with reasonable resources its advantage $\mathbf{Adv}^{\text{ind\$-atk}}_{\mathcal{SE}}(\mathcal{A})$ is small.*

STATEFUL INDISTINGUISHABILITY NOTIONS. Secure protocols like SSH, SSL/TLS and IPsec aim to protect against replay and reordering of ciphertexts. These security goals are not captured by any of the above security notions. Bellare, Kohno, and Namprempre [7] introduced a notion called IND-sfCCA. This notion implies IND-CCA security and additionally protects against replay and reordering of ciphertexts. We recall this notion and introduce natural variants in terms of indistinguishability from random bits and ciphertext-validity attacks. Of course, our definitions are also for the setting of multiple errors. In what follows we will classify the adversary's decryption queries to be *in-sync*, if the sequence of queried ciphertexts is a prefix of the sequence of ciphertexts returned by the encryption oracle. Accordingly we refer to the first decryption query (and any subsequent one) for which this is no longer true as an *out-of-sync* query.

Definition 6 (Stateful indistinguishability). *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. For an adversary $\mathcal{A}$ and a bit b, define experiments $\mathbf{Exp}^{\text{ind-sfcca-b}}_{\mathcal{SE}}(\mathcal{A})$ and $\mathbf{Exp}^{\text{ind\$-atk-b}}_{\mathcal{SE}}(\mathcal{A})$ where $\text{atk} \in \{\text{sfcva}, \text{sfcca}\}$ as shown in Figure 4. In all three experiments, a key K is first generated by calling $\mathcal{K}$. In the* ind-sfcca *experiment the adversary is given access to a left-or-right encryption oracle* $\mathsf{LoR}(\cdot)$*, and a stateful decryption oracle* $\mathsf{sfDec}(\cdot)$*. The stateful decryption oracle returns the decrypted ciphertexts only for out-of-sync queries, and returns $\lightning$ otherwise. Similarly in the* ind$-atk *experiments the adversary is given access to the special encryption oracle* $\mathsf{Enc\$}(\cdot)$*, and either a stateful ciphertext-validity oracle* $\mathsf{sfVal}(\cdot)$ *or a stateful decryption oracle* $\mathsf{sfDec}(\cdot)$*.*

The adversary's goal is to output a bit b', as its guess of the challenge bit b, and the experiment returns b' as well. For each of these three experiments we define the corresponding advantages of an adversary $\mathcal{A}$ as:

$$\mathbf{Adv}^{\text{ind-sfcca}}_{\mathcal{SE}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{ind-sfcca-1}}_{\mathcal{SE}}(\mathcal{A}) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-sfcca-0}}_{\mathcal{SE}}(\mathcal{A}) = 1\right]$$
$$\mathbf{Adv}^{\text{ind\$-atk}}_{\mathcal{SE}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{ind\$-atk-1}}_{\mathcal{SE}}(\mathcal{A}) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind\$-atk-0}}_{\mathcal{SE}}(\mathcal{A}) = 1\right].$$

The scheme $\mathcal{SE}$ is said to be IND-sfCCA *or* IND$-ATK *secure, if for every adversary $\mathcal{A}$ with reasonable resources its respective advantage $\mathbf{Adv}^{\text{ind-sfcca}}_{\mathcal{SE}}(\mathcal{A})$ or $\mathbf{Adv}^{\text{ind\$-atk}}_{\mathcal{SE}}(\mathcal{A})$ is small.*

The naming of these notions is partly justified by the fact that the decryption and ciphertext-validity oracles are stateful. In addition, it is easy to see that for an encryption scheme to be IND-sfCCA or IND$-sfCCA secure, its decryption algorithm must be stateful. However, a scheme need not have a stateful decryption algorithm to be IND$-sfCVA secure. As the reader may have noticed, we did not define an IND-sfCVA notion. This is because in the presence of a left-or-right encryption oracle, the $\mathsf{sfVal}(\cdot)$ oracle reduces to a $\mathsf{Val}(\cdot)$ oracle, and therefore IND-sfCVA (defined in the obvious way) is equivalent to IND-CVA.

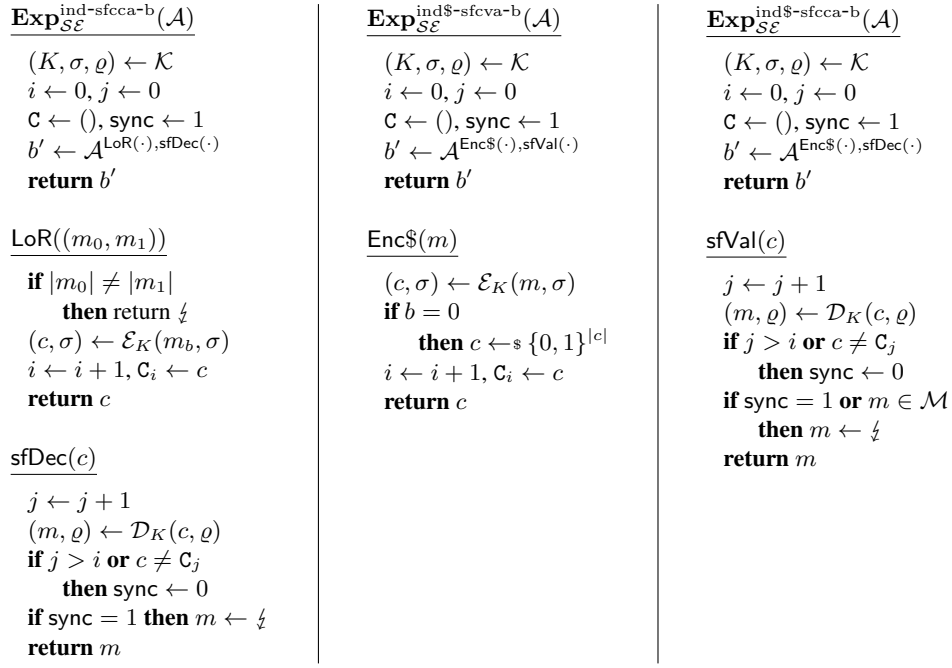


Fig. 4. Stateful indistinguishability experiments for symmetric encryption schemes.

CIPHERTEXT INTEGRITY. We define ciphertext integrity analogously to Bellare and Namprempre [9], and we also consider its stateful variant [7] which additionally protects against replay and reordering attacks. Here an adversary trying to forge a ciphertext is granted multiple attempts by giving it access to a verification oracle $\mathsf{Try}(\cdot)$, in addition to a standard encryption oracle. When extending these notions to schemes with multiple errors, it is not clear how to interpret the verification oracle's functionality. That is, should the verification oracle indicate only whether a ciphertext is valid or not, or should it additionally return the exact error message output by the decryption algorithm if the ciphertext is invalid? For single-error schemes the two interpretations are equivalent, but this does not hold in general (see Section 4). For each of the standard and stateful notions we consider both variants and we denote the weaker variant (i.e. the one that is less informative to the adversary) with '$*$'. In what follows we classify verification queries to be in-sync or out-of-sync in an analogous manner as we did for decryption.

Definition 7 (Ciphertext Integrity). *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. For an adversary $\mathcal{A}$ define the experiments $\mathbf{Exp}^{\text{int-atk}}_{\mathcal{SE}}(\mathcal{A})$ where $\text{atk} \in \{\text{ctxt}, \text{ctxt}*, \text{sfctxt}, \text{sfctxt}*\}$ as shown in Figure 5. In all experiments, a key K is first generated by calling $\mathcal{K}$. The adversary $\mathcal{A}$ is then given access to an encryption oracle $\mathsf{Enc}(\cdot)$, and one of the following verification oracles $\mathsf{Try}(\cdot), \mathsf{Try}^*(\cdot), \mathsf{sfTry}(\cdot)$, or $\mathsf{sfTry}^*(\cdot)$. The $\mathsf{Try}^*(\cdot)$ oracle (and similarly the $\mathsf{sfTry}^*(\cdot)$ oracle) returns $\lightning$ if the queried ciphertext is valid, or if the ciphertext has been previously output by the encryption oracle (respectively: if the verification query is in-sync), and returns $\perp$ if the ciphertext is invalid. The $\mathsf{Try}(\cdot)$ and $\mathsf{sfTry}(\cdot)$ oracles operate analogously but return the exact error message output by the decryption oracle when a ciphertext is invalid.*

In the int-ctxt *and* int-ctxt$*$ *experiments the adversary's goal is to make a valid verification query not previously output by the encryption oracle. In the* int-sfctxt *and* int-sfctxt$*$ *experiments the adversary's goal is to make a valid out-of-sync verification query. In all cases the experiment outputs a bit indicating the adversary's success. For each experiment we define the advantage of an adversary $\mathcal{A}$ as:*

$$\mathbf{Adv}^{\text{int-atk}}_{\mathcal{SE}}(\mathcal{A}) = \Pr\left[\, \mathbf{Exp}^{\text{int-atk}}_{\mathcal{SE}}(\mathcal{A}) = 1 \,\right].$$

The scheme $\mathcal{SE}$ is said to be INT-ATK *secure, if for every adversary $\mathcal{A}$ with reasonable resources its advantage $\mathbf{Adv}^{\text{int-atk}}_{\mathcal{SE}}(\mathcal{A})$ is small.*

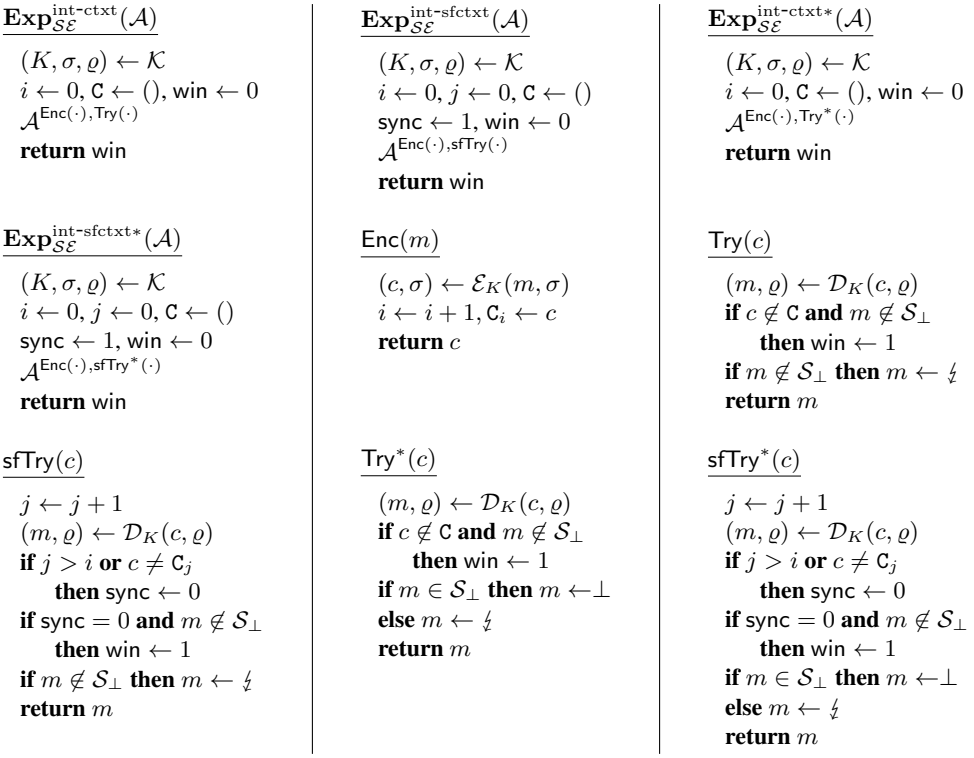


Fig. 5. Ciphertext integrity experiments for symmetric encryption schemes.

ERROR INVARIANCE. Although an encryption scheme may have multiple error messages, not all error messages may be 'available' to the adversary. In particular an adversary may not be able to produce (invalid) ciphertexts that generate all possible error messages. We introduce a simple security notion that captures exactly this situation. Informally an encryption scheme is *error-invariant* if no efficient adversary can generate more than one of the possible error messages. Of course any single-error scheme is trivially error invariant.

Definition 8 (INV-ERR security). *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme with error space $\mathcal{S}_\perp$. For any $\perp \in \mathcal{S}_\perp$ and an adversary $\mathcal{A}$, define the experiment $\mathbf{Exp}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A})$ as shown in Figure 6. A key K is first generated by calling $\mathcal{K}$. The adversary $\mathcal{A}$ is then given access to an encryption oracle $\mathsf{Enc}(\cdot)$ and a decryption oracle $\mathsf{Dec}(\cdot)$.*

The adversary's goal is to submit a ciphertext to the decryption oracle which results in an error message not equal to $\perp$. The experiment outputs a bit indicating the adversary's success. We define the advantage of an adversary $\mathcal{A}$ with respect to $\perp$ as:

$$\mathbf{Adv}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A}) = \Pr\left[\, \mathbf{Exp}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A}) = 1 \,\right].$$

The scheme $\mathcal{SE}$ is said to be INV-ERR *secure if there exists a unique $\perp \in \mathcal{S}_\perp$ such that for every adversary $\mathcal{A}$ with reasonable resources its advantage $\mathbf{Adv}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A})$ is small.*

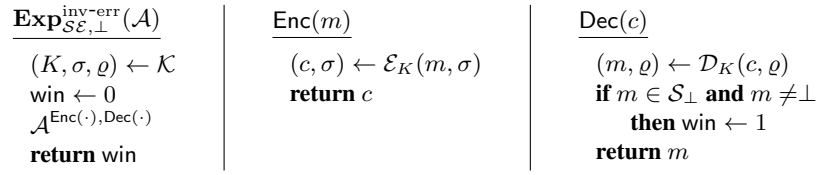


Fig. 6. INV-ERR experiment for symmetric encryption schemes.

ADDITIONAL NOTES. The reader may be wondering how exactly to interpret the $\not\downarrow$ symbol, given that we assign to it different meanings in our security definitions. In general we use it to 'suppress' certain outputs from an oracle, and hence limit the information conveyed by the oracle to the adversary. We use it to avoid trivial win conditions by suppressing the output of in-sync decryption queries, or left-or-right queries containing messages of different lengths. We also use it to define ciphertext-validity and verification oracles by suppressing any plaintext that is output by the decryption algorithm.

For each security definition we have defined the corresponding advantage of an adversary with respect to some cryptographic scheme. We will sometimes refer to the *maximum* advantage with respect to a cryptographic scheme over all adversaries consuming reasonable resources. Any advantage not parametrized by an adversary is to be interpreted this way.

4 Relations and Separations

INTERPRETING OUR IMPLICATIONS AND SEPARATIONS. An implication from security notion X to security notion Y, indicated by $\mathsf{X} \longrightarrow \mathsf{Y}$, means that any scheme which is X-secure is also Y-secure. More formally there exists a constant $\kappa > 0$ such that for any symmetric encryption scheme $\mathcal{SE}$ and any Y adversary $\mathcal{A}_y$ there exists a X adversary $\mathcal{A}_x$ (with similar resources) such that:

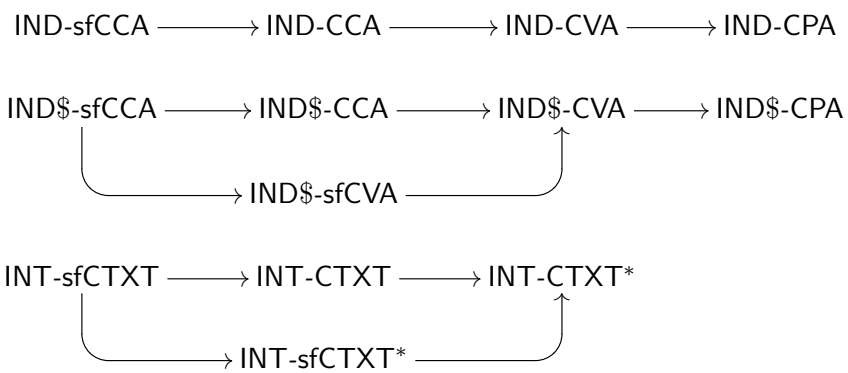
$$\mathbf{Adv}^{\mathrm{y}}_{\mathcal{SE}}(\mathcal{A}_y) \leq \kappa \cdot \mathbf{Adv}^{\mathrm{x}}_{\mathcal{SE}}(\mathcal{A}_x)$$

A separation from security notion X to security notion Y indicated by $\mathsf{X} \not\longrightarrow \mathsf{Y}$, means that there exists a symmetric encryption scheme which meets notion X but for which we can exhibit an attack showing that it does not meet notion Y. The separation is interesting only if there exists some scheme which meets security notion X, as otherwise the implication $\mathsf{X} \longrightarrow \mathsf{Y}$ is vacuously true. Our separations can be categorised into two types. In the former we will assume that there exists some scheme $\mathcal{SE}$ which meets notion X, and use it to construct a scheme $\overline{\mathcal{SE}}$ which meets notion X but is insecure in the Y sense. From the foregoing discussion, such an assumption is in some sense minimal. In the second type of separations we will assume the existence of pseudorandom functions and UF-CMA MACs to construct a scheme which meets notion X but not notion Y. In this paper for all separations of the latter type we will have that $\mathsf{X} \longrightarrow$ IND-CPA. It is a well-known result that the existence of IND-CPA-secure symmetric encryption implies the existence of pseudorandom functions [17, 16, 14]. In addition a pseudorandom function can be combined with an almost-universal hash function to obtain a variable-input-length pseudorandom function, which in turn yields a UF-CMA MAC. Thus from a theoretical viewpoint the underlying assumptions for either type of separation are equivalent.

Note that when proving a separation we do not require the scheme to have distinct error messages, as we are interested solely in the *existence* of a counterexample showing that the relation under question cannot be established. Secondly any multiple-error scheme which is secure under some notion X implies the existence of a single-error scheme which is also secure under notion X (simply by mapping all error messages to a single error message). Consequently it is best to prove separations using schemes with an error space of *minimal cardinality*. It then follows that the separation also holds for all schemes of higher error-space cardinality.

STRAIGHTFORWARD RELATIONS. The following set of relations are self-evident. We state them here for the sake of completeness without proofs.

Proposition 9.



REVISITING CLASSIC RELATIONS. If a symmetric encryption scheme having a single error symbol satisfies both passive confidentiality (IND-CPA) and integrity of ciphertexts (INT-CTXT), then a result of [9] guarantees that it also offers confidentiality against chosen-ciphertext attacks. An analogous result for the stateful setting was proved in [7]. Often, when analysing a particular scheme, its chosen-plaintext security and ciphertext integrity are proved first, and then these classic results are used to guarantee chosen-ciphertext security. Indeed, the combination of IND-CPA and INT-CTXT (or their stateful versions) has come to be the accepted security notion for symmetric encryption. We proceed to re-examine these relations from [9, 7] in the context of encryption schemes with multiple error messages.

The following theorem serves as the basis for the two separations in Corollaries 11 and 12, showing that the classic relations no longer hold for multiple-error schemes. Its proof is in Appendix A.1. We point out that in proving the separations, we adopt the stronger interpretations of ciphertext integrity so as to make the results as strong as possible.

Theorem 10 (IND-CPA $\wedge$ INT-sfCTXT$\not\longrightarrow$ IND-CCA)**.** *Let $F : \mathcal{K}_e \times \{0,1\}^\ell \to \{0,1\}^n$ be a pseudorandom function, and let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a* UF-CMA *secure MAC with tag length $\ell_{tag} < n$. Consider the stateful symmetric encryption scheme $\mathcal{SE}_1$ having message space $\{0,1\}^{n-\ell_{tag}}$ and error space $\{\perp_0, \perp_1\}$ shown in Figure 7. For any* IND-CPA *adversary $\mathcal{A}_{cpa}$ and any* INT-sfCTXT *adversary $\mathcal{A}_{int}$ against $\mathcal{SE}_1$, both making at most $2^\ell - 1$ encryption queries, there exist two corresponding adversaries $\mathcal{A}_{prf}$ and $\mathcal{A}_{uf}$ using roughly the same resources as $\mathcal{A}_{cpa}$ and $\mathcal{A}_{int}$, respectively, such that:*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}_1}(\mathcal{A}_{cpa}) \leq 2 \cdot \mathbf{Adv}^{\text{prf}}_{F}(\mathcal{A}_{prf}) \,, \tag{1a}$$

$$\mathbf{Adv}^{\text{int-sfctxt}}_{\mathcal{SE}_1}(\mathcal{A}_{int}) \leq \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{MA}}(\mathcal{A}_{uf}) \,. \tag{1b}$$

Moreover there exist efficient adversaries $\mathcal{A}_{cca}$ and $\mathcal{A}'_{uf}$ such that:

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE}_1}(\mathcal{A}_{cca}) = 1 - \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{MA}}(\mathcal{A}'_{uf}) \,. \tag{1c}$$

Combining Theorem 10 and Proposition 9 yields the following two separations corresponding to the aforementioned relations from [9] and [7].

Corollary 11 (IND-CPA $\wedge$ INT-CTXT$\not\longrightarrow$ IND-CCA)**.** *Let $F : \mathcal{K}_e \times \{0,1\}^\ell \to \{0,1\}^n$ be a pseudorandom function, and let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a* UF-CMA *secure MAC with tag length $\ell_{tag} < n$. Then there exists a symmetric encryption scheme that is both* IND-CPA *secure and* INT-CTXT *secure but that is not secure in the* IND-CCA *sense.*

Corollary 12 (IND-CPA $\wedge$ INT-sfCTXT$\not\longrightarrow$ IND-sfCCA)**.** *Let $F : \mathcal{K}_e \times \{0,1\}^\ell \to \{0,1\}^n$ be a pseudorandom function, and let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a* UF-CMA *secure MAC with tag length $\ell_{tag} < n$. Then there exists a symmetric encryption scheme that is both* IND-CPA *secure and* INT-sfCTXT *secure but that is not secure in the* IND-sfCCA *sense.*

Algorithm $\mathcal{K}$

$K_e \leftarrow\!\!\$\ \mathcal{K}_e$
$K_m \leftarrow \mathcal{K}_m$
$\sigma \leftarrow 1,\ \varrho \leftarrow 1$
$K \leftarrow K_e \parallel K_m$
return (K, σ, ϱ)

Algorithm $\mathcal{E}_K(m, \sigma)$

$\tau \leftarrow \mathcal{T}_{K_m}(\langle\sigma\rangle_\ell \parallel m)$
$c \leftarrow F_{K_e}(\langle\sigma\rangle_\ell) \oplus (m \parallel \tau)$
$\sigma \leftarrow \sigma + 1 \bmod 2^\ell$
return (c, σ)

Algorithm $\mathcal{D}_K(c, \varrho)$

if $|c| \neq n$ **then** $\varrho \leftarrow 0$
if $\varrho = 0$ **then**
  return $(\perp_0, \varrho)$
$w \leftarrow F_{K_e}(\langle\varrho\rangle_\ell) \oplus c$
parse w **as** $m \parallel \tau$
$v \leftarrow \mathcal{V}_{K_m}(\langle\varrho\rangle_\ell \parallel m, \tau)$
if $v = 1$
  then $\varrho \leftarrow \varrho + 1 \bmod 2^\ell$
else
  $\varrho \leftarrow 0$
  if $m[1] = 0$ **then** $m \leftarrow \perp_0$
  else $m \leftarrow \perp_1$
return (m, ϱ)

Fig. 7. The scheme $\mathcal{SE}_1$ of Theorem 10.

Note that in proving Theorem 10 we resorted to a stateful scheme. Only a stateful scheme can be INT-sfCTXT secure, and therefore the counterexample used to prove Corollary 12 needs to be stateful. The same cannot be said however about the separation in Corollary 11, and in fact it can be proven more generally using a stateless scheme, but we omit the details for the sake of brevity.

NEW RELATIONS. We now go on to investigate how chosen-ciphertext security can be obtained in the multiple-error setting. Given how useful the relations of [9] and [7] have turned out to be, it would make sense to attempt to derive analogous relations that hold more generally. The following theorem extends the relation of [9] to schemes with multiple errors. Its proof is in Appendix A.2.

Theorem 13 (IND-CVA $\wedge$ INT-CTXT $\longrightarrow$ IND-CCA)**.** *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. For any* IND-CCA *adversary $\mathcal{A}_{cca}$ there exist adversaries $\mathcal{A}_{cva}$ and $\mathcal{A}_{int}$ consuming similar resources to $\mathcal{A}_{cca}$ such that:*

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE}}(\mathcal{A}_{cca}) \leq \mathbf{Adv}^{\text{ind-cva}}_{\mathcal{SE}}(\mathcal{A}_{cva}) + 2 \cdot \mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE}}(\mathcal{A}_{int})\,. \tag{2}$$

A similar relation can be established for stateful chosen-ciphertext security, and each of these relations can be re-proven for security notions involving indistinguishability from random bits. We state these relations below.

Proposition 14.

$$\text{IND-CVA} \ \wedge \text{INT-sfCTXT} \longrightarrow \text{IND-sfCCA}$$

$$\text{IND\$-CVA} \ \wedge \ \text{INT-CTXT} \longrightarrow \text{IND\$-CCA}$$

$$\text{IND\$-sfCVA} \wedge \text{INT-sfCTXT} \longrightarrow \text{IND\$-sfCCA}$$

NECESSITY OF STRONG CIPHERTEXT INTEGRITY. The above relations can be seen as strengthened variants of the relations from [9] and [7], where we replaced CPA security with CVA security and adopted the stronger notions of ciphertext integrity. It is natural to ask whether the left-hand side of each relation can be somehow relaxed. We have seen in Corollaries 11 and 12 that reverting from CVA security to CPA security is not an option. However it is not evident whether it is necessary to require the stronger variants of ciphertext integrity. Theorem 15 answers this question by means of a separation, proving that strong ciphertext integrity is necessary for Theorem 13 to hold. Its proof is in Appendix A.3.

Theorem 15 (IND-CVA $\wedge$ INT-CTXT* $\not\longrightarrow$ IND-CCA)**.** *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme with a large message space $\mathcal{M}$ and an error space $\{\perp_0\}$, such that it is both* IND-CVA *secure and* INT-CTXT* *secure. Let the length of its ciphertexts be bounded above by 2^ℓ for some integer*

Algorithm $\overline{\mathcal{K}}$

$(K, \sigma, \varrho) \leftarrow \mathcal{K}$
$m^* \leftarrow\!\!\$\ \mathcal{M}$
$(c^*, \sigma) \leftarrow \mathcal{E}_K(m^*, \sigma)$
$(m, \varrho) \leftarrow \mathcal{D}_K(c^*, \varrho)$
$K_0 \leftarrow (K, m^*, c^*)$
return (K_0, σ, ϱ)

Algorithm $\overline{\mathcal{E}}_{K_0}(m, \sigma)$

if $(m = m^*)$ **then** $c \leftarrow c^*$
else $(c, \sigma) \leftarrow \mathcal{E}_K(m, \sigma)$
return $(0 \parallel c, \sigma)$

Algorithm $\overline{\mathcal{D}}_{K_0}(c, \varrho)$

parse c **as** $b \parallel c'$
if $(b = 0)$ **then**
  if $(c' = c^*)$ **then** $m \leftarrow m^*$
  else $(m, \varrho) \leftarrow \mathcal{D}_K(c', \varrho)$
else $\psi \leftarrow \langle |c^*| \rangle_\ell \parallel c^*$
  if $\langle c' \rangle^{-1} \leq |\psi|$ **then**
    $d \leftarrow \psi[\langle c' \rangle^{-1}], m \leftarrow \perp_d$
  else $m \leftarrow \perp_0$
return (m, ϱ)

Fig. 8. The scheme $\overline{\mathcal{SE}}$ of Theorem 15.

ℓ. Consider the scheme $\overline{\mathcal{SE}}$ having message space $\mathcal{M}$ and error space $\{\perp_0, \perp_1\}$ shown in Figure 8. For any IND-CVA *adversary $\mathcal{A}_{cva}$ making q_e left-or-right queries, and any* INT-CTXT* *adversary $\mathcal{A}_{int}$ making q_t verification queries, there exist adversaries $\mathcal{A}^1_{cva}$, $\mathcal{A}^2_{cva}$, and $\mathcal{A}^1_{int}$ (consuming similar resources to $\mathcal{A}_{cva}$ and $\mathcal{A}_{int}$) such that:*

$$\mathbf{Adv}^{\text{ind-cva}}_{\overline{\mathcal{SE}}}(\mathcal{A}_{cva}) \leq \mathbf{Adv}^{\text{ind-cva}}_{\mathcal{SE}}(\mathcal{A}^1_{cva}) + \frac{1}{2} \cdot \mathbf{Adv}^{\text{ind-cva}}_{\mathcal{SE}}(\mathcal{A}^2_{cva}) + \frac{q_e}{|\mathcal{M}|}\,, \tag{3a}$$

$$\mathbf{Adv}^{\text{int-ctxt*}}_{\overline{\mathcal{SE}}}(\mathcal{A}_{int}) \leq \mathbf{Adv}^{\text{int-ctxt*}}_{\mathcal{SE}}(\mathcal{A}^1_{int}) + \frac{q_t}{|\mathcal{M}|}\,. \tag{3b}$$

Moreover there exists an adversary $\mathcal{A}_{cca}$, making at most $(\ell + \max_{m \in \mathcal{M}}(|m|) + 1)$ decryption queries and one left-or-right query such that:

$$\mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{SE}}}(\mathcal{A}_{cca}) = 1\,. \tag{3c}$$

Theorem 15 also serves as a separation between INT-CTXT* and INT-CTXT, showing that the latter is strictly stronger. Separations similar to that of Theorem 15 corresponding to the relations of Proposition 14 can also be established.

Proposition 16.

$$\text{IND-CVA} \wedge \text{INT-sfCTXT}^* \not\rightarrow \text{IND-sfCCA}$$

$$\text{IND\$-CVA} \wedge \text{INT-CTXT}^* \not\rightarrow \text{IND\$-CCA}$$

$$\text{IND\$-sfCVA} \wedge \text{INT-sfCTXT}^* \not\rightarrow \text{IND\$-sfCCA}$$

4.1 More Separations

We now present a separation showing that IND-CVA is strictly stronger than IND-CPA. We actually show something slightly stronger, in that the separation also holds for schemes which are error invariant. This separation further serves to point out that, even for single-error schemes, Theorem 13 does not reduce to the relation of Bellare and Namprempre from [9]. The proof of Theorem 17 is in Appendix A.4.

Theorem 17 (IND-CPA$\wedge$INV-ERR$\not\rightarrow$ IND-CVA)**.** *Let $F : \mathcal{K}_e \times \{0,1\}^\ell \rightarrow \{0,1\}^n$ be a pseudorandom function, where ℓ is sufficiently large. Then the symmetric encryption scheme $\mathcal{SE}_2$ having message space $\cup_{k \geq 1}\{0,1\}^{nk}$ and error space $\{\perp\}$ shown in Figure 9 is such that, for any* IND-CPA *adversary $\mathcal{A}_{cpa}$ making q encryption queries totalling μ bits of plaintext, there exists a corresponding adversary $\mathcal{A}_{prf}$ (consuming similar resources to $\mathcal{A}_{cpa}$) with:*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}_2}(\mathcal{A}_{cpa}) \leq 2 \cdot \mathbf{Adv}^{\text{prf}}_F(\mathcal{A}_{prf}) + \left(\frac{\mu}{n} + q\right)\left(\frac{q-1}{2^\ell}\right)\,. \tag{4a}$$

Moreover there exists an efficient adversary $\mathcal{A}_{cva}$ such that:

$$\mathbf{Adv}^{\text{ind-cva}}_{\mathcal{SE}_2}(\mathcal{A}_{cva}) = 1\,. \tag{4b}$$

Algorithm $\overline{\mathcal{K}}$
$K \xleftarrow{\$} \mathcal{K}_e$
$\sigma \leftarrow \varepsilon$, $\varrho \leftarrow \varepsilon$
return (K, σ, ϱ)

Algorithm $\overline{\mathcal{E}}_K(m, \sigma)$
if $|m| \notin \{\alpha n : \alpha \geq 1\}$ **then**
  return $\perp$
$p \leftarrow |m|/n$
parse m **as** $m_1 \parallel \ldots \parallel m_p$
$m_{p+1} \leftarrow 0^n$, $c_0 \xleftarrow{\$} \{0,1\}^\ell$
for $i \leftarrow 1$ **to** $p+1$ **do**
  $c_i \leftarrow F_K(c_0 + i) \oplus m_i$
$c \leftarrow c_0 \parallel c_1 \parallel \ldots \parallel c_{p+1}$
return (c, σ)

Algorithm $\overline{\mathcal{D}}_K(c, \varrho)$
if $|c| \notin \{\ell + \alpha n : \alpha \geq 2\}$ **then**
  return $\perp$
$q \leftarrow (|c| - \ell)/n$
parse c **as** $c_0 \parallel \ldots \parallel c_q$
for $i \leftarrow 1$ **to** q **do**
  $m_i \leftarrow F_K(c_0 + i) \oplus c_i$
if $m_q \neq 0^n$ **then** $m \leftarrow \perp$
else $m \leftarrow m_1 \parallel \ldots \parallel m_{q-1}$
return (m, ϱ)

Fig. 9. The scheme $\mathcal{SE}_2$ of Theorem 17.

In Section 3 it was noted that if the IND-sfCVA experiment is defined in the obvious way, it would be syntactically equivalent to the IND-CVA experiment. In the case of indistinguishability from random bits, an analogous equivalence is not evident from the syntax. Theorem 18 settles this in the negative. Its proof is in Appendix A.5.

Theorem 18 (IND$-CVA $\wedge$ INV-ERR$\not\longrightarrow$ IND$-sfCVA)**.** *Let $F : \mathcal{K}_e \times \{0,1\}^\ell \to \{0,1\}^n$ be a pseudorandom function, where ℓ is sufficiently large. Let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a single-error MAC where $\mathcal{T} : \mathcal{K}_m \times \{0,1\}^* \to \{0,1\}^{\ell_{tag}}$ is pseudorandom. Consider the symmetric encryption scheme $\mathcal{SE}_3$ having message space $\cup_{k \geq 1}\{0,1\}^{nk}$ and error space $\{\perp\}$ shown in Figure 10. For any* IND$-CVA *adversary $\mathcal{A}_{cva}$ making q encryption queries totalling μ bits of plaintext, there exist three adversaries $\mathcal{A}^1_{prf}$, $\mathcal{A}^2_{prf}$, and $\mathcal{A}_{uf}$ with:*

$$\mathbf{Adv}^{\text{ind\$-cva}}_{\mathcal{SE}_3}(\mathcal{A}_{cva}) \leq \mathbf{Adv}^{\text{prf}}_{F}(\mathcal{A}^1_{prf}) + \mathbf{Adv}^{\text{prf}}_{\mathcal{T}}(\mathcal{A}^2_{prf}) + \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{MA}}(\mathcal{A}_{uf}) + \frac{\mu}{n} \cdot \left(\frac{q-1}{2^\ell}\right) + \frac{q(q-1)}{2^{\ell+n+1}}\,.$$

Moreover there exist efficient adversaries $\mathcal{A}_{sfcva}$ and $\mathcal{A}'_{uf}$ such that:

$$\mathbf{Adv}^{\text{ind\$-sfcva}}_{\mathcal{SE}_3}(\mathcal{A}_{sfcva}) = 1 - \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{MA}}(\mathcal{A}'_{uf})\,. \tag{5a}$$

Algorithm $\overline{\mathcal{K}}$
$K_e \xleftarrow{\$} \mathcal{K}_e$
$K_m \xleftarrow{\$} \mathcal{K}_m$
$K \leftarrow K_e \parallel K_m$
$\sigma \leftarrow \varepsilon$, $\varrho \leftarrow \varepsilon$
return (K, σ, ϱ)

Algorithm $\overline{\mathcal{E}}_K(m, \sigma)$
if $|m| \notin \{\alpha n : \alpha \geq 1\}$ **then**
  return $\perp$
$p \leftarrow |m|/n$
parse m **as** $m_1 \parallel \ldots \parallel m_p$
$c_0 \xleftarrow{\$} \{0,1\}^\ell$
for $i \leftarrow 1$ **to** p **do**
  $c_i \leftarrow F_K(c_0 + i) \oplus m_i$
$c \leftarrow c_0 \parallel c_1 \parallel \ldots \parallel c_p$
$\tau \leftarrow \mathcal{T}_{K_m}(c)$
return $(c \parallel \tau, \sigma)$

Algorithm $\overline{\mathcal{D}}_K(\psi, \varrho)$
if $|\psi| \notin \{\ell + \ell_{tag} + \alpha n : \alpha \geq 1\}$
  then return $(\perp, \varrho)$
parse ψ **as** $c \parallel \tau$
$v \leftarrow \mathcal{V}_{K_m}(c, \tau)$
if $(v \neq 1)$ **then**
  return $(\perp, \varrho)$
$q \leftarrow (|c| - \ell)/n$
parse c **as** $c_0 \parallel \ldots \parallel c_q$
for $i \leftarrow 1$ **to** q **do**
  $m_i \leftarrow F_K(c_0 + i) \oplus c_i$
$m \leftarrow m_1 \parallel \ldots \parallel m_q$
return (m, ϱ)

Fig. 10. The scheme $\mathcal{SE}_3$ of Theorem 18.

$\underline{\mathbf{Exp}^{\text{ind\$-cca3-b}}_{\mathcal{SE},\perp}(\mathcal{A})}$
$(K, \sigma, \varrho) \leftarrow \mathcal{K}$
$i \leftarrow 0, \mathtt{C} \leftarrow ()$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc\$}(\cdot),\mathsf{Dec}\varnothing(\cdot)}$
return (b')

$\underline{\mathsf{Enc\$}(m)}$
$(c, \sigma) \leftarrow \mathcal{E}_K(m, \sigma)$
if $b = 0$ **then** $c \leftarrow \{0,1\}^{|c|}$
$i \leftarrow i + 1, \mathtt{C}_i \leftarrow c$
return c

$\underline{\mathsf{Dec}\varnothing(c)}$
$(m, \varrho) \leftarrow \mathcal{D}_K(c, \varrho)$
if $b = 0$ **then** $m \leftarrow \perp$
if $c \in \mathtt{C}$ **then** $m \leftarrow \lightning$
return m

Fig. 11. IND$-CCA3 experiment for multiple-error symmetric encryption schemes.

5 Further Relations and the IND$-CCA3 Notion

AUTHENTICATED-ENCRYPTION SECURITY. Following the work of Bellare and Namprempre [9], chosen-plaintext security and ciphertext integrity were identified as the two security goals for symmetric encryption. Rogaway and Shrimpton [25] presented a *single* security notion, sometimes referred to as IND$-CCA3 and more commonly called authenticated-encryption security, that is equivalent to the combination of chosen plaintext security and ciphertext integrity. We now present a natural extension of this notion to the multiple error setting. Then in Theorem 20 we show that this characterisation is equivalent to the combination of chosen-plaintext security, weak chosen ciphertext integrity, and error invariance.

Definition 19 (IND$-CCA3 **notion for multiple-error symmetric encryption).** *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a multiple-error symmetric encryption scheme with error space $\mathcal{S}_\perp$. For an adversary $\mathcal{A}$, an error $\perp \in \mathcal{S}_\perp$ and a bit b, define experiment $\mathbf{Exp}^{\text{ind\$-cca3-b}}_{\mathcal{SE},\perp}(\mathcal{A})$ as shown in Figure 11. First $\mathcal{K}$ is called to generate a key K, an initial encryption state σ, and an initial decryption state ϱ. The adversary $\mathcal{A}$ is then given access to a special encryption oracle $\mathsf{Enc\$}(\cdot)$ and a special decryption oracle $\mathsf{Dec}\varnothing(\cdot)$. When $b = 1$ both oracles behave as normal encryption and decryption oracles. When $b = 0$ then $\mathsf{Enc\$}(\cdot)$ will return a random bit string (of the same length as an actual ciphertext would have been), and $\mathsf{Dec}\varnothing(\cdot)$ will always return $\perp$ (unless the queried ciphertext was output by $\mathsf{Enc\$}(\cdot)$, in which case it will return $\lightning$).*

The adversary's goal is to output a bit b', as its guess of the challenge bit b. The experiment returns b' as well and, for $\perp \in \mathcal{S}_\perp$ and an adversary $\mathcal{A}$, the advantage is defined as:

$$\mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{ind\$-cca3-1}}_{\mathcal{SE},\perp}(\mathcal{A}) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind\$-cca3-0}}_{\mathcal{SE},\perp}(\mathcal{A}) = 1\right].$$

The scheme $\mathcal{SE}$ is said to be IND$-CCA3 *secure if there exists $\perp \in \mathcal{S}_\perp$ such that for every adversary $\mathcal{A}$ with reasonable resources its advantage $\mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A})$ is small.*

Note: An IND-CCA3 notion can be defined by replacing the $\mathsf{Enc\$}(\cdot)$ oracle with a *real-or-random* encryption oracle (cf. [5]). Such an oracle returns either an encryption of the queried message or an encryption of a random message of the same length.

The proof of the following theorem can be found in Appendix A.6.

Theorem 20 (IND$-CPA $\wedge$ INT-CTXT* $\wedge$ INV-ERR $\rightleftarrows$ IND$-CCA3)**.** *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme with error space $\mathcal{S}_\perp$.*

- *For any $\perp \in \mathcal{S}_\perp$ and any adversary $\mathcal{A}_{cca3}$ there exist adversaries $\mathcal{A}_{cpa}$, $\mathcal{A}_{int}$ and $\mathcal{A}_{err}$ (consuming similar resources to $\mathcal{A}_{cca3}$) such that:*

$$\mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}_{cca3}) \leq \mathbf{Adv}^{\text{ind\$-cpa}}_{\mathcal{SE}}(\mathcal{A}_{cpa}) + \mathbf{Adv}^{\text{int-ctxt}*}_{\mathcal{SE}}(\mathcal{A}_{int}) + \mathbf{Adv}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A}_{err}). \quad (6)$$

- *For any $\perp \in \mathcal{S}_\perp$ and any three adversaries $\mathcal{A}'_{cpa}$, $\mathcal{A}'_{int}$ and $\mathcal{A}'_{err}$ there exist three corresponding adversaries $\mathcal{A}^1_{cca3}$, $\mathcal{A}^2_{cca3}$ and $\mathcal{A}^3_{cca3}$ (consuming similar resources to $\mathcal{A}'_{cpa}$, $\mathcal{A}'_{int}$ and $\mathcal{A}'_{err}$, respectively) such that:*

$$\mathbf{Adv}^{\text{ind\$-cpa}}_{\mathcal{SE}}(\mathcal{A}'_{cpa}) \leq \mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}^1_{cca3}), \quad (7a)$$

$$\mathbf{Adv}^{\text{int-ctxt}*}_{\mathcal{SE}}(\mathcal{A}'_{int}) \leq 2 \cdot \mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}^2_{cca3}), \quad (7b)$$

$$\mathbf{Adv}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A}'_{err}) \leq 2 \cdot \mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}^3_{cca3}). \quad (7c)$$

It can be similarly shown that:

Proposition 21. $\mathsf{IND\text{-}CPA} \wedge \mathsf{INT\text{-}CTXT}^* \wedge \mathsf{INV\text{-}ERR} \rightleftarrows \mathsf{IND\text{-}CCA3}$.

It is easy to see that IND$-CCA3 security guarantees IND$-CCA security in the multiple error setting, which is what we are ultimatly after. In fact we can say something slightly stronger, as indicated in Proposition 22. The proof is straightforward and we omit it.

Proposition 22.

$$\mathsf{IND\$\text{-}CCA3} \longrightarrow \mathsf{IND\$\text{-}CVA} \wedge \mathsf{INT\text{-}CTXT} \longrightarrow \mathsf{IND\$\text{-}CCA}$$

$$\mathsf{IND\text{-}CCA3} \longrightarrow \mathsf{IND\text{-}CVA} \wedge \mathsf{INT\text{-}CTXT} \longrightarrow \mathsf{IND\text{-}CCA}$$

6 The Security of Encode-then-Encrypt-then-MAC

Results of Bellare and Namprempre [9] and Krawczyk [19] provide formal evidence for preferring Encrypt-then-MAC (EtM) over other generic compositions like MAC-then-encrypt (MtE). On the other hand, by combining results from [19] and [6], it can be shown that MtE is actually IND-CCA secure when instantiated with CBC-mode encryption or a secure stream cipher (instantiated using counter-mode encryption, for example). Thus the analysis of [9, 19] does not help to separate EtM and MtE when both are suitably instantiated.

Nonetheless practical secure communications systems (employing CBC and counter-mode encryption) based on EtM have so far proved themselves less vulnerable to attack than ones based on MtE. For example, attacks on TLS in [10, 2, 3] and IPsec in [12] exploit weaknesses in specific MtE constructions, while attacks against deployed EtM constructions seem rarer.

Reconsidering the EtM and MtE compositions in the multiple-error setting provides new formal grounds for preferring the EtM composition. In what follows, we show that the EtM composition enjoys a robust form of security (in a sense to be made precise). We then go on to show how the above-mentioned attacks on specific MtE constructions can be captured in our multiple-error setting.

To make our considerations more realistic, in place of EtM, we actually consider an encode-then-encrypt-then-MAC (EEM) composition, where the encoding step accounts for the pre-processing (such as padding) that is common in practical schemes. Similarly, we will consider the MAC-then-Encode-then-Ecrypt (MEE) composition in place of MtE when discussing attacks.

Our EEM composition is specified in Figure 12. Theorem 23 shows that the EEM composition is robust, in the sense that it provides IND-CVA and INT-CTXT security, and therefore IND-CCA security, in the multiple-error setting. The result holds irrespective of the encoding scheme used (and any error messages it returns) and independent of whatever error messages the encryption component returns, so long as the encryption component is IND-CPA and the MAC is SUF-CMA. The proof of Theorem 23 is in Appendix A.7.

Theorem 23 (EEM provides IND-CVA + INT-CTXT). *Suppose $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ is a symmetric encryption scheme with message space $\mathcal{M}$ and error space $\mathcal{S}_\perp$. Let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a MAC with error space $\mathcal{Q}_\perp$ producing tags of length ℓ_{tag}. Let $\mathcal{ES} = (\mathcal{EC}, \mathcal{DC})$ be a length-regular encoding scheme with domain $\overline{\mathcal{M}}$, codomain $\mathcal{M}$, and error space $\mathcal{U}_\perp$. Figure 12 then defines a symmetric encryption scheme $\mathcal{EEM}$ with message space $\overline{\mathcal{M}}$ and error space $\overline{\mathcal{S}_\perp} = \mathcal{S}_\perp \cup \mathcal{Q}_\perp \cup \mathcal{U}_\perp \cup \{\perp_0\}$, for some $\perp_0 \notin \mathcal{S}_\perp \cup \mathcal{Q}_\perp \cup \mathcal{U}_\perp$. For any* IND-CVA *adversary $\mathcal{A}_{cva}$ and any* INT-CTXT *adversary $\mathcal{A}_{int}$ against $\mathcal{EEM}$, there exist adversaries $\mathcal{A}_{cpa}$, $\mathcal{A}^1_{suf}$, and $\mathcal{A}^2_{suf}$ such that:*

$$\mathbf{Adv}^{\text{ind-cva}}_{\mathcal{EEM}}(\mathcal{A}_{cva}) \leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(\mathcal{A}_{cpa}) + \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(\mathcal{A}^1_{suf}), \tag{8}$$

$$\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{EEM}}(\mathcal{A}_{int}) \leq \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(\mathcal{A}^2_{suf}). \tag{9}$$

Moreover, these adversaries consume similar resources to $\mathcal{A}_{cva}$ and $\mathcal{A}_{int}$.

Algorithm $\overline{\mathcal{K}}$

$(K_e, \sigma, \varrho) \leftarrow \mathcal{K}_e$
$K_m \leftarrow \mathcal{K}_m$
$K \leftarrow K_e \parallel K_m$
return (K, σ, ϱ)

Algorithm $\overline{\mathcal{E}}_K(m, \sigma)$

$w \leftarrow \mathcal{EC}(m)$
$(c, \sigma) \leftarrow \mathcal{E}_{K_e}(w, \sigma)$
$\tau \leftarrow \mathcal{T}_{K_m}(c)$
return $(c \parallel \tau, \sigma)$

Algorithm $\overline{\mathcal{D}}_K(\psi, \varrho)$

if $|\psi| < \ell_{tag} + 1$ **then**
  return $(\perp_0, \varrho)$
parse ψ **as** $c \parallel \tau$
$v \leftarrow \mathcal{V}_{K_m}(c, \tau)$
if $v \in \mathcal{Q}_\perp$ **then**
  return (v, ϱ)
$(w, \varrho) \leftarrow \mathcal{D}_{K_e}(c, \varrho)$
if $w \in \mathcal{S}_\perp$ **then**
  return (w, ϱ)
$m \leftarrow \mathcal{DC}(w)$
return (m, ϱ)

Fig. 12. The generic Encode-then-Encypt-then-MAC composition $\mathcal{EEM}$ with distinguishable decryption failures.

In fact, we can prove that EEM also provides IND-CCA3 security if its MAC component only has a single error message. We omit the details.

As a complement to the above result, it is instructive to model attacks on instantiations of the MAC-then-Encode-then-Ecrypt (MEE) composition in our multiple-error setting.

- TLS uses a MEE composition in which the encoding step involves the addition of padding having a specific format. This format should be checked for upon decryption, with a failure resulting in an error message. Likewise, the MAC verification may fail, resulting in an error message. Error messages in TLS are encrypted in general, and MAC failures and padding failures are indicated by the same error message. The attacks on TLS [10] and on DTLS [2] use timing differences to distinguish MAC failures from padding failures. These differences can be modelled by introducing distinct error messages for the two failure events (even if at the byte level, the messages are indistinguishable).
- Certain configurations of IPsec use a MEE composition to cryptographically protect IP packets. The security of these configurations were studied in detail in [12]. Here, the encoding step includes a padding portion as well as a header portion, and it is the ability to discern between malformed padding and a malformed header that gives rise to the attacks in [12]. In fact, malformed padding leads to packets being silently dropped, while malformed headers lead to encrypted error messages being sent on the network. Again, the attacks can be modelled by introducing distinct error messages for the different events, even though one of the events does not result in an actual error message being sent (since the absence of a message also leaks information to the adversary).
- The recent Lucky 13 attack on TLS [3] exploits timing differences arising in HMAC's verification algorithm. More specifically each compression function evaluation in HMAC results in additional processing time during decryption that can be detected by the adversary from the time delay in returning TLS's MAC failure message; the size of the delay relates to the amount of TLS padding previously removed and can be used to infer plaintext in an extension of Vaudenay's padding oracle attack [26]. This timing channel can be modelled in our framework by transforming HMAC into a multiple-error MAC. Then the error messages that this version of HMAC returns can be easily predicted from the length of the string on which the tag is to be verified. It follows from this observation that any proof of SUF-CMA security for the usual single-error HMAC can be extended to this multiple-error version of HMAC. So, while this multiple-error HMAC is still SUF-CMA secure, its interaction with the TLS padding renders the MEE composition used in TLS insecure. By contrast, as established in Theorem 23, an EEM composition would not be compromised by such an implementation flaw.

Acknowledgements

This work has been supported in part by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. Alexandra Boldyreva is supported by NSF: CNS-0831184. Jean Paul Degabriele is supported by Vodafone Group Services Limited, a Thomas Holloway Research

Studentship, and the Strategic Educational Pathways Scholarship Scheme (Malta), part-financed by the European Union European Social Fund. Kenneth Paterson is supported by EPSRC Leadership Fellowship EP/H005455/1.

References

[1] M.R. Albrecht, K.G. Paterson, and G.J. Watson. Plaintext recovery attacks against SSH. In *IEEE Symposium on Security and Privacy*, pages 16–26. IEEE Computer Society, 2009.

[2] N.J. AlFardan and K.G. Paterson. Plaintext-recovery attacks against Datagram TLS. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS 2012)*.

[3] N.J. AlFardan and K.G. Paterson. Lucky thirteen: Breaking the TLS and DTLS record protocols. To appear in *IEEE Symposium on Security and Privacy 2013*, available at `http://www.isg.rhul.ac.uk/tls/TLStiming.pdf`.

[4] A. Bauer, J.S. Coron, D. Naccache, M. Tibouchi, and D. Vergnaud. On the broadcast and validity-checking security of PKCS#1 v1.5 encryption. In J. Zhou and M. Yung (eds.), *ACNS 2010*, volume 6123 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2010.

[5] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of 38th Annual Symposium on Foundations of Computer Science (FOCS 1997)*, pages 394–403. IEEE, 1997.

[6] M. Bellare, O. Goldreich, A. Mityagin. The power of verification queries in message authentication and authenticated encryption. *IACR Cryptology ePrint Archive*, `http://eprint.iacr.org/2004/309`.

[7] M. Bellare, T. Kohno, and C. Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the encode-then-encrypt-and-MAC paradigm. *ACM Transactions on Information and Systems Security*, 7(2):206–241, 2004.

[8] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In H. Krawczyk (ed.), *CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 1998.

[9] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto (ed.), *ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.

[10] B. Canvel, A.P. Hiltgen, S. Vaudenay, and M. Vuagnoux. Password interception in a SSL/TLS channel. In D. Boneh (ed.), *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 583–599. Springer, 2003.

[11] J.P. Degabriele and K.G. Paterson. Attacking the IPsec standards in encryption-only configurations. In *IEEE Symposium on Security and Privacy*, pages 335–349. IEEE Computer Society, 2007.

[12] J.P. Degabriele and K.G. Paterson. On the (in)security of IPsec in MAC-then-encrypt configurations. In E. Al-Shaer, A.D. Keromytis and V. Shmatikov (eds.), *ACM Conference on Computer and Communications Security*, pages 493–504. ACM, 2010.

[13] T. Duong and J. Rizzo. Cryptography in the web: The case of cryptographic design flaws in ASP.NET. In *IEEE Symposium on Security and Privacy*, pages 481–489. IEEE Computer Society, 2011

[14] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *Journal of ACM*, volume 33 no. 4, pages 792–807. ACM, 1986.

[15] C. Hall, I. Goldberg and B. Schneier. Reaction attacks against several public-key cryptosystems. In V. Varadharajan and Y. Mu (eds.), *ICICS 99, Sydney*, volume 1726 of *Lecture Notes in Computer Science*, pages 2–12. Springer, 1999.

[16] J. Håstad, R. Impagliazzo, L.A. Levin, and M. Luby. A pseudorandom generator from any one-way function. In *SIAM Journal on Computing*, volume 28 no. 4, pages 1364–1396. SIAM, 1999.

[17] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *Proceedings of 30th Annual Symposium on Foundations of Computer Science (FOCS 1989)*, pages 230–235. IEEE, 1989.

[18] T. Jager and J. Somorovsky. How to break XML encryption. In Y. Chen, G. Danezis and V. Shmatikov (eds.) *ACM Conference on Computer and Communications Security*, pages 413–422. ACM, 2011.

[19] H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In J. Kilian (ed.), *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 310–331. Springer, 2001.

[20] J. Manger. A chosen ciphertext attack on RSA optimal asymmetric encryption padding (OAEP) as Standardized in PKCS #1 v2.0. In J. Kilian (ed.), *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 230–238. Springer, 2001.

[21] K.G. Paterson and G.J. Watson. Plaintext-dependent decryption: A formal security treatment of SSH-CTR. In H. Gilbert (ed.), *EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 345–361. Springer, 2010.

[22] K.G. Paterson, T.E. Shrimpton and T. Ristenpart. Tag size does matter: Attacks and proofs for the TLS record protocol. In D.H. Lee and X. Wang (eds.), *ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 372–389, Springer, 2011.

[23] K.G. Paterson and G.J. Watson. Authenticated-encryption with padding: A formal security treatment. In D. Naccache (ed.), *Cryptography and Security 2012*, volume 6805 of *Lecture Notes in Computer Science*, pages 83–107. Springer, 2012.

[24] P. Rogaway. Nonce-based symmetric encryption. In B. Roy and W. Meier (eds.), *FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 348–359. Springer 2004.

[25] P. Rogaway, and T. Shrimpton. A provable-security treatment of the key-wrap problem. In S. Vaudenay (ed.), *EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.
[26] S. Vaudenay. Security flaws Induced by CBC padding - Applications to SSL, IPSEC, WTLS. In L.R. Knudsen (ed.), *EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 534–546. Springer 2002.

A Proofs

A.1 Proof of Theorem 10

The correctness of the constructed scheme is easy to verify and we therefore proceed to prove the first part of the theorem. For any adversry $\mathcal{A}_{cpa}$, making at most $2^\ell - 1$ encryption queries, we construct $\mathcal{A}_{prf}$ as follows. Adversary $\mathcal{A}_{prf}$ runs $\mathcal{K}_m$ to get a key for the MAC, it then runs $\mathcal{A}_{cpa}$ and provides it with a simulation of its left-or-right encryption oracle. Essentially $\mathcal{A}_{prf}$ selects a uniformly random bit d and uses its own oracle together with its MAC key to encrypt m_d according to the construction in Figure 7, where the pseudorandom function is replaced by its own oracle. Finally, if $\mathcal{A}_{cpa}$'s output is equal to d, then $\mathcal{A}_{prf}$ outputs 1 otherwise it outputs 0. Now when $\mathcal{A}_{prf}$'s oracle is instantiated with F it provides $\mathcal{A}_{prf}$ with a perfect simulation of the IND-CPA experiment. On the other hand when $\mathcal{A}_{prf}$'s oracle is a random function, the ciphertexts returned to $\mathcal{A}_{cpa}$ provide no information about d, i.e. d is information-theoretically hidden. Therefore we have that:

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{prf}}_{F}(\mathcal{A}_{prf}) &= \mathbf{Pr}\left[K_e \leftarrow\!\!\$\ \mathcal{K}_e : \mathcal{A}_{prf}^{F_{K_e}(\cdot)} = 1\right] - \mathbf{Pr}\left[f \leftarrow\!\!\$\ \mathrm{Func}(\ell, n) : \mathcal{A}_{prf}^{f(\cdot)} = 1\right] \\
&= \mathbf{Pr}\left[d \leftarrow \{0,1\} : \mathbf{Exp}^{\text{ind-cpa-d}}_{\mathcal{SE}_1}(\mathcal{A}_{cpa}) = d\right] - \frac{1}{2} \\
&= \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}_1}(\mathcal{A}_{cpa}) - \frac{1}{2} = \frac{1}{2} \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}_1}(\mathcal{A}_{cpa})\,.
\end{aligned}$$

Inequality (1a) thus follows, and we now prove the second inequality.

For any adversary $\mathcal{A}_{int}$ making at most $2^\ell - 1$ encryption queries, adversary $\mathcal{A}_{uf}$ proceeds as follows. It samples a key K_e for the pseudorandom function F and then runs adversary $\mathcal{A}_{int}$. It simulates the encryption oracle by using its own tagging oracle and the pseudorandom function under the sampled key. In addition it also maintains an ordered list of the messages $\mathcal{A}_{int}$ queries to the encryption oracle together with their corresponding ciphertexts. It then simulates the try oracle as follows. As long as $\mathcal{A}_{int}$'s queries are in sync, i.e. they match the ciphertexts in $\mathcal{A}_{uf}$'s list in the exact same order, it returns $\lightning$. Alternatively consider $\mathcal{A}_{int}$'s first out-of-sync query c_i, let this be its i^{th} try query. In this case $\mathcal{A}_{uf}$ first checks that $|c_i| = n$ and if not it halts, otherwise it computes the XOR of c_i and $F_{K_e}(\langle i \rangle_\ell)$. It then parses the result into a message and a tag, prepends the message with the string $\langle i \rangle_\ell$, submits it together with the tag to its verification oracle and halts.

Note that due to the scheme's construction $\mathcal{A}_{int}$ can only win within its first $2^\ell - 1$ try queries. Since we are interested in bounding its advantage we only need to consider the case where $i \leq 2^\ell - 1$. Now $\mathcal{A}_{uf}$ provides $\mathcal{A}_{int}$ with a perfect simultation of the INT-sfCTXT experiment until $\mathcal{A}_{int}$ makes its first out-of-sync try query, at which point $\mathcal{A}_{int}$ will either win or lose the experiment (again due to the scheme's construction). Moreover because $\mathcal{A}_{uf}$'s only verification query corresponds to an out-of-sync query and $\mathcal{A}_{int}$ can only make at most $2^\ell - 1$ encryption queries, it follows that the message prepended with $\langle i \rangle_\ell$ could not have been previously queried by $\mathcal{A}_{uf}$ to its tagging oracle. Thus whenever $\mathcal{A}_{int}$ wins $\mathcal{A}_{uf}$ also wins, and inequality (1a) follows.

We conclude the proof by describing adversary $\mathcal{A}_{cca}$ which breaks the IND-CCA security of $\mathcal{SE}_1$. The adversary submits $(0\|0^{n-\ell_{tag}-1}, 1\|0^{n-\ell_{tag}-1})$ to the left-or-right oracle and gets in return a ciphertext c^*. It then submits $c^* \oplus (0^{n-\ell_{tag}-1}\|1)$ to the decryption oracle. If the decryption oracle returns $\perp_0$ then $\mathcal{A}_{cca}$ outputs 0, otherwise it outputs 1. Due to the scheme's construction, this adversary will always win except for the case where the decryption oracle returns $m \notin \{\perp_0, \perp_1\}$. However this would imply a MAC forgery. Adversary $\mathcal{A}_{cca}$ can then be easily transformed into a UF-CMA adversary $\mathcal{A}'_{uf}$ against $\mathcal{MA}$ such that equation (1c) holds.

□

A.2 Proof of Theorem 13

To any IND-CCA adversary $\mathcal{A}_{cca}$ we can associate an IND-CVA adversary $\mathcal{A}_{cva}$ and an INT-CTXT adversary $\mathcal{A}_{int}$. Both $\mathcal{A}_{cva}$ and $\mathcal{A}_{int}$ operate by running $\mathcal{A}_{cca}$, and then attempt to simulate its enviroment as follows. Adversary $\mathcal{A}_{cva}$ forwards $\mathcal{A}_{cca}$'s left-or-right queries to its own left-or-right oracle, and forwards decryption queries to its validation oracle. If the ciphertext turns out to be invalid it returns the error message to $\mathcal{A}_{cca}$, otherwise it aborts. It then outputs whatever $\mathcal{A}_{cca}$ outputs. Adversary $\mathcal{A}_{int}$ picks a bit uniformly at random, and uses this together with its encryption oracle to simulate $\mathcal{A}_{cca}$'s left-or-right oracle. It forwards decryption queries to its verification oracle and returns any error messages back to $\mathcal{A}_{cca}$.

Let W represent the event $\mathbf{Exp}^{\text{ind-cca-b}}_{\mathcal{SE}}(\mathcal{A}_{cca}) = b$ where b is picked uniformly at random. Let E represent the event that $\mathcal{A}_{cca}$ makes a valid decryption query. We then have that:

$$\begin{aligned}\mathbf{Pr}\left[W\right] &= \mathbf{Pr}\left[W \wedge \overline{E}\right] + \mathbf{Pr}\left[W \wedge E\right] \\ &\leq \mathbf{Pr}\left[W \wedge \overline{E}\right] + \mathbf{Pr}\left[E\right].\end{aligned}$$

We now bound each of the terms on the right-hand side of the last inequality. Note that $\mathcal{A}_{int}$ simulates $\mathcal{A}_{cca}$'s environment perfectly until the point where $\mathcal{A}_{cca}$ makes a valid decryption query. Thus it follows that whenever E occurs, $\mathcal{A}_{int}$ wins the INT-CTXT experiment. On the other hand if E does not occur, then $\mathcal{A}_{cva}$'s simulation of $\mathcal{A}_{cca}$'s environment is perfect. Consequently whenever event $W \wedge \overline{E}$ occurs, $\mathcal{A}_{cva}$ wins the IND-CVA experiment. Equation (2) follows by combining the above and noting that:

$$\mathbf{Adv}^{\text{ind-atk}}_{\mathcal{SE}}(\mathcal{A}) = 2 \cdot \mathbf{Pr}\left[b \leftarrow \{0,1\} : \mathbf{Exp}^{\text{ind-atk-b}}_{\mathcal{SE}}(\mathcal{A}) = b\right] - 1\,. \tag{10}$$

□

A.3 Proof of Theorem 15

Correctness of the constructed scheme follows easily from the correctness of the original scheme, and we thus proceed to prove equation (3a). Adversary $\mathcal{A}^1_{cva}$ starts by picking a message m^* uniformly at random from the message space and computes ψ by querying (m^*, m^*) to its left-or-right oracle. $\mathcal{A}^1_{cva}$ also submits c^* to its ciphertext-validity oracle to maintain the states of its oracles synchronised and thereby correctly simulate $\overline{\mathcal{SE}}$. It then runs $\mathcal{A}_{cva}$ and simulates its oracles according to the construction in Figure 8 using its own oracles. Note that $\mathcal{A}^1_{cva}$ provides a perfect simulation to $\mathcal{A}_{cva}$, unless the latter queries m^*. In that case $\mathcal{A}^1_{cva}$ aborts and outputs a bit chosen uniformly at random.

Let W and W^1 represent respectively the events $\mathbf{Exp}^{\text{ind-cva-b}}_{\overline{\mathcal{SE}}}(\mathcal{A}_{cva}) = b$ and $\mathbf{Exp}^{\text{ind-cva-d}}_{\mathcal{SE}}(\mathcal{A}^1_{cva}) = d$ where b and d are picked uniformly at random. Furthermore let E denote the event that $\mathcal{A}_{cva}$ makes an encryption query which includes m^*. We then have that:

$$\mathbf{Pr}\left[W^1\right] = \mathbf{Pr}\left[W \wedge \overline{E}\right] + \frac{1}{2} \cdot \mathbf{Pr}\left[E\right]$$

$$\mathbf{Pr}\left[W^1\right] - \frac{1}{2} \cdot \mathbf{Pr}\left[E\right] + \mathbf{Pr}\left[W \wedge E\right] = \mathbf{Pr}\left[W \wedge \overline{E}\right] + \mathbf{Pr}\left[W \wedge E\right]$$

$$\mathbf{Pr}\left[W^1\right] + \frac{1}{2} \cdot \mathbf{Pr}\left[E\right] \geq \mathbf{Pr}\left[W\right]$$

$$\mathbf{Adv}^{\text{ind-cva}}_{\mathcal{SE}}(\mathcal{A}^1_{cva}) + \frac{1}{2} \cdot \mathbf{Pr}\left[E\right] \geq \mathbf{Adv}^{\text{ind-cva}}_{\overline{\mathcal{SE}}}(\mathcal{A}_{cva})\,. \tag{11}$$

It now remains to bound $\mathbf{Pr}\left[E\right]$. Note that (due to the details of $\overline{\mathcal{SE}}$'s construction) $\mathcal{A}_{cva}$ can recover the encryption of m^* from its ciphertext-validity oracle, and consequently we cannot bound $\mathbf{Pr}\left[E\right]$ using an information-theoretic argument. Instead we construct adversary $\mathcal{A}^2_{cva}$ such that if $\mathcal{A}_{cva}$ can do significantly better than what is information-theoretically possible, then $\mathcal{A}^2_{cva}$ breaks the IND-CVA security of

$\mathcal{SE}$. Adversary $\mathcal{A}^2_{cva}$ proceeds exactly as $\mathcal{A}^1_{cva}$, except that it computes c^* by querying (m^+, m^*) to its left-or-right oracle for some message m^+ chosen uniformly at random. Then if at any point during its runtime $\mathcal{A}_{cva}$ queries m^*, $\mathcal{A}^2_{cva}$ outputs 1 else it outputs 0. It then follows that:

$$\begin{aligned}\mathbf{Adv}^{\text{ind-cva}}_{\mathcal{SE}}(\mathcal{A}^2_{cva}) &= \Pr\left[\,\mathbf{Exp}^{\text{ind-cva-1}}_{\mathcal{SE}}(\mathcal{A}^2_{cva}) = 1\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{ind-cva-0}}_{\mathcal{SE}}(\mathcal{A}^2_{cva}) = 1\,\right] \\ &\geq \mathbf{Pr}\left[E\right] - \frac{2q_e}{|\mathcal{M}|}\,. \end{aligned} \tag{12}$$

The second line follows from the fact that by definition $\mathcal{A}^2_{cva}$ ouptuts 1 exactly when E occurs; whereas in the second experiment $\mathcal{A}^2_{cva}$ has no information about m^* and hence an information theoretic argument can be applied. Combining equations (11) and (12) yields equation (3a).

Adversary $\mathcal{A}^1_{int}$ picks a message m^* uniformly at random, computes c^* using its encryption oracle, and then queries c^* to its try oracle to maintain the states synchronised. It then runs $\mathcal{A}_{int}$ and simulates its environment using its own oracles. Specifically it forwards encryption queries to its own encryption oracle and prepends the resulting ciphertexts with a 0 bit. If $\mathcal{A}_{int}$ queries m^* it returns $0\|c^*$. As regards verification queries, it returns $\perp$ for ciphertexts starting with a 1 bit, and for all other queries it chops off the first bit and forwards the remaining ciphertext to its own verification oracle. This provides $\mathcal{A}_{int}$ with a perfect simulation of its environment. Let Z and Z^1 represent respectively the events that $\mathcal{A}_{int}$ and $\mathcal{A}^1_{int}$ win the INT-CTXT* experiment, and let F represent the event that $\mathcal{A}_{int}$ queries $0\|c^*$ to its verification oracle without querying m^* to its encryption oracle. We then have that:

$$\begin{aligned}\mathbf{Pr}\left[Z\right] &= \mathbf{Pr}\left[Z \wedge \overline{F}\right] + \mathbf{Pr}\left[Z \wedge F\right] \\ &\leq \mathbf{Pr}\left[Z^1\right] + \mathbf{Pr}\left[F\right] \\ \mathbf{Adv}^{\text{int-ctxt}*}_{\overline{\mathcal{SE}}}(\mathcal{A}_{int}) &\leq \mathbf{Adv}^{\text{int-ctxt}*}_{\mathcal{SE}}(\mathcal{A}^1_{int}) + \frac{q_t}{|\mathcal{C}|}\,. \end{aligned} \tag{13}$$

The bound on $\mathbf{Pr}\left[F\right]$ follows from the fact that unless $\mathcal{A}_{int}$ queries m^* to its encryption oracle, it has no partial information about c^*. Thus equation (3b) follows from equation (13) by noting that $|\mathcal{C}|$ is at least as large as $|\mathcal{M}|$ (from the correctness of $\mathcal{SE}$).

We now conclude the proof by describing adversary $\mathcal{A}_{cca}$. Note (from the construction of $\overline{\mathcal{SE}}$) that the decryption of $1\|\langle i\rangle_\ell$ leaks the i^{th} bit of the string $\psi = \langle|c^*|\rangle_\ell\|c^*$ through the returned error message. Thus $\mathcal{A}_{cca}$ starts by making a series of ℓ decryption queries, $1\|\langle 0\rangle_\ell$, $1\|\langle 1\rangle_\ell$, $1\|\langle 2\rangle_\ell$, …, $1\|\langle \ell-1\rangle_\ell$, to recover the value $|c^*|$. It then makes a second series of decryption queries, $1\|\langle \ell\rangle_\ell$, $1\|\langle \ell+1\rangle_\ell$, …, $1\|\langle \ell-1+|c^*|\rangle_\ell$, to recover c^*. It can now recover the message m^* by querying the ciphertext $0\|c^*$ to its decryption oracle. Having recovered m^*, it submits the pair (m^*, m°) to its left-or-right oracle, where $m^* \neq m^\circ$. If the returned ciphertext is equal to $0\,\|\,c^*$ the adversary outputs 0, otherwise it outputs 1. This adversary is always successful, and hence equation (3c) follows.

□

A.4 Proof of Theorem 17

It is easy to verify that the constructed scheme is correct, and since its error space contains only a single element it is trivially INV-ERR. We therefore proceed to prove that it is IND-CPA secure. For any adversary $\mathcal{A}_{cpa}$ we construct adversary $\mathcal{A}_{prf}$ as follows. Adversary $\mathcal{A}_{prf}$ selects a uniformly random bit d, runs $\mathcal{A}_{cpa}$ and simulates its left-or-right encryption oracle. It does so by using its own oracle to encrypt m_d according to the construction in Figure 9, where the pseudorandom function is replaced by $\mathcal{A}_{prf}$'s oracle. Then if $\mathcal{A}_{cpa}$'s output is equal to d, $\mathcal{A}_{prf}$ outputs 1 otherwise it outputs 0. Note that when $\mathcal{A}_{prf}$'s oracle is instantiated with F it provides $\mathcal{A}_{prf}$ with a perfect simulation of the IND-CPA experiment. On the other hand when $\mathcal{A}_{prf}$'s oracle is a random function, the ciphertexts returned to $\mathcal{A}_{cpa}$ provide no information about d, unless $\mathcal{A}_{prf}$'s oracle is queried on the same input more than once. Let E denote

the event that $\mathcal{A}_{prf}$ queries its oracle on the same input more than once when simulating the left-or-right oracle. We then have that:

$$\mathbf{Adv}^{\mathrm{prf}}_{F}(\mathcal{A}_{prf}) = \mathbf{Pr}\left[K \leftarrow\!\!\$\, \mathcal{K}_e : \mathcal{A}_{prf}^{F_K(\cdot)} = 1\right] - \mathbf{Pr}\left[f \leftarrow\!\!\$\, \mathrm{Func}(\ell, n) : \mathcal{A}_{prf}^{f(\cdot)} = 1\right]$$

$$\mathbf{Adv}^{\mathrm{prf}}_{F}(\mathcal{A}_{prf}) + \mathbf{Pr}\left[f \leftarrow\!\!\$\, \mathrm{Func}(\ell, n) : \mathcal{A}_{prf}^{f(\cdot)} = 1\right] = \mathbf{Pr}\left[d \leftarrow \{0,1\} : \mathbf{Exp}^{\text{ind-cpa-d}}_{\mathcal{SE}_2}(\mathcal{A}_{cpa}) = d\right]$$

bounding the left-hand side and using equation (10) on the right-hand side,

$$\mathbf{Adv}^{\mathrm{prf}}_{F}(\mathcal{A}_{prf}) + \frac{1}{2}\cdot(1 - \mathbf{Pr}\left[E\right]) + \mathbf{Pr}\left[E\right] \geq \frac{1}{2} + \frac{1}{2}\cdot\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}_2}(\mathcal{A}_{cpa})$$

$$\mathbf{Adv}^{\mathrm{prf}}_{F}(\mathcal{A}_{prf}) + \frac{1}{2}\cdot\mathbf{Pr}\left[E\right] \geq \frac{1}{2}\cdot\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}_2}(\mathcal{A}_{cpa})\,.$$

Furthermore it can be shown that (cf. [5, Lemma 10]):

$$\mathbf{Pr}\left[E\right] \leq \left(\frac{\mu}{n} + q\right)\left(\frac{q-1}{2^\ell}\right)\,.$$

By combining the above we get inequality (4a). Now, adversary $\mathcal{A}_{cva}$ proceeds as follows. It queries the message pair $(1^n \parallel 1^n, 1^n \parallel 0^n)$ to the left-or-right oracle, and gets an $\ell + 3n$ bit long ciphertext in return. It then takes this ciphertext, truncates the last n bits, and submits it to the validation oracle. If the oracle returns $\perp$ the adversary outputs 0 (left), else if $\lightning$ is returned it outputs 1 (right). It is easy to see that $\mathcal{A}_{cva}$ always succeeds and therefore its advantage is 1.

□

A.5 Proof of Theorem 18

The constructed scheme is similar to that of Theorem 17, except that it does not append the message with a block of 0's, and the ciphertext is additionally authenticated with a MAC. We will prove that the scheme is IND$-CVA secure in two steps. For any adversary $\mathcal{A}_{cva}$ we first construct adversary $\mathcal{A}_{uf}$ and an IND$-CPA adversary $\mathcal{A}_{cpa}$ against $\mathcal{SE}_3$. In the second step we then show how to construct adversaries $\mathcal{A}^1_{prf}$ and $\mathcal{A}^2_{prf}$ from any such IND$-CPA adversary. Combining the two steps yields the desired result.

Adversary $\mathcal{A}_{uf}$ runs $\mathcal{K}_e$ to obtain a key for F, picks a bit uniformly at random, and then runs $\mathcal{A}_{cva}$. It then uses the random bit, the PRF indexed by the generated key, and its own tagging oracle to simulate an Enc$(\cdot)$ oracle for $\mathcal{A}_{cva}$ according to the construction of Figure 10. It handles validation queries by parsing the queried ciphertext into a 'message' and a tag, and forwards the two to its verification oracle. Adversary $\mathcal{A}_{cpa}$ runs $\mathcal{A}_{cva}$, and simulates its encryption oracle using its own oracle. To all validation queries it responds with $\perp$, and it outputs whatever $\mathcal{A}_{cva}$ outputs. Note that $\mathcal{A}_{cpa}$ also makes q encryption queries totalling μ bits of plaintext. Now let W represent the event that $\mathcal{A}_{cva}$ wins the IND-CVA experiment, and let F represent the event that it makes a successful validation query. It then follows that:

$$\mathbf{Pr}\left[W\right] \leq \mathbf{Pr}\left[W \wedge \overline{F}\right] + \mathbf{Pr}\left[F\right]\,.$$

We can assume without loss of generality that $\mathcal{A}_{cva}$ never queries to its validation oracle a ciphertext that was previously returned by the encryption oracle. We can then bound each of the terms on the right-hand side of the inequality as follows. First note that $\mathcal{A}_{uf}$ provides $\mathcal{A}_{cva}$ with a perfect simulation of the IND-CVA experiment, and clearly whenever E occurs $\mathcal{A}_{uf}$ successfully forges a tag for a new message.

Contrarily if F does not occur then $\mathcal{A}_{cpa}$ simulates $\mathcal{A}_{cva}$'s environment perfectly, and thus whenever $W \wedge \overline{F}$ occurs, $\mathcal{A}_{cpa}$ wins the IND-CPA experiment. This yields:

$$\mathbf{Adv}^{\text{ind\$-cva}}_{\mathcal{SE}_3}(\mathcal{A}_{cva}) \leq \mathbf{Adv}^{\text{ind\$-cpa}}_{\mathcal{SE}_3}(\mathcal{A}_{cpa}) + \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{MA}}(\mathcal{A}_{uf})\,. \tag{14}$$

We now move to the second step of the proof and bound $\mathcal{A}_{cpa}$'s advantage. Towards this aim we define a hybrid experiment $\mathbf{ExpH}$, similar in spirit to the two IND\$-CPA experiments corresponding to each bit value. The hybrid experiment proceeds exactly as the $\mathbf{Exp}^{\text{ind\$-cpa-1}}_{\mathcal{SE}_3}$ experiment except for one detail. In the encryption oracle the intermediate string which constitutes the unauthenticated ciphertext is replaced with a uniformly random string of the same length and the MAC is then applied to this string instead. Thus we have that:

$$\begin{aligned}\mathbf{Adv}^{\text{ind\$-cpa}}_{\mathcal{SE}_3}(\mathcal{A}_{cpa}) = &\left(\Pr\left[\,\mathbf{Exp}^{\text{ind\$-cpa-1}}_{\mathcal{SE}_3}(\mathcal{A}_{cpa}) = 1\,\right] - \Pr\left[\,\mathbf{ExpH}(\mathcal{A}_{cpa}) = 1\,\right]\right)\\ &+ \left(\Pr\left[\,\mathbf{ExpH}(\mathcal{A}_{cpa}) = 1\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{ind\$-cpa-0}}_{\mathcal{SE}_3}(\mathcal{A}_{cpa}) = 1\,\right]\right)\,. \end{aligned}\tag{15}$$

Now we consider each of the above terms in the braces separately, and in each case consider $\mathcal{A}_{cpa}$'s success in distinguishing between the two experiments. For any adversary $\mathcal{A}_{cpa}$ distinguishing between the two experiments in the first term we can associate a PRF adversary $\mathcal{A}^1_{prf}$ against F. Adversary $\mathcal{A}^1_{prf}$ proceeds by running $\mathcal{K}_m$ to obtain a key for $\mathcal{MA}$, and then runs $\mathcal{A}_{cpa}$. It simulates its encryption oracle by using $\mathcal{MA}$ under the obtaineds key and its own oracle to recreate the encryption algorithm of Figure 10. Then $\mathcal{A}^1_{prf}$ outputs whatever $\mathcal{A}_{cpa}$ outputs. Note that if $\mathcal{A}^1_{prf}$'s oracle is instantiated with F, it perfectly simulates a 'real' encryption oracle for $\mathcal{A}^1_{prf}$. On the other hand if its oracle is a random function it simulates the encryption oracle of the hybrid experiment as long as it does not query the random funtion on the same input more than once. Let E_1 denote the event that $\mathcal{A}^1_{prf}$ queries its oracle on the same input more than once when simulating $\mathcal{A}_{cpa}$'s encryption oracle, let Z_b represent the event that $\mathbf{Exp}^{\text{ind\$-cpa-b}}_{\mathcal{SE}_3}(\mathcal{A}_{cpa}) = 1$, and let Z_H represent the event that $\mathbf{ExpH}(\mathcal{A}_{cpa}) = 1$. The first term in equation (15) can then be bounded as follows:

$$\begin{aligned}\mathbf{Pr}\left[Z_1\right] - \mathbf{Pr}\left[Z_H\right] &\leq \mathbf{Pr}\left[Z_1 \mid \overline{E_1}\right] - \mathbf{Pr}\left[Z_H \mid \overline{E_1}\right] + \mathbf{Pr}\left[E_1\right]\\ &\leq \mathbf{Pr}\left[K \leftarrow\!\!\$\ \mathcal{K}_e : \mathcal{A}^{1\ F_K(\cdot)}_{prf} = 1\right]\\ &\qquad - \mathbf{Pr}\left[f \leftarrow\!\!\$\ \mathrm{Func}(\ell, n) : \mathcal{A}^{1\ f(\cdot)}_{prf} = 1\right] + \mathbf{Pr}\left[E_1\right]\\ &\leq \mathbf{Adv}^{\text{prf}}_{F}(\mathcal{A}^1_{prf}) + \frac{\mu}{n}\cdot\left(\frac{q-1}{2^\ell}\right)\,.\end{aligned}\tag{16}$$

The bound on $\mathbf{Pr}\left[E_1\right]$ follows from Lemma 10 in [5]. Now for any adversary $\mathcal{A}_{cpa}$ distinguishing between the two experiments in the second term we construct a PRF adversary $\mathcal{A}^2_{prf}$ against $\mathcal{T}$. Adversary $\mathcal{A}^2_{prf}$ runs $\mathcal{A}_{cpa}$ and simulates its encryption oracle as follows. It first verifies that queried message is in the message space and outputs $\perp$ otherwise. It then samples a random string of length $\ell + |m|$, where m is the queried message, and submits it to its own oracle. It then appends the oracle's output to the random string and returns it to $\mathcal{A}_{cpa}$. Now when $\mathcal{A}^2_{prf}$'s oracle is instantiated with $\mathcal{T}$, it provides $\mathcal{A}_{cpa}$ with a perfect simulation of the hybrid experiment. Alternatively if its oracle is a random function it simulates the 'random' encryption oracle of the IND\$-CPA experiment, as long as it does not query the

same string more than once. Let E_2 denote this event, we then have that:

$$\begin{aligned}\mathbf{Pr}\left[Z_H\right] - \mathbf{Pr}\left[Z_0\right] &\leq \mathbf{Pr}\left[Z_H \mid \overline{E_2}\right] - \mathbf{Pr}\left[Z_0 \mid \overline{E_2}\right] + \mathbf{Pr}\left[E_2\right] \\ &\leq \mathbf{Pr}\left[K \leftarrow\!\!\$\, \mathcal{K}_m : \mathcal{A}_{prf}^{2\,\mathcal{T}_K(\cdot)} = 1\right] \\ &\qquad - \mathbf{Pr}\left[f \leftarrow\!\!\$\, \mathrm{Func}(*, n) : \mathcal{A}_{prf}^{2\,f(\cdot)} = 1\right] + \mathbf{Pr}\left[E_2\right] \\ &\leq \mathbf{Adv}_F^{\mathrm{prf}}(\mathcal{A}_{prf}^2) + \frac{q(q-1)}{2^{\ell+n+1}}\,. \end{aligned} \tag{17}$$

The second term on the right-hand-side of the last inequality results from a brithday bound on event E_2. Combining equations (14) (15) (16) (17) yields inequality (5a). This proves that scheme $\mathcal{SE}_3$ is IND$-CVA secure. To conclude the proof we now describe an adversary $\mathcal{A}_{sfcva}$ that breaks the IND$-sfCVA security of this scheme. Adversary $\mathcal{A}_{sfcva}$ queries two distinct messages m_1 and m_2 to its encryption oracle in this exact order, and gets in return two corresponding ciphertexts c_1 and c_2. It then makes an out-of-sync query c_2 to the validation oracle. If the oracle returns $\lightning$ it outputs 1 otherwise it outputs 0. Now if the encryption oracle returned a 'real' encryption the validation oracle will always return $\lightning$. Alternatively if c_2 is a random string the probability that the validation oracle returns $\lightning$ is bounded by $\mathbf{Adv}_{\mathcal{MA}}^{\text{uf-cma}}$ (otherwise there exists a trivial adversary against $\mathcal{MA}$). Inequality (5a) thus follows.

□

A.6 Proof of Theorem 20

We prove the first part of Theorem 20 by showing that for any $\perp \in \mathcal{S}_\perp$ and any IND$-CCA3 adversary $\mathcal{A}_{cca3}$ we can construct three adversaries $\mathcal{A}_{cpa}$, $\mathcal{A}_{int}$, $\mathcal{A}_{err}$ that correspond to the IND$-CPA, INT-CTXT*, and INV-ERR experiments respectively. Moreover whenever $\mathcal{A}_{cca3}$ is successful, then at least one of the three constructed adversaries will also be successful. Each of the three adversaries runs $\mathcal{A}_{cca3}$ and attempts to simulate its enviroment as follows. $\mathcal{A}_{cpa}$ forwards encryption queries to its own Enc$(\cdot)$ oracle and responds to decryption queries always with $\perp$. It then outputs whatever $\mathcal{A}_{cca3}$ outputs. As for $\mathcal{A}_{int}$ and $\mathcal{A}_{err}$, these respond to $\mathcal{A}_{cca3}$'s encryption queries using their own encryption oracle, and hence always returns a valid encryption. Furthermore $\mathcal{A}_{int}$ forwards any decryption queries that $\mathcal{A}_{cca3}$ makes to its Try$(\cdot)$ oracle, and always returns $\perp$. Finally $\mathcal{A}_{err}$ forwards all decryption queries to its own decryption oracle.

Now let Z_b represent the event that $\mathbf{Exp}_{\mathcal{SE},\perp}^{\text{ind-cca3-b}}(\mathcal{A}_{cca3}) = 1$. For $b = 1$ let E and F denote the respective events where $\mathcal{A}_{cca3}$ queries a ciphertext c to its decryption oracle such that Dec$\varnothing(c) \in \mathcal{S}_\perp \setminus \{\perp\}$, and Dec$\varnothing(c) \in \mathcal{M}$. We then have that:

$$\begin{aligned}\mathbf{Pr}\left[Z_1\right] &= \mathbf{Pr}\left[Z_1 \wedge \overline{F} \wedge \overline{E}\right] + \mathbf{Pr}\left[Z_1 \wedge F \wedge \overline{E}\right] + \mathbf{Pr}\left[W \wedge E\right] \\ &\leq \mathbf{Pr}\left[Z_1 \wedge \overline{F} \wedge \overline{E}\right] + \mathbf{Pr}\left[F \wedge \overline{E}\right] + \mathbf{Pr}\left[E\right]\,.\end{aligned}$$

$\mathcal{A}_{cca3}$'s advantage can then be expressed as:

$$\begin{aligned}\mathbf{Adv}_{\mathcal{SE},\perp}^{\text{ind\$-cca3}}(\mathcal{A}_{cca3}) &= \mathbf{Pr}\left[Z_1\right] - \mathbf{Pr}\left[Z_0\right] \\ &\leq (\mathbf{Pr}\left[Z_1 \wedge \overline{F} \wedge \overline{E}\right] - \mathbf{Pr}\left[Z_0\right]) + \mathbf{Pr}\left[F \wedge \overline{E}\right] + \mathbf{Pr}\left[E\right]\,.\end{aligned}$$

Now each of the three terms on the right-hand side of the last inequality can be bounded as follows. Note that $\mathcal{A}_{err}$ provides $\mathcal{A}_{cca3}$ with a perfect simulation of the IND$-CCA3 experiment for the case when $b = 1$. Thus whenever E occurs, $\mathcal{A}_{err}$ wins the INV-ERR experiment for $\perp$. On the other hand if E does not occur then $\mathcal{A}_{int}$ provides $\mathcal{A}_{cca3}$'s with a perfect simulation of the IND$-CCA3 experiment for the

case when $b = 1$. This is true until F occurs, at which point $\mathcal{A}_{int}$ wins the INT-CTXT* experiment. Finally if E and F do not occur, then $\mathcal{A}_{cpa}$ provides $\mathcal{A}_{cca3}$ with a perfect simulation of IND$-CCA3 experiment. It then follows that the first term corresponds to $\mathcal{A}_{cpa}$'s advantage. Combining the above yields inequality (6). Note that each of the three adversaries uses similar resources as $\mathcal{A}_{cca3}$.

The second part of the theorem is easier to prove. Adversary $\mathcal{A}^1_{cca3}$ runs $\mathcal{A}'_{cpa}$, forwards encryption queries to its own Enc$(\cdot)$ oracle and outputs whatever $\mathcal{A}'_{cpa}$ outputs. Since this provides $\mathcal{A}'_{cpa}$ with a perfect simulation of its environment, it follows that they both have the same advantage. Adversary $\mathcal{A}^2_{cca3}$ runs $\mathcal{A}'_{int}$ and simulates its oracles using its Enc$(\cdot)$ oracle and its Dec$\varnothing(\cdot)$ oracle. If at any point $\mathcal{A}'_{int}$ queries a ciphertext (not previously returned by the encryption oracle) wich decrypts successfully, then $\mathcal{A}^2_{cca3}$ halts and outputs 1. Otherwise it outputs a uniformly-random bit. Note that when $b = 1$ $\mathcal{A}^2_{cca3}$ provides $\mathcal{A}'_{int}$ with a perfect simulation of its environment, but when $b = 0$ $\mathcal{A}'_{int}$ has zero probability of winning. Inequality (7b) then follows from:

$$\begin{aligned}
\mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}^2_{cca3}) &= \mathbf{Exp}^{\text{ind\$-cca3-1}}_{\mathcal{SE},\perp}(\mathcal{A}^2_{cca3}) - \mathbf{Exp}^{\text{ind\$-cca3-0}}_{\mathcal{SE},\perp}(\mathcal{A}^2_{cca3}) \\
&= \frac{1}{2} \cdot (1 - \mathbf{Adv}^{\text{int-ctxt}*}_{\mathcal{SE}}(\mathcal{A}'_{int})) + \mathbf{Adv}^{\text{int-ctxt}*}_{\mathcal{SE}}(\mathcal{A}'_{int}) - \frac{1}{2} \\
&= \frac{1}{2} \cdot \mathbf{Adv}^{\text{int-ctxt}*}_{\mathcal{SE}}(\mathcal{A}'_{int}) \,.
\end{aligned}$$

Adversary $\mathcal{A}^3_{cca3}$ proceeds in a similar fashion. It runs $\mathcal{A}'_{err}$ and simulates its oracles using its Enc$(\cdot)$ oracle and its Dec$\varnothing(\cdot)$ oracle. If at any point $\mathcal{A}'_{int}$ queries a ciphertext which returns an error symbol in $\mathcal{S}_\perp \setminus \{\perp\}$, then $\mathcal{A}^3_{cca3}$ halts and outputs 1. Otherwise it outputs a uniformly-random bit. Again when $b = 1$ $\mathcal{A}^3_{cca3}$ provides $\mathcal{A}'_{err}$ with a perfect simulation of its environment, but when $b = 0$ $\mathcal{A}'_{err}$ has zero probability of winning. Inequality (7c) then follows as in the previous case. Finally note that in all three cases the respective constructed IND$-CCA3 adversaries use the same resources as $\mathcal{A}'_{cpa}$, $\mathcal{A}'_{int}$ and $\mathcal{A}'_{err}$. □

A.7 Proof of Theorem 23

Correctness of the constructed scheme follows easily from the correctness of its constituent schemes, and we thus proceed to prove its security. We start by proving inequality (8).

Adversary $\mathcal{A}_{cpa}$ simply runs $\mathcal{K}_m$ to get a key for the MAC and then runs $\mathcal{A}_{cva}$. It answers its left-or-right encryption queries by first encoding both messages, it then submits them to its own oracle, computes a tag for the resulting ciphertext and retuns the ciphertext cocatenated with the tag. Validation queries are handled by extracting the tag from the submitted ciphertext, verifiying the tag on the remaining string using the derived MAC key, and returning the output to $\mathcal{A}_{cva}$. If the submitted ciphertext is shorter than ℓ_{tag} it returns $\perp_0$ instead. It then outputs whatever $\mathcal{A}_{cva}$ outputs.

Adversary $\mathcal{A}^1_{suf}$ runs $\mathcal{K}_e$ to get an encryption key, picks a bit uniformly at random, and uses these together with its tagging oracle to simulate $\mathcal{A}_{cva}$'s left-or-right oracle. It handles decryption queries by extracting the tag from the submitted ciphertext, and submitting the tag together with the remaining string to its verification oracle, and forwards the output to $\mathcal{A}_{cva}$. If the submitted ciphertext is shorter than ℓ_{tag} it returns $\perp_0$ instead.

Now let W represent the event $\mathbf{Exp}^{\text{ind-cva-b}}_{\mathcal{EEM}}(\mathcal{A}_{cva}) = b$ where b is picked uniformly at random. Let E represent the event that $\mathcal{A}_{cva}$ makes a validation query which returns an error message in $\overline{\mathcal{S}_\perp} \setminus \mathcal{Q}_\perp$. We then have that:

$$\begin{aligned}
\mathbf{Pr}\left[W\right] &= \mathbf{Pr}\left[W \wedge \overline{E}\right] + \mathbf{Pr}\left[W \wedge E\right] \\
&\leq \mathbf{Pr}\left[W \wedge \overline{E}\right] + \mathbf{Pr}\left[E\right] \,.
\end{aligned}$$

We bound each term on the right-hand side of the last inequality as follows. Note that $\mathcal{A}^1_{suf}$ simulates $\mathcal{A}_{cva}$'s environment perfectly until one of $\mathcal{A}_{cva}$'s queries results in a forgery for $\mathcal{A}^1_{suf}$. It then follows

that whenever E occurs, $\mathcal{A}^1_{suf}$ wins the SUF-CMA experiment. On the other hand if E does not occur, then $\mathcal{A}_{cpa}$'s simulation of $\mathcal{A}_{cva}$'s environment is perfect. Consequently whenever event $W \wedge \overline{E}$ occurs, $\mathcal{A}_{cpa}$ wins the IND-CPA experiment. Equation (8) then follows by combining the above and using equation (10).

Adversary $\mathcal{A}^2_{suf}$ runs $\mathcal{K}_e$ to get an encryption key, picks a bit uniformly at random, and uses these together with its tagging oracle to simulate $\mathcal{A}_{int}$'s encryption oracle. For each encryption query that $\mathcal{A}_{int}$'s submits, it first encodes the message and then encrypts it with $\mathcal{E}_K$. It then obtains a tag for the resulting ciphertext from its own oracle, and returns the ciphertext concatenated with the tag. Queries to the $\mathsf{Try}(\cdot)$ oracle are handled by extracting a tag from the ciphertext, and submitting tag together with the remaining string to its verification oracle, and the output is returned to $\mathcal{A}_{int}$. On the other hand if the submitted ciphertext cannot be parsed $\perp_0$ is returned. Note that $\mathcal{A}^2_{suf}$ provides $\mathcal{A}_{int}$ with a perfect simulation of the INT-CTXT experiment until the point at which $\mathcal{A}_{int}$ makes a successful try query. Moreover whenever $\mathcal{A}_{int}$ forges a ciphertext, $\mathcal{A}^2_{suf}$'s corresponding verification query will also constitute a forgery. Inequality (9) thus follows.

□