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, Atlanta, USA
[2] Royal Holloway, University of London, London, UK
kenny.paterson@rhul.ac.uk
[3] University of Bristol, Bristol, UK

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 [9], 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 [16,21] and called *reaction attacks*. Vaudenay then showed that similar issues can arise in the symmetric setting [27], and his ideas were extended to produce significant attacks against (among others) SSL/TLS [11,23], IPsec [12,13], ASP.NET [14], XML encryption [19] 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].

S. Moriai (Ed.): FSE 2013, LNCS 8424, pp. 367–390, 2014.
DOI: 10.1007/978-3-662-43933-3_19,

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?

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 [20], and one might reasonably conclude that its symmetric encryption scheme is IND-CCA secure. Yet Canvel et al. [11] 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 [27]. 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 [23]. Even there, however, it was necessary to assume that all decryption failures are indistinguishable (since, otherwise, attacks like those of [2,3,11,27] are possible). A similar story could be told for MAC-then-encryption configurations of IPsec, to which the theory in [20] and the attacks of [13] 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, [22,24]), 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 [10] 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 [10] breaks down: in the passage from the INT-CTXT security game to the IND-CPA security game, the simulation in [10] 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 [26], 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 [10] and Krawczyk [20], 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 [25], and for the stateful setting introduced by Bellare et al. [8]. Many of these additional results follow rather straightforwardly, but we consider it valuable to include them for completeness.

For reasons of space, all proofs are deferred to the full version [6].

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 $y \leftarrow \mathcal{A}(x_1, x_2, \ldots)$ to denote executing $\mathcal{A}$ with fresh coins on inputs $x_1, x_2, \ldots$ 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 $\langle \langle w \rangle^{-1} + i \mod 2^\ell \rangle_\ell$. 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 Fig. 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.

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 an element from its domain and maps it to some

$\underline{\mathbf{Exp}^{\text{suf-cma}}_{\mathcal{SE}}(\mathcal{A})}$
$K \leftarrow \mathcal{K}$
$\mathtt{L} \leftarrow \emptyset, \mathsf{win} \leftarrow 0$
$\mathcal{A}^{\mathsf{Tag}(\cdot),\mathsf{Ver}(\cdot,\cdot)}$
return win

$\underline{\mathsf{Tag}(m)}$
$\tau \leftarrow \mathcal{T}_K(m)$
$\mathtt{L} \leftarrow \mathtt{L} \cup (m,\tau)$
return τ

$\underline{\mathsf{Ver}(m,\tau)}$
$v \leftarrow \mathcal{V}_K(m,\tau)$
if $v \notin \mathcal{Q}_\perp$ **and** $(m,\tau) \notin \mathtt{L}$
  then $\mathsf{win} \leftarrow 1$
return v

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

element in its codomain. The *decoding* algorithm $\mathcal{DC}$ is deterministic and takes an element from its codomain and returns either an element in its domain or an error symbol from its error space. The scheme must be correct, i.e. for every element m in its domain it holds with probability 1 that $\mathcal{DC}(\mathcal{EC}(m)) = m$.

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 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 Fig. 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

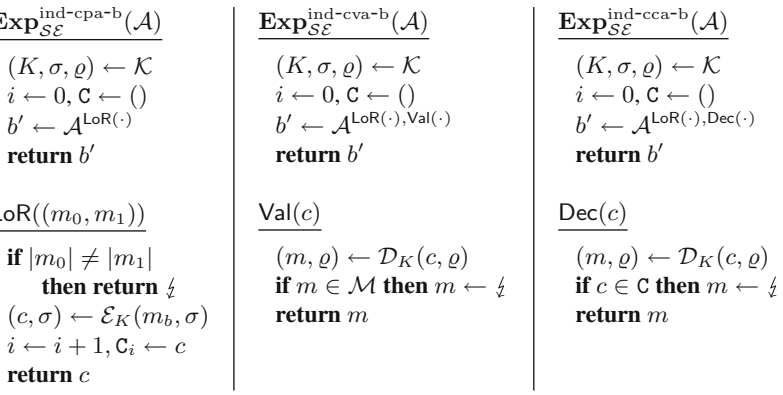

Fig. 2. IND-ATK experiments for symmetric encryption schemes.

by Rogaway [25]. 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* atk $\in \{\text{cpa}, \text{cva}, \text{cca}\}$ *as shown in Fig. 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:

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

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 et al. [8] introduced a notion called IND-sfCCA. This notion implies IND-CCA security and additionally protects against replay and reordering of ciphertexts. We recall

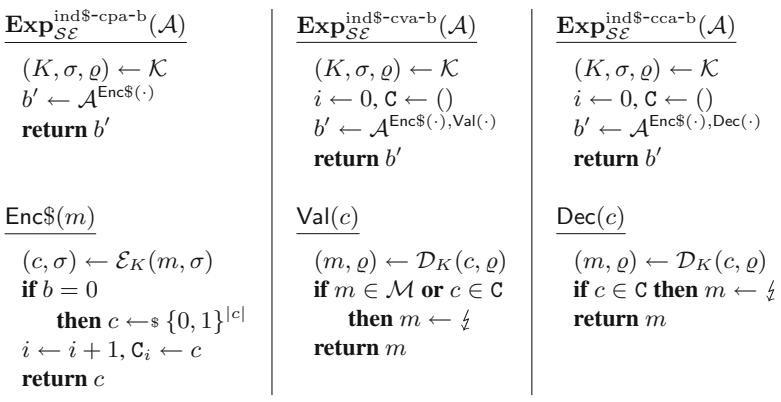

Fig. 3. IND$-ATK experiments for symmetric encryption schemes.

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 Fig. 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 $\not\perp$ 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.

Ciphertext integrity. We define ciphertext integrity analogously to Bellare and Namprempre [10], and we also consider its stateful variant [8] 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

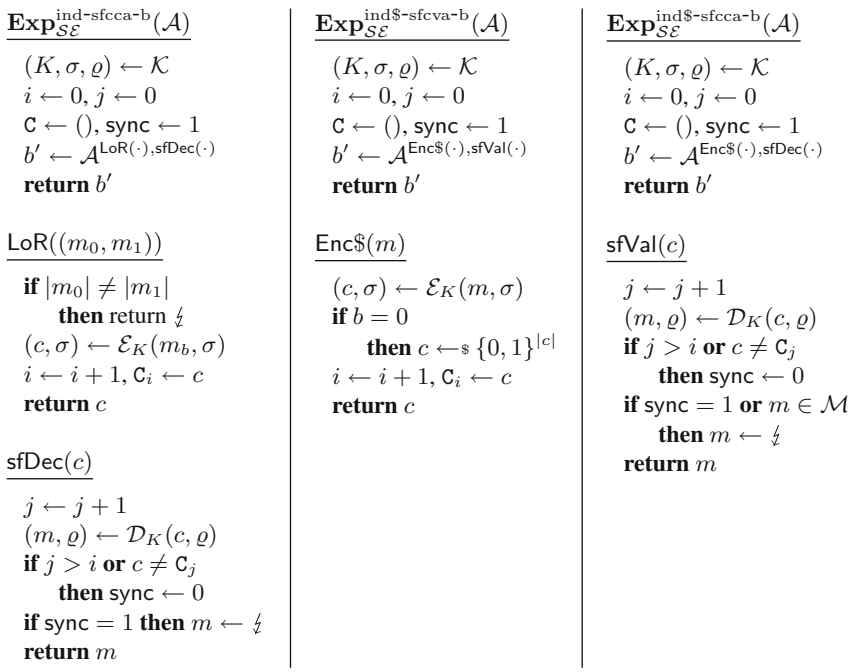

Fig. 4. Stateful indistinguishability experiments for symmetric encryption schemes.

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 Sect. 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* atk $\in \{\text{ctxt}, \text{ctxt}*, \text{sfctxt}, \text{sfctxt}*\}$ *as shown in Fig. 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*

$\underline{\mathbf{Exp}^{\text{int-ctxt}}_{\mathcal{SE}}(\mathcal{A})}$
$(K, \sigma, \varrho) \leftarrow \mathcal{K}$
$i \leftarrow 0, \mathtt{C} \leftarrow (), \mathsf{win} \leftarrow 0$
$\mathcal{A}^{\mathsf{Enc}(\cdot),\mathsf{Try}(\cdot)}$
return win

$\underline{\mathbf{Exp}^{\text{int-sfctxt}}_{\mathcal{SE}}(\mathcal{A})}$
$(K, \sigma, \varrho) \leftarrow \mathcal{K}$
$i \leftarrow 0, j \leftarrow 0, \mathtt{C} \leftarrow ()$
$\mathsf{sync} \leftarrow 1, \mathsf{win} \leftarrow 0$
$\mathcal{A}^{\mathsf{Enc}(\cdot),\mathsf{sfTry}(\cdot)}$
return win

$\underline{\mathbf{Exp}^{\text{int-ctxt}*}_{\mathcal{SE}}(\mathcal{A})}$
$(K, \sigma, \varrho) \leftarrow \mathcal{K}$
$i \leftarrow 0, \mathtt{C} \leftarrow (), \mathsf{win} \leftarrow 0$
$\mathcal{A}^{\mathsf{Enc}(\cdot),\mathsf{Try}^*(\cdot)}$
return win

$\underline{\mathbf{Exp}^{\text{int-sfctxt}*}_{\mathcal{SE}}(\mathcal{A})}$
$(K, \sigma, \varrho) \leftarrow \mathcal{K}$
$i \leftarrow 0, j \leftarrow 0, \mathtt{C} \leftarrow ()$
$\mathsf{sync} \leftarrow 1, \mathsf{win} \leftarrow 0$
$\mathcal{A}^{\mathsf{Enc}(\cdot),\mathsf{sfTry}^*(\cdot)}$
return win

$\underline{\mathsf{Enc}(m)}$
$(c, \sigma) \leftarrow \mathcal{E}_K(m, \sigma)$
$i \leftarrow i + 1, \mathtt{C}_i \leftarrow c$
return c

$\underline{\mathsf{Try}(c)}$
$(m, \varrho) \leftarrow \mathcal{D}_K(c, \varrho)$
if $c \notin \mathtt{C}$ **and** $m \notin \mathcal{S}_\perp$
  then $\mathsf{win} \leftarrow 1$
if $m \notin \mathcal{S}_\perp$ **then** $m \leftarrow \lightning$
return m

$\underline{\mathsf{sfTry}(c)}$
$j \leftarrow j + 1$
$(m, \varrho) \leftarrow \mathcal{D}_K(c, \varrho)$
if $j > i$ **or** $c \neq \mathtt{C}_j$
  then $\mathsf{sync} \leftarrow 0$
if $\mathsf{sync} = 0$ **and** $m \notin \mathcal{S}_\perp$
  then $\mathsf{win} \leftarrow 1$
if $m \notin \mathcal{S}_\perp$ **then** $m \leftarrow \lightning$
return m

$\underline{\mathsf{Try}^*(c)}$
$(m, \varrho) \leftarrow \mathcal{D}_K(c, \varrho)$
if $c \notin \mathtt{C}$ **and** $m \notin \mathcal{S}_\perp$
  then $\mathsf{win} \leftarrow 1$
if $m \in \mathcal{S}_\perp$ **then** $m \leftarrow \perp$
else $m \leftarrow \lightning$
return m

$\underline{\mathsf{sfTry}^*(c)}$
$j \leftarrow j + 1$
$(m, \varrho) \leftarrow \mathcal{D}_K(c, \varrho)$
if $j > i$ **or** $c \neq \mathtt{C}_j$
  then $\mathsf{sync} \leftarrow 0$
if $\mathsf{sync} = 0$ **and** $m \notin \mathcal{S}_\perp$
  then $\mathsf{win} \leftarrow 1$
if $m \in \mathcal{S}_\perp$ **then** $m \leftarrow \perp$
else $m \leftarrow \lightning$
return m

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

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.*

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* $\mathcal{A}$*,*

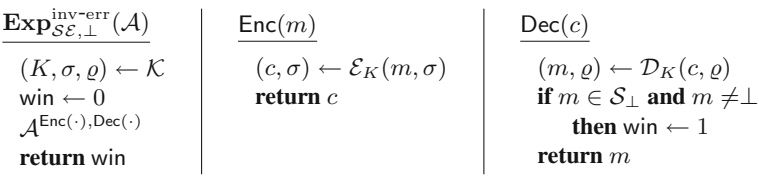

Fig. 6. INV-ERR experiment for symmetric encryption schemes.

define the experiment $\mathbf{Exp}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A})$ *as shown in Fig. 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.*

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 [15,17,18]. 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 1

$$\mathsf{IND-sfCCA} \longrightarrow \mathsf{IND-CCA} \longrightarrow \mathsf{IND-CVA} \longrightarrow \mathsf{IND-CPA}$$

$$\mathsf{IND\$-sfCCA} \longrightarrow \mathsf{IND\$-CCA} \longrightarrow \mathsf{IND\$-CVA} \longrightarrow \mathsf{IND\$-CPA}$$

$$\mathsf{IND\$-sfCCA} \longrightarrow \mathsf{IND\$-sfCVA} \longrightarrow \mathsf{IND\$-CVA}$$

$$\mathsf{INT-sfCTXT} \longrightarrow \mathsf{INT-CTXT} \longrightarrow \mathsf{INT-CTXT}^*$$

$$\mathsf{INT-sfCTXT} \longrightarrow \mathsf{INT-sfCTXT}^* \longrightarrow \mathsf{INT-CTXT}^*$$

Revisiting classic relations. If a symmetric encryption scheme that only supports a single possible error symbol satisfies both passive confidentiality

(IND-CPA) and integrity of ciphertexts, then it offers confidentiality against chosen-ciphertext attacks [8,10]. Often, when analysing a particular scheme, its chosen-plaintext security and ciphertext integrity are proven first, and then the results of [8,10] are used to guarantee chosen-ciphertext security. Indeed, the combination of IND-CPA and INT-sfCTXT (or their stateful versions) has come to be the accepted security notion for symmetric encryption. We proceed to re-examine the classic relations from [8,10] in the context of encryption schemes with multiple error messages.

The following theorem serves as the basis for the two separations in Corollaries 1 and 2, showing that the classic relations no longer hold for multiple-error schemes.

We point out that in proving the separations, we adopt the stronger interpretations of ciphertext integrity so as to avoid any ambiguity in the results.

Theorem 1 (IND-CPA $\wedge$ INT-sfCTXT$\not\rightarrow$ 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 $\overline{\mathcal{SE}}_1$ having message space $\{0,1\}^{n-\ell_{tag}}$ and error space $\{\perp_0, \perp_1\}$ shown in Fig. 7. For any* IND-CPA *adversary $\mathcal{A}_{cpa}$ and any* INT-sfCTXT *adversary $\mathcal{A}_{int}$ against $\overline{\mathcal{SE}}_1$, 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:*

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

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

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

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

Combining Theorem 1 and Proposition 1 yields the following two separations corresponding to the aforementioned relations from [10] and [8].

Corollary 1 (IND-CPA $\wedge$ INT-CTXT$\not\rightarrow$ 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 2 (IND-CPA $\wedge$ INT-sfCTXT$\not\rightarrow$ 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.*

Note that in proving Theorem 1 we resorted to a stateful scheme. Only a stateful scheme can be INT-sfCTXT secure, and therefore the counterexample

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

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

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

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

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

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 $\overline{\mathcal{SE}}_1$ of Theorem 1.

used to prove Corollary 2 needs to be stateful. The same cannot be said however about the separation in Corollary 1, 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 [10] and [8] 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 [10] to schemes with multiple errors.

Theorem 2 (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 2

$$\begin{aligned}
\text{IND-CVA} \ \wedge\ \text{INT-sfCTXT} &\longrightarrow \text{IND-sfCCA}\\
\text{IND\$-CVA} \ \wedge\ \text{INT-CTXT} &\longrightarrow \text{IND\$-CCA}\\
\text{IND\$-sfCVA} \ \wedge\ \text{INT-sfCTXT} &\longrightarrow \text{IND\$-sfCCA}
\end{aligned}$$

Necessity of strong ciphertext integrity. The above relations can be seen as strengthened variants of the relations from [10] and [8], 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 1 and 2 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 3 answers this question by means of a separation, proving that strong ciphertext integrity is necessary for Theorem 2 to hold.

Theorem 3 (IND-CVA $\wedge$ INT-CTXT* $\not\rightarrow$ 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 ℓ. Consider the scheme $\overline{\mathcal{SE}}_2$ having message space $\mathcal{M}$ and error space $\{\perp_0, \perp_1\}$ shown in Fig. 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:*

$$\mathbf{Adv}^{\text{ind-cva}}_{\overline{\mathcal{SE}}_2}(\mathcal{A}_{cva}) \leq \mathbf{Adv}^{\text{ind-cva}}_{\mathcal{SE}}(\mathcal{A}^1_{cva}) + \frac{1}{2} \cdot \mathbf{Adv}^{\text{ind-cva}}_{\mathcal{SE}}(\mathcal{A}^2_{cva}) + \frac{q_e}{|\mathcal{M}|}\,, \tag{3a}$$

$$\mathbf{Adv}^{\text{int-ctxt}*}_{\overline{\mathcal{SE}}_2}(\mathcal{A}_{int}) \leq \mathbf{Adv}^{\text{int-ctxt}*}_{\mathcal{SE}}(\mathcal{A}^1_{int}) + \frac{q_t}{|\mathcal{M}|}\,. \tag{3b}$$

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:

$$\mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{SE}}_2}(\mathcal{A}_{cca}) = 1. \tag{3c}$$

Theorem 3 also serves as a separation between INT-CTXT* and INT-CTXT, showing that the latter is strictly stronger. Separations similar to that of Theorem 3 corresponding to the relations of Proposition 2 can also be established.

Proposition 3

$$\begin{aligned} 2\mathsf{IND}-\mathsf{CVA} \wedge \mathsf{INT}-\mathsf{sfCTXT}^* &\not\rightarrow \mathsf{IND}-\mathsf{sfCCA} \\ \mathsf{IND\$}-\mathsf{CVA} \wedge \mathsf{INT}-\mathsf{CTXT}^* &\not\rightarrow \mathsf{IND\$}-\mathsf{CCA} \\ \mathsf{IND\$}-\mathsf{sfCVA} \wedge \mathsf{INT}-\mathsf{sfCTXT}^* &\not\rightarrow \mathsf{IND\$}-\mathsf{sfCCA} \end{aligned}$$

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

$(K, \sigma, \varrho) \leftarrow \mathcal{K}$
$m^* \leftarrow\!\!\$\ \mathcal{M}$
$(c^*, \sigma) \leftarrow \mathcal{E}_K(m^*, \sigma)$
$K_0 \leftarrow (K, m^*, c^*)$
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 $\overline{\mathcal{SE}}_2$ of Theorem 3.

5 Further Relations and the IND$-CCA3 Notion

Authenticated-encryption security. Following the work of Bellare and Namprempre [10], chosen-plaintext security and ciphertext integrity were identified as the two security goals for symmetric encryption. Rogaway and Shrimpton [26] 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 4 we show that this characterisation is equivalent to the combination of chosen-plaintext security, weak chosen ciphertext integrity, and error invariance.

Definition 9 (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 Fig. 9. 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.

Theorem 4 (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:*

$$\mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}_{cca3}) \leq \mathbf{Adv}^{\text{ind\$-cpa}}_{\mathcal{SE}}(\mathcal{A}_{cpa}) + \mathbf{Adv}^{\text{int-ctxt*}}_{\mathcal{SE}}(\mathcal{A}_{int}) + \mathbf{Adv}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A}_{err}). \quad (4)$$

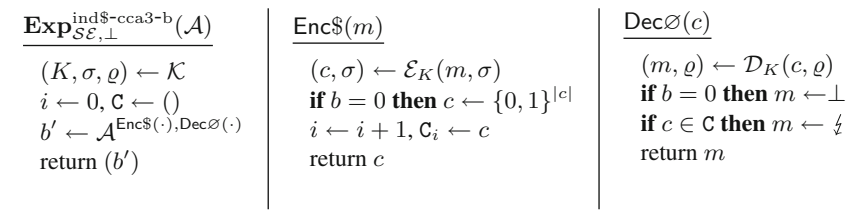

Fig. 9. IND$-CCA3 experiment for multiple-error symmetric encryption schemes.

– *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:*

$$\mathbf{Adv}^{\text{ind\$-cpa}}_{\mathcal{SE}}(\mathcal{A}'_{cpa}) \leq \mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}^1_{cca3}), \tag{5a}$$

$$\mathbf{Adv}^{\text{int-ctxt*}}_{\mathcal{SE}}(\mathcal{A}'_{int}) \leq 2 \cdot \mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}^2_{cca3}), \tag{5b}$$

$$\mathbf{Adv}^{\text{inv-err}}_{\mathcal{SE},\perp}(\mathcal{A}'_{err}) \leq 2 \cdot \mathbf{Adv}^{\text{ind\$-cca3}}_{\mathcal{SE},\perp}(\mathcal{A}^3_{cca3}). \tag{5c}$$

It can be similarly shown that:

Proposition 4 IND-CPA $\wedge$ INT-CTXT* $\wedge$ INV-ERR $\rightleftarrows$ IND$-CCA3.

The question remains whether IND$-CCA3 security guarantees IND$-CCA security in the multiple error setting, which is the ultimate target security notion. Proposition 5 tells us that this is indeed the case. In fact it says something stronger, in that it relates IND$-CCA3 to the security notions from Proposition 2.

Proposition 5 IND$-CCA3 $\longrightarrow$ IND$-CVA $\wedge$ INT-CTXT $\longrightarrow$ IND$-CCA.

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

The works of Bellare and Namprempre [10] and Krawczyk [20] provide formal evidence for preferring Encrypt-then-MAC (EtM) over other generic compositions like MAC-then-encrypt (MtE). However we believe that the merits of EtM as a generic composition technique go beyond the implications of their work. By combining results from [20] and [7], we know that MtE is actually IND-CCA secure when instantiated with CBC or counter-mode encryption. Thus the analysis of [10,20] does not explain why EtM should be more secure than MtE when both are instantiated with CBC or counter-mode encryption. Nonetheless practical cryptosystems (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, the attacks in [2,3,11,13] exploit features of the encoding schemes used in specific MtE constructions and the fact that an adversary can distinguish among distinct decryption failures. Neither of these aspects were considered in [10]. Reconsidering the generic compositions in the light of

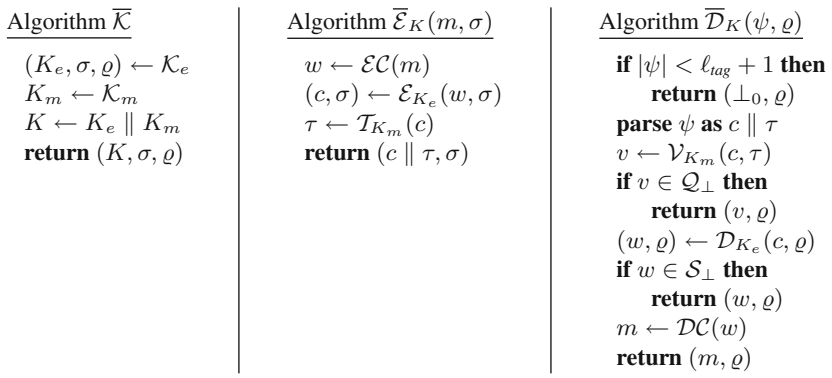

Fig. 10. The generic Encode-then-Encypt-then-MAC composition $\overline{\mathcal{SE}}_{EEM}$ with distinguishable decryption failures.

multiple-error messages (or equivalently distinguishable decryption failures) provides new formal grounds for preferring the EtM composition. More specifically we consider an encode-then-encrypt-then-MAC (EEM) composition to account for the pre-processing (such as padding) that is common in practical schemes. The EEM composition is specified in Fig. 10. Theorem 5 shows that EEM is a *robust* composition, in the sense that it provides IND-CVA and INT-CTXT security, and therefore IND-CCA security, in the multiple-error setting, irrespective of the encoding scheme used (and the error messages it returns) and the error messages that the encryption component may return, as long as the encryption component is IND-CPA and the MAC is SUF-CMA. In fact, we can prove that EEM provides IND-CCA3 security if its MAC component only has a single error message.

Theorem 5 (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 10 then defines a symmetric encryption scheme $\overline{\mathcal{SE}}_{EEM}$ 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 $\overline{\mathcal{SE}}_{EEM}$, there exist adversaries $\mathcal{A}_{cpa}$, $\mathcal{A}^1_{suf}$, and $\mathcal{A}^2_{suf}$ such that:*

$$\mathbf{Adv}^{\text{ind-cva}}_{\overline{\mathcal{SE}}_{EEM}}(\mathcal{A}_{cva}) \leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(\mathcal{A}_{cpa}) + \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(\mathcal{A}^1_{suf}), \tag{6}$$

$$\mathbf{Adv}^{\text{int-ctxt}}_{\overline{\mathcal{SE}}_{EEM}}(\mathcal{A}_{int}) \leq \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(\mathcal{A}^2_{suf}). \tag{7}$$

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

It is instructive to consider some distinguishable decryption failure attacks that have been discovered on instantiations of the MAC-then-Encode-then-Ecrypt

(MEE) composition, in order to see how such implementation flaws are captured by our treatment. The attacks on TLS [11] and on DTLS [2] use timing differences to distinguish a MAC failure from a padding failure. In the case of IPsec [13], the encoding 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 attack. The recent Lucky 13 attack on TLS [3] exploits timing differences in the verification algorithm of HMAC. 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 [27]. This timing channel can be modelled in our framework by transforming HMAC into a multiple-error MAC. Then the error messages that 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 5, an EEM composition would not be compromised by such an implementation flaw.

7 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 2 does not reduce to the relation of Bellare and Namprempre from [10].

Theorem 6 (IND-CPA $\wedge$ INV-ERR $\not\rightarrow$ IND-CVA). *Let $F : \mathcal{K}_e \times \{0,1\}^\ell \to \{0,1\}^n$ be a pseudorandom function, where ℓ is sufficiently large. Then the symmetric encryption scheme $\overline{\mathcal{SE}}_3$ having message space $\cup_{k\geq 1}\{0,1\}^{nk}$ and error space $\{\perp\}$ shown in Fig. 11 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:*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{SE}}_3}(\mathcal{A}_{cpa}) \leq 2 \cdot \mathbf{Adv}^{\text{prf}}_{F}(\mathcal{A}_{prf}) + \left(\frac{\mu}{n} + q\right)\left(\frac{q-1}{2^\ell}\right). \tag{8a}$$

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

$$\mathbf{Adv}^{\text{ind-cva}}_{\overline{\mathcal{SE}}_3}(\mathcal{A}_{cva}) = 1. \tag{8b}$$

In Sect. 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 7 settles this in the negative.

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

$K \leftarrow\!\!\$\ \mathcal{K}_e$
$\sigma \leftarrow \varepsilon,\ \varrho \leftarrow \varepsilon$
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** $m_1 \parallel \ldots \parallel m_p$
$m_{p+1} \leftarrow 0^n,\ c_0 \leftarrow\!\!\$\ \{0,1\}^\ell$
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. 11. The scheme $\overline{\mathcal{SE}}_3$ of Theorem 6.

Theorem 7 (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 $\overline{\mathcal{SE}}_4$ having message space $\cup_{k \geq 1}\{0,1\}^{nk}$ and error space $\{\perp\}$ shown in Fig. 12. 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:*

$$\mathbf{Adv}^{\text{ind\$-cva}}_{\overline{\mathcal{SE}}_4}(\mathcal{A}_{cva}) \leq \mathbf{Adv}^{\text{prf}}_{F}(\mathcal{A}^1_{prf}) + \mathbf{Adv}^{\text{prf}}_{\mathcal{T}}(\mathcal{A}^2_{prf}) + \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{MA}}(\mathcal{A}_{uf}) + \frac{\mu}{n} \cdot \left(\frac{q-1}{2^\ell}\right) + \frac{q(q-1)}{2^{\ell+n+1}}\,.$$

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

$$\mathbf{Adv}^{\text{ind\$-sfcva}}_{\overline{\mathcal{SE}}_4}(\mathcal{A}_{sfcva}) = 1 - \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{MA}}(\mathcal{A}'_{uf})\,. \tag{9a}$$

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

$K_e \leftarrow\!\!\$\ \mathcal{K}_e$
$K_m \leftarrow\!\!\$\ \mathcal{K}_m$
$K \leftarrow K_e \parallel K_m$
$\sigma \leftarrow \varepsilon,\ \varrho \leftarrow \varepsilon$
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** $m_1 \parallel \ldots \parallel m_p$
$c_0 \leftarrow\!\!\$\ \{0,1\}^\ell$
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. 12. The scheme $\overline{\mathcal{SE}}_4$ of Theorem 7.

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: CT-ISG 36566D3. 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. Albrecht, M.R., Paterson, K.G., Watson, G.J.: Plaintext recovery attacks against SSH. In: IEEE Symposium on Security and Privacy, pp. 16–26. IEEE Computer Society (2009)
2. AlFardan, N.J., Paterson, K.G.: Plaintext-recovery attacks against datagram TLS. In: Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS 2012)
3. AlFardan, N.J., Paterson, K.G.: Lucky thirteen: breaking the TLS and DTLS record protocols. In: IEEE Symposium on Security and Privacy 2013. http://www.isg.rhul.ac.uk/tls/TLStiming.pdf (To appear)
4. Bauer, A., Coron, J.-S., Naccache, D., Tibouchi, M., Vergnaud, D.: On the broadcast and validity-checking security of PKCS#1 v1.5 encryption. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 1–18. Springer, Heidelberg (2010)
5. Bellare, M., Desai, A., Jokipii, E., Rogaway. P.: A concrete security treatment of symmetric encryption. In: Proceedings of 38th Annual Symposium on Foundations of Computer Science (FOCS 1997), pp. 394–403. IEEE (1997)
6. Boldyreva, A., Degabriele, J.P., Paterson, K.G., Stam, M.: On symmetric encryption with distinguishable decryption failures. IACR Cryptology ePrint Archive. http://eprint.iacr.org (full version of this paper)
7. Bellare, M., Goldreich, O., Mityagin, A.: The power of verification queries in message authentication and authenticated encryption. IACR Cryptology ePrint Archive. http://eprint.iacr.org/2004/309
8. Bellare, M., Kohno, T., Namprempre, C.: Breaking and provably repairing the SSH authenticated encryption scheme: a case study of the encode-then-encrypt-and-MAC paradigm. ACM Trans. Inf. Syst. Secur. **7**(2), 206–241 (2004)
9. Bleichenbacher, D.: Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 1–12. Springer, Heidelberg (1998)
10. Bellare, M., Namprempre, C.: Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000)
11. Canvel, B., Hiltgen, A.P., Vaudenay, S., Vuagnoux, M.: Password interception in a SSL/TLS channel. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 583–599. Springer, Heidelberg (2003)
12. Degabriele, J.P., Paterson, K.G.: Attacking the IPsec standards in encryption-only configurations. In: IEEE Symposium on Security and Privacy, pp. 335–349. IEEE Computer Society (2007)
13. Degabriele, J.P., Paterson, K.G.: On the (in)security of IPsec in MAC-then-encrypt configurations. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM Conference on Computer and Communications Security, pp. 493–504. ACM (2010)

14. Duong, T., Rizzo, J.: Cryptography in the web: the case of cryptographic design flaws in ASP.NET. In: IEEE Symposium on Security and Privacy, pp. 481–489. IEEE Computer Society (2011)
15. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
16. Hall, C., Goldberg, I., Schneier, B.: Reaction attacks against several public-key cryptosystem. In: Varadharajan, V., Mu, Y. (eds.) ICICS 1999. LNCS, vol. 1726, pp. 2–12. Springer, Heidelberg (1999)
17. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
18. Impagliazzo, R., Luby, M.: One-way functions are essential for complexity based cryptography (extended abstract). In: Proceedings of 30th Annual Symposium on Foundations of Computer Science (FOCS 1989), pp. 230–235. IEEE (1989)
19. Jager, T., Somorovsky, J.: How to break XML encryption. In: Chen, Y., Danezis, G., Shmatikov, V. (eds.) ACM Conference on Computer and Communications Security, pp. 413–422. ACM (2011)
20. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: how secure is SSL?). In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 310–331. Springer, Heidelberg (2001)
21. Manger, J.: A chosen ciphertext attack on RSA optimal asymmetric encryption padding (OAEP) as standardized in PKCS #1 v2.0. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 230–238. Springer, Heidelberg (2001)
22. Paterson, K.G., Watson, G.J.: Plaintext-dependent decryption: a formal security treatment of SSH-CTR. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 345–361. Springer, Heidelberg (2010)
23. Paterson, K.G., Ristenpart, T., Shrimpton, T.: Tag size *does* matter: attacks and proofs for the TLS record protocol. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 372–389. Springer, Heidelberg (2011)
24. Paterson, K.G., Watson, G.J.: Authenticated-encryption with padding: a formal security treatment. In: Naccache, D. (ed.) Cryphtography and Security: From Theory to Applications. LNCS, vol. 6805, pp. 83–107. Springer, Heidelberg (2012)
25. Rogaway, P.: Nonce-based symmetric encryption. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 348–359. Springer, Heidelberg (2004)
26. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (2006)
27. Vaudenay, S.: Security flaws induced by CBC padding - applications to SSL, IPSEC, WTLS. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 534–546. Springer, Heidelberg (2002)