A preliminary version of this paper appears in *Advances in Cryptology – ASIACRYPT '00*, Lecture Notes in Computer ScienceVol. ??, T. Okamoto ed., Springer-Verlag, 2000. This is the full version.

Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm

Mihir Bellare* Chanathip Namprempre†

September 25, 2000

Abstract

We consider two possible notions of authenticity for symmetric encryption schemes, namely integrity of plaintexts and integrity of ciphertexts, and relate them to the standard notions of privacy for symmetric encryption schemes by presenting implications and separations between all notions considered. We then analyze the security of authenticated encryption schemes designed by "generic composition," meaning making black-box use of a given symmetric encryption scheme and a given MAC. Three composition methods are considered, namely *Encrypt-and-MAC*, *MAC-then-encrypt*, and *Encrypt-then-MAC*. For each of these, and for each notion of security, we indicate whether or not the resulting scheme meets the notion in question assuming the given symmetric encryption scheme is secure against chosen-plaintext attack and the given MAC is unforgeable under chosen-message attack. We provide proofs for the cases where the answer is "yes" and counter-examples for the cases where the answer is "no."

Keywords: Symmetric encryption, message authentication, authenticated encryption, concrete security.

*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-Mail: mihir@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/mihir. Supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

†Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-mail: meaw@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/cnamprem. Supported in part by grants of first author.

Contents

1 Introduction **3**
1.1 Relations among notions 3
1.2 Analysis of generic composition 4
1.3 Related work 6

2 Definitions **7**
2.1 Syntax of (symmetric) encryption schemes 7
2.2 Privacy of symmetric encryption schemes 8
2.3 Integrity of symmetric encryption schemes 10
2.4 Message authentication schemes 11
2.5 Notation for adversary execution 12

3 Relations among notions of symmetric encryption **13**

4 Security of the Composite Schemes **17**
4.1 Encrypt-and-MAC 18
4.2 MAC-then-Encrypt 21
4.3 Encrypt-then-MAC 24

References **28**

1 Introduction

We use the term *authenticated encryption scheme* to refer to a shared-key based transform whose goal is to provide *both* privacy *and* authenticity of the encapsulated data. In such a scheme the *encryption* process applied by the sender takes the key and a plaintext to return a ciphertext, while the *decryption* process applied by the receiver takes the same key and a ciphertext to return either a plaintext or a special symbol indicating that it considers the ciphertext invalid or unauthentic.

The design of such schemes has attracted a lot of attention historically. The early schemes were typically based on adding "redundancy" to the message before CBC encrypting, and many of these schemes were broken. Today authenticated encryption schemes continue to be the target of design and standardization efforts. A popular modern design paradigm is to combine MACs with standard block cipher modes of operation.

The goal of symmetric encryption is usually viewed as privacy, but an authenticated encryption scheme is simply a symmetric encryption scheme meeting additional authenticity goals. The first part of this paper formalizes several different possible notions of authenticity for symmetric encryption schemes, and integrates them into the existing mosaic of notions by relating them to the main known notions of privacy for symmetric encryption, via implications and separations in the style of [3]. The second part of this paper is motivated by emerging standards such as [17] which design authenticated encryption schemes by what we call "generic composition" of encryption and MAC schemes. We analyze, with regard to meeting the previous notions, several generic composition methods. Let us now look at these items in more detail.

1.1 Relations among notions

Privacy goals for symmetric encryption schemes include indistinguishability and non-malleability, each of which can be considered under either chosen-plaintext or (adaptive) chosen-ciphertext attack, leading to four notions of security we abbreviate IND-CPA, IND-CCA, NM-CPA, NM-CCA. (The original definitions were in the asymmetric setting [12, 10, 20] but can be "lifted" to the symmetric setting using the encryption oracle based template of [2]). The relations among these notions are well-understood [3, 11]. (These papers state results for the asymmetric setting, but as noted in [3] it is an easy exercise to transfer them to the symmetric setting.)

We consider two notions of integrity (we use the terms authenticity and integrity interchangeably) for symmetric encryption schemes. INT-PTXT (integrity of plaintexts) requires that it be computationally infeasible to produce a ciphertext decrypting to a message which the sender had never encrypted, while INT-CTXT (integrity of ciphertexts) requires that it be computationally infeasible to produce a ciphertext not previously produced by the sender, regardless of whether or not the underlying plaintext is "new." (In both cases, the adversary is allowed a chosen-message attack.) The first of these notions is the more natural security requirement while the interest of the second, stronger notion is perhaps more in the implications we discuss below.

These notions of authenticity are by themselves quite disjoint from the notions of privacy; for example, sending the message in the clear with an accompanying (strong) MAC achieves INT-CTXT but no kind of privacy. To make for useful comparisons, we consider each notion of authenticity coupled with IND-CPA, the weakest notion of privacy; namely the notions on which we focus for comparison purposes are INT-PTXT $\wedge$ IND-CPA and INT-CTXT $\wedge$ IND-CPA. (Read "$\wedge$" as "and".)

Figure 1 shows the graph of relations between these notions and the above-mentioned older ones in the style of [3]. An "implication" $\mathbf{A} \rightarrow \mathbf{B}$ means that every symmetric encryption scheme meeting notion $\mathbf{A}$ also meets notion $\mathbf{B}$. A "separation" $\mathbf{A} \not\rightarrow \mathbf{B}$ means that there exists a symmetric encryption scheme meeting notion $\mathbf{A}$ but not notion $\mathbf{B}$. (This under the minimal assumption that

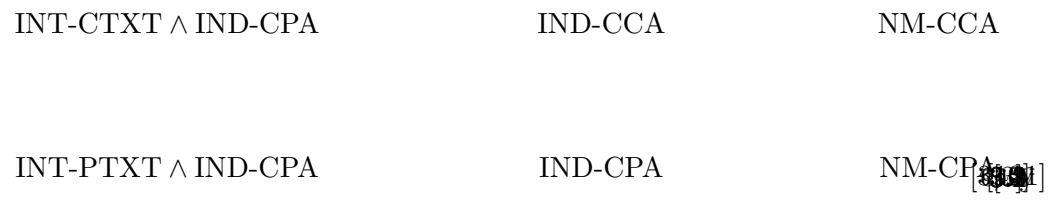

Figure 1: **Relations among notions of symmetric encryption:** An arrow denotes an implication while a barred arrow denotes a separation. The full arrows are relations proved in this paper, annotated with the number of the corresponding Proposition or Theorem, while dotted arrows are reminders of existing relations, annotated with citations to the papers establishing them.

---

some scheme meeting notion **A** exists since otherwise the question is moot.) Only a minimal set of relations is explicitly indicated; the relation between any two notions can be derived from the shown ones. (For example, IND-CCA does not imply INT-CTXT ∧ IND-CPA because otherwise, by following arrows, we would get IND-CCA → INT-PTXT ∧ IND-CPA contradicting a stated separation.) The dotted lines are reminders of existing relations while the numbers annotating the dark lines are pointers to Propositions or Theorems in this paper.

A few points may be worth highlighting. Integrity of ciphertexts —even when coupled only with the weak privacy requirement IND-CPA— emerges as the most powerful notion. Not only does it imply security against chosen-ciphertext attack, but it is strictly stronger than this notion. Non-malleability —whether under chosen-plaintext or chosen-ciphertext attack— does not imply any type of integrity. The intuitive reason is that non-malleability only prevents the generation of ciphertexts whose plaintexts are meaningfully related to those of some challenge ciphertexts, while integrity requires it to be hard to generate ciphertexts of new plaintexts even if these are unrelated to plaintexts underlying any existing ciphertexts. Finally, INT-PTXT ∧ IND-CPA does not imply INT-CTXT ∧ IND-CPA.

1.2 Analysis of generic composition

There are many possible ways to design authenticated encryption schemes. We focus in this paper on "generic composition:" simply combine a standard symmetric encryption scheme with a MAC in some way. There are a few possible ways to do it, and our goal is to analyze and compare their security. (The motivation, as we will argue, is that these "obvious" methods, as often the case in practice, remain the most pragmatic from the point of view of performance and security architecture design.)

Generic composition. Assume we are given a symmetric encryption scheme $\mathcal{SE}$ specified by an encryption algorithm $\mathcal{E}$ and a decryption algorithm $\mathcal{D}$. (Typically this will be a block cipher mode of operation.) Also assume we are given a message authentication scheme $\mathcal{MA}$ specified by a tagging algorithm $\mathcal{T}$ and a tag verifying algorithm $\mathcal{V}$ and meeting some appropriate notion of unforgeability under chosen-message attack. (Possibilities include the CBC-MAC, HMAC [1], or UMAC [7]). We want to "compose" (meaning, appropriately combine) these to create an authenticated encryption scheme meeting either INT-CTXT ∧ IND-CPA or INT-PTXT ∧ IND-CPA. Below are the composition methods we consider. We call them "generic" because the algorithms of the authenticated encryption scheme appeal to the given ones as black-boxes only. (After we present the results we will explain why this is important.) In each case K_e is a key for encryption and

Composition Method	Privacy			Integrity	
	IND-CPA	IND-CCA	NM-CPA	INT-PTXT	INT-CTXT
Encrypt-and-MAC	insecure	insecure	insecure	secure	insecure
MAC-then-encrypt	secure	insecure	insecure	secure	insecure
Encrypt-then-MAC	secure	insecure	insecure	secure	insecure

Figure 2: Summary of security results for the composed authenticated encryption schemes under the assumption that the given encryption scheme is IND-CPA and the given MAC is weakly unforgeable.

Composition Method	Privacy			Integrity	
	IND-CPA	IND-CCA	NM-CPA	INT-PTXT	INT-CTXT
Encrypt-and-MAC	insecure	insecure	insecure	secure	insecure
MAC-then-encrypt	secure	insecure	insecure	secure	insecure
Encrypt-then-MAC	secure	secure	secure	secure	secure

Figure 3: Summary of security results for the composed authenticated encryption schemes under the assumption that the given encryption scheme is IND-CPA and the given MAC is strongly unforgeable.

K_m is a key for message authentication— We consider the following ways of "composing" them in order to create an authenticated encryption scheme meeting either INT-CTXT $\wedge$ IND-CPA or INT-PTXT $\wedge$ IND-CPA.

— *Encrypt-and-MAC:* $\overline{\mathcal{E}}_{K_e,K_m}(M) = \mathcal{E}_{K_e}(M) \| \mathcal{T}_{K_m}(M)$.[1] Namely, encrypt the plaintext and append a MAC of the plaintext. "Decrypt+verify" is performed by first decrypting to get the plaintext and then verifying the tag.

— *MAC-then-encrypt:* $\overline{\mathcal{E}}_{K_e,K_m}(M) = \mathcal{E}_{K_e}(M \| \mathcal{T}_{K_m}(M))$. Namely, append a MAC to the plaintext and then encrypt them together. "Decrypt+verify" is performed by first decrypting to get the plaintext and candidate tag, and then verifying the tag.

— *Encrypt-then-MAC:* $\overline{\mathcal{E}}_{K_e,K_m}(M) = C \| \mathcal{T}_{K_m}(C)$ where $C = \mathcal{E}_{K_e}(M)$. Namely, encrypt the plaintext to get a ciphertext C and append a MAC of C. "Decrypt+verify" is performed by first verifying the tag and then decrypting C. This is the method of Internet RFC [17].

Here $\overline{\mathcal{E}}$ is the encryption algorithm of the authenticated encryption scheme while the "decrypt+verify" process specifies a decryption algorithm $\overline{\mathcal{D}}$. The latter will either return a plaintext or a special symbol indicating that it considers the ciphertext unauthentic.

Security results. Figure 2 and Figure 3 summarize the security results for the three composite authenticated encryption schemes. (We omit NM-CCA since it is equivalent to IND-CCA). Figure 2 shows the results assuming that the base MAC is weakly unforgeable while Figure 3 shows the results assuming that the MAC is strongly unforgeable. Weak unforgeability is the standard notion [4]— it should be computationally infeasible for the adversary to find a message-tag pair in which the message is "new," even after a chosen-message attack. Strong unforgeability requires that it be computationally infeasible for the adversary to find a new message-tag pair even after a chosen-

[1] Here (and everywhere in this paper) "$\|$" denotes an operation that combines several strings into one in such a way that the constituent strings are uniquely recoverable from the final one. (If lengths of all strings are fixed and known, concatenation will serve the purpose.)

message attack. (The message does not have to be new as long as the output tag was not previously attached to this message by the legitimate parties.) We note that any pseudorandom function is a strongly unforgeable MAC, and most practical MACs seem to be strongly unforgeable. Therefore, analyzing the composition methods under this notion is a realistic and useful approach. Entries in the above tables have the following meaning:

— *Secure:* The composite encryption scheme in question is proven to meet the security requirement in question, assuming only that the component encryption scheme meets IND-CPA and the message authentication scheme is unforgeable under chosen-message attack.

— *Insecure:* There exists *some* IND-CPA secure symmetric encryption and some message authentication scheme unforgeable under chosen-message attack such that the composite scheme based on them does not meet the security requirement in question.

As we can see from Figure 3, the *encrypt-then-MAC* method of [17] is secure from all points of view, making it a good choice for a standard.

The use of a generic composition method secure in the sense above is advantageous from the point of view both of performance and of security architecture. The performance benefit arises from the presence of fast MACs such as HMAC [1] and UMAC [7, 8] . The architectural benefits arise from the stringent notion of security being used. To be secure, the composition must be secure for *all* possible secure instantiations of its constituent primitives. (If it is secure for some instantiations but not others, we declare it insecure.) An application can thus choose a symmetric encryption scheme and a message authentication scheme independently (these are usually already supported by existing security analyses) and then appeal to some fixed and standard composition technique to combine them. No tailored security analysis of the composed scheme is required.

In Section 4 we state formal theorems to support the above claims, providing quantitative bounds for the positive results, and counter-examples with attacks for the negative result.

Quantitative results and comparisons. Above we have discussed our results at a qualitative level. Each result also has a quantitative counterpart; these are what our theorems actually state and prove. These "concrete security" analyses enable a designer to estimate the security of the authenticated encryption scheme in terms of that of its components. All the reductions in this paper are tight, meaning there is little to no loss of security.

1.3 Related work

The notions IND-CCA, NM-CCA were denoted IND-CCA2 and NM-CCA2, respectively, in [3]. The chosen-ciphertext attacks here are the adaptive kind [20]. Consideration of non-adaptive chosen-ciphertext attacks [18] leads to two more notions, denoted IND-CCA1 and NM-CCA1 by [3], who worked out the relations between six notions of privacy, these two and the four we consider here. (Their results hold for both the asymmetric and the symmetric settings, as mentioned before.) Three additional notions of privacy are considered and related to these six by [15]. In this paper, we have for simplicity avoided consideration of all the possible notions of privacy, focusing instead on what we consider the (four) main ones and their relations to the notions of authenticity. Relations of the remaining notions of privacy to the notions of authenticity considered here can be easily worked out.

Authenticity of an encryption scheme has been understood as a goal by designers for many years. The first formalization of which we are aware is that of [5]. (Early versions of their work date to 1998.) The notion they formalized was INT-CTXT. The formalization of INT-PTXT we use here seems to be new. In independent and concurrent work (both papers were submitted to FSE00) Katz and Yung [16] formalize INT-CTXT plus two other notions of authenticity not

considered here. They also observe the implication INT-CTXT $\wedge$ IND-CPA $\rightarrow$ IND-CCA.

Generic composition is one of many approaches to the design of authenticated encryption schemes. Two more general approaches are "encryption with redundancy" —append redundancy to the message before encrypting, the latter typically with some block cipher mode of operation— and "encode then encipher" [5] —add randomness and redundancy and then encipher rather than encrypt. As indicated above, attacks have been found on many encrypt with redundancy schemes. Encode then encipher, however, can be proven to work [5] —meaning yields schemes achieving INT-CTXT $\wedge$ IND-CPA— but requires a variable-input length pseudorandom permutation, which can be relatively expensive to construct. In addition, there are many specific schemes. One such scheme is the RPC mode of [16] but it is computation and space inefficient compared to the generic composition methods. (Processing an n-block plaintext requires $(1+c)n$ block cipher computations and results in a ciphertext of this many blocks, where $c \geq 0.3$.) Another scheme is the elegant IACBC mode of Jutla [14] which uses $n + O(\log n)$ block cipher operations to process an n-block plaintext. Implementation and testing would be required to compare its speed with that of generic composition methods that use fast MACs (cf. [1, 7, 8]).

Authenticated encryption is not the only approach to achieving security against chosen-ciphertext attacks. Direct approaches yielding more compact schemes have been provided by Desai [9].

2 Definitions

We present definitions for symmetric encryption following [2], first specifying the *syntax* —meaning what kinds of algorithms make up the scheme— and then specifying formal security measures. Associated with each scheme, each notion of security and each adversary is an advantage function that measures the success probability of this adversary as a function of the security parameter. We define asymptotic notions of security result by asking this function to be negligible for adversaries of time complexity polynomial in the security parameter. Concrete security assessments are made by associating to the scheme another advantage function that for each value of the security parameter and given resources for an adversary returns the maximum, over all adversaries limited to the given resources, of the success probability.

The concrete security assessments are important in practical applications— block cipher based schemes have no associated asymptotics. Hence, we provide concrete security assessments for all positive results (implications or proofs that composition methods meet some notion of security). For simplicity, however, negative results (separations or counter-examples) are phrased in the asymptotic style. (Concrete security statements are, however, easily derived from the proofs.)

2.1 Syntax of (symmetric) encryption schemes

A *(symmetric) encryption scheme* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three algorithms. The randomized *key generation* algorithm $\mathcal{K}$ takes input a security parameter $k \in \mathsf{N}$ and returns a key K; we write $K \xleftarrow{R} \mathcal{K}(k)$. The *encryption* algorithm $\mathcal{E}$ could be randomized or stateful. It takes the key K and a *plaintext* M to return a *ciphertext* C; we write $C \xleftarrow{R} \mathcal{E}_K(M)$. (If randomized, it flips coins anew on each invocation. If stateful, it uses and then updates a state that is maintained across invocations.) The *decryption* algorithm $\mathcal{D}$ is deterministic and stateless. It takes the key K and a string C to return either the corresponding plaintext M or the symbol $\perp$; we write $x \leftarrow \mathcal{D}_K(C)$ where $x \in \{0,1\}^* \cup \{\perp\}$. We require that $\mathcal{D}_K(\mathcal{E}_K(M)) = M$ for all $M \in \{0,1\}^*$. An authenticated encryption scheme is syntactically identical to an encryption scheme as defined above; we will use the term only to emphasize cases where we are targeting authenticity goals.

2.2 Privacy of symmetric encryption schemes

We measure indistinguishability via the "left-or-right" model of [2]. Define the *left-or-right encryption oracle* $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$, where $b \in \{0,1\}$, to take input (x_0, x_1) and do the following: if $b = 0$ it computes $C \leftarrow \mathcal{E}_K(x_0)$ and returns C; else it computes $C \leftarrow \mathcal{E}_K(x_1)$ and returns C. (It is understood that the oracle picks any coins that $\mathcal{E}$ might need if $\mathcal{E}$ is randomized, or updates its state appropriately if $\mathcal{E}$ is stateful.) The adversary makes oracle queries of the form (x_0, x_1) consisting of two equal length messages and must guess the bit b. We consider an encryption scheme to be "good" if a "reasonable" adversary cannot obtain "significant" advantage in distinguishing the cases $b = 0$ and $b = 1$ given access to the oracle. To model chosen-ciphertext attacks we allow the adversary to also have access to a decryption oracle. Note that if the adversary queries the decryption oracle at a ciphertext output by the left-or-right oracle, then it can obviously easily win the game. Therefore, we disallow it from doing so. Any other query is permissible.

Definition 2.1 (Indistinguishability of a Symmetric Encryption Scheme [2]) Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. Let $b \in \{0,1\}$ and $k \in \mathsf{N}$. Let A_{cpa} be an adversary that has access to one oracle and let A_{cca} be an adversary that has access to two oracles. Now, we consider the following experiments:

Experiment $\mathbf{Exp}^{\text{ind-cpa-}b}_{\mathcal{SE},A_{\text{cpa}}}(k)$	Experiment $\mathbf{Exp}^{\text{ind-cca-}b}_{\mathcal{SE},A_{\text{cca}}}(k)$
$K \xleftarrow{R} \mathcal{K}(k)$	$K \xleftarrow{R} \mathcal{K}(k)$
$x \leftarrow A_{\text{cpa}}^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))}(k)$	$x \leftarrow A_{\text{cca}}^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b)),\mathcal{D}_K(\cdot)}(k)$
Return x	Return x

Above it is mandated that A_{cca} never queries $\mathcal{D}_K(\cdot)$ on a ciphertext C output by the $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ oracle, and that the two messages queried of $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ always have equal length. We define the *advantages* of the adversaries via

$$\begin{aligned}
\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},A_{\text{cpa}}}(k) &= \Pr\left[\mathbf{Exp}^{\text{ind-cpa-1}}_{\mathcal{SE},A_{\text{cpa}}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-cpa-0}}_{\mathcal{SE},A_{\text{cpa}}}(k) = 1\right] \\
\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE},A_{\text{cca}}}(k) &= \Pr\left[\mathbf{Exp}^{\text{ind-cca-1}}_{\mathcal{SE},A_{\text{cca}}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-cca-0}}_{\mathcal{SE},A_{\text{cca}}}(k) = 1\right] .
\end{aligned}$$

We define the *advantage functions of the scheme* as follows. For any integers $t, q_e, q_d, \mu_e, \mu_d$,

$$\begin{aligned}
\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &= \max_{A_{\text{cpa}}}\{\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},A_{\text{cpa}}}(k)\} \\
\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE}}(k, t, q_e, q_d, \mu_e, \mu_d) &= \max_{A_{\text{cca}}}\{\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE},A_{\text{cca}}}(k)\}
\end{aligned}$$

where the maximum is over all $A_{\text{cpa}}, A_{\text{cca}}$ with time-complexity t, each making to the $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ oracle at most q_e queries the sum of whose lengths is at most μ_e bits, and, in the case of A_{cca}, also making to the $\mathcal{D}_K(\cdot)$ oracle at most q_d queries the sum of those lengths is at most μ_d bits. The scheme $\mathcal{SE}$ is said to be *IND-CPA secure* —resp. *IND-CCA secure*— if the function $\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},A}(\cdot)$ —resp. $\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE},A}(\cdot)$— is negligible for any adversary A whose time-complexity is polynomial in k. ▌

We discuss some important conventions. The *time-complexity* mentioned above is the worst case total execution time of the experiment, plus the size of the code of the adversary, in some fixed RAM model of computation. We stress that the the total execution time of the experiment is more than the running time of the adversary. It includes the time of *all* operations in the experiment, including the time for key generation and the computation of answers to oracle queries. Thus, when

the time complexity is polynomially bounded, so are all the other parameters. This convention for measuring time complexity and other resources of an adversary is used for all definitions in this paper.

Another convention, also used throughout this paper, is that the length of a query M_0, M_1 to a left-or-right encryption oracle is defined as $|M_0|$. (This equals $|M_1|$ since the messages must have the same length.) In other words, it is the length of one of the messages. This convention is used in measuring the parameter μ_e.

The advantage function is the maximum probability that the security of the scheme $\mathcal{SE}$ can be compromised by an adversary using the indicated resources, and is used for concrete security analyses.

We will not use definitions of non-malleability as per [10, 3] but instead use the equivalent indistinguishability under parallel chosen-ciphertext attack characterization of [6]. This facilitates our proofs and analyses and also facilitates concrete security measurements. The notation $\vec{\mathcal{D}}_K(\cdot)$ denotes the algorithm which takes input a vector $\vec{c} = (c_1, \ldots, c_n)$ of ciphertexts and returns the corresponding vector $\vec{p} = (\mathcal{D}_K(c_1), \ldots, \mathcal{D}_K(c_n))$ of plaintexts.

Definition 2.2 (Non-Malleability of a Symmetric Encryption Scheme [6]) Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. Let $b \in \{0,1\}$ and $k \in \mathsf{N}$. Let $A_{\text{cpa}} = (A_{\text{cpa}_1}, A_{\text{cpa}_2})$ be an adversary that has access to one oracle and let $A_{\text{cca}} = (A_{\text{cca}_1}, A_{\text{cca}_2})$ be an adversary that has access to two oracles. Now, we consider the following experiments:

Experiment $\mathbf{Exp}^{\text{nm-cpa-}b}_{\mathcal{SE},A_{\text{cpa}}}(k)$	Experiment $\mathbf{Exp}^{\text{nm-cca-}b}_{\mathcal{SE},A_{\text{cca}}}(k)$
$K \xleftarrow{R} \mathcal{K}(k)$	$K \xleftarrow{R} \mathcal{K}(k)$
$(\vec{c}, s) \leftarrow A_{\text{cpa}_1}^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))}(k)$	$(\vec{c}, s) \leftarrow A_{\text{cca}_1}^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b)),\mathcal{D}_K(\cdot)}(k)$
$\vec{p} \leftarrow \vec{\mathcal{D}}_K(\vec{c})$	$\vec{p} \leftarrow \vec{\mathcal{D}}_K(\vec{c})$
$x \leftarrow A_{\text{cpa}_2}(\vec{p}, \vec{c}, s)$	$x \leftarrow A_{\text{cca}_2}(\vec{p}, \vec{c}, s)$
`Return` x	`Return` x

Above it is mandated that the vector $\vec{c}$ output by A_{cpa_1} or A_{cca_1} does not contain any of the ciphertexts output by the $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ oracle, that the pairs of messages queried of $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ are always of equal length, and that A_{cca} does not query $\mathcal{D}_K(\cdot)$ on an output of $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$. We define the *advantages* of the adversaries via

$$\begin{aligned}
\mathbf{Adv}^{\text{nm-cpa}}_{\mathcal{SE},A_{\text{cpa}}}(k) &= \Pr\left[\mathbf{Exp}^{\text{nm-cpa-1}}_{\mathcal{SE},A_{\text{cpa}}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{nm-cpa-0}}_{\mathcal{SE},A_{\text{cpa}}}(k) = 1\right] \\
\mathbf{Adv}^{\text{nm-cca}}_{\mathcal{SE},A_{\text{cca}}}(k) &= \Pr\left[\mathbf{Exp}^{\text{nm-cca-1}}_{\mathcal{SE},A_{\text{cca}}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{nm-cca-0}}_{\mathcal{SE},A_{\text{cca}}}(k) = 1\right] .
\end{aligned}$$

We define the *advantage functions of the scheme* as follows. For any integers $t, q_e, q_d, \mu_e, \mu_d$,

$$\begin{aligned}
\mathbf{Adv}^{\text{nm-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &= \max_{A_{\text{cpa}}}\{\mathbf{Adv}^{\text{nm-cpa}}_{\mathcal{SE},A_{\text{cpa}}}(k)\} \\
\mathbf{Adv}^{\text{nm-cca}}_{\mathcal{SE}}(k, t, q_e, q_d, \mu_e, \mu_d) &= \max_{A_{\text{cca}}}\{\mathbf{Adv}^{\text{nm-cca}}_{\mathcal{SE},A_{\text{cca}}}(k)\}
\end{aligned}$$

where the maximum is over all $A_{\text{cpa}}, A_{\text{cca}}$ with time-complexity t, each making to the $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ oracle at most q_e queries the sum of whose lengths is at most μ_e bits, and, in the case of A_{cca}, also making to the $\mathcal{D}_K(\cdot)$ oracle at most q_d queries the sum of whose lengths is at most μ_d bits. The scheme $\mathcal{SE}$ is said to be *NM-CPA secure* —resp. *NM-CCA secure*— if the function $\mathbf{Adv}^{\text{nm-cpa}}_{\mathcal{SE},A}(\cdot)$ —resp. $\mathbf{Adv}^{\text{nm-cca}}_{\mathcal{SE},A}(\cdot)$— is negligible for any adversary A whose time complexity is polynomial in k. ▌

2.3 Integrity of symmetric encryption schemes

Now we specify security definitions for integrity (authenticity) of a symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. It is convenient to define an algorithm $\mathcal{D}^*_K(\cdot)$ as follows:

Algorithm $\mathcal{D}^*_K(C)$
  If $\mathcal{D}_K(C) \neq \perp$, then return 1.
    Else return 0.

We call this the *verification algorithm* or *verification oracle*. The model is similar to that used for message authentication except that the messages are no longer in the clear, but specified implicitly via ciphertexts. The adversary is allowed to mount a chosen-message attack on the scheme, modeled by giving it access to an encryption oracle $\mathcal{E}_K(\cdot)$. It also has oracle access to the verification oracle. It is successful if it makes the verification oracle accept a ciphertext that was not "legitimately produced." There are two possible interpretations of the phrase in quotes. One is to consider the ciphertext illegitimate —meaning consider the adversary successful— if the corresponding plaintext was never queried of the encryption oracle. A scheme in which it is computationally infeasible for the adversary to achieve this type of success is said to preserve the *integrity of plaintexts*. The other possibility is to consider the ciphertext illegitimate —meaning consider the adversary successful— if the ciphertext was never returned by the encryption oracle, even if the corresponding plaintext was queried of the encryption oracle. A scheme in which it is computationally infeasible for the adversary to achieve this type of success is said to preserve the *integrity of ciphertexts*.

Definition 2.3 (Integrity of an Authenticated Encryption Scheme) Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. Let $k \in \mathsf{N}$, and let A_{ptxt} and A_{ctxt} be adversaries each of which has access to two oracles. Consider the following experiments:

Experiment $\mathbf{Exp}^{\text{int-ptxt}}_{\mathcal{SE},A_{\text{ptxt}}}(k)$
  $K \xleftarrow{R} \mathcal{K}(k)$
  If $A^{\mathcal{E}_K(\cdot),\mathcal{D}^*_K(\cdot)}_{\text{ptxt}}(k)$ makes a query C to
  the oracle $\mathcal{D}^*_K(\cdot)$ such that
    – $\mathcal{D}^*_K(C)$ returns 1, and
    – $M \stackrel{\text{def}}{=} \mathcal{D}_K(C)$ was never a query to $\mathcal{E}_K(\cdot)$
  then return 1 else return 0.

Experiment $\mathbf{Exp}^{\text{int-ctxt}}_{\mathcal{SE},A_{\text{ctxt}}}(k)$
  $K \xleftarrow{R} \mathcal{K}(k)$
  If $A^{\mathcal{E}_K(\cdot),\mathcal{D}^*_K(\cdot)}_{\text{ctxt}}(k)$ makes a query C to
  the oracle $\mathcal{D}^*_K(\cdot)$ such that
    – $\mathcal{D}^*_K(C)$ returns 1, and
    – C was never a response of $\mathcal{E}_K(\cdot)$
  then return 1 else return 0.

We define the *advantages* of the adversaries via

$$\begin{aligned}
\mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{SE},A_{\text{ptxt}}}(k) &= \Pr\left[\,\mathbf{Exp}^{\text{int-ptxt}}_{\mathcal{SE},A_{\text{ptxt}}}(k) = 1\,\right] \\
\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},A_{\text{ctxt}}}(k) &= \Pr\left[\,\mathbf{Exp}^{\text{int-ctxt}}_{\mathcal{SE},A_{\text{ctxt}}}(k) = 1\,\right]
\end{aligned}$$

We define the *advantage functions of the scheme* as follows. For any integers $t, q_e, q_d, \mu_e, \mu_d$,

$$\begin{aligned}
\mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{SE}}(k, t, q_e, q_d, \mu_e, \mu_d) &= \max_{A_{\text{ptxt}}}\{\mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{SE},A_{\text{ptxt}}}(k)\} \\
\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE}}(k, t, q_e, q_d, \mu_e, \mu_d) &= \max_{A_{\text{ctxt}}}\{\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},A_{\text{ctxt}}}(k)\}
\end{aligned}$$

where the maximum is over all A_{ptxt}, A_{ctxt} with time-complexity t, each making to the oracle $\mathcal{E}_K(\cdot)$ at most q_e queries the sum of whose lengths is at most μ_e bits, and each making to the $\mathcal{D}^*_K(\cdot)$ oracle t most q_d queries the sum of whose lengths is at most μ bits. The scheme $\mathcal{SE}$ is said to be *INT-PTXT secure* —resp. *INT-CTXT secure*— if the function $\mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{SE},A}(\cdot)$ — resp. $\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},A}(\cdot)$— is negligible for any adversary A whose time-complexity is polynomial in k. ▮

2.4 Message authentication schemes

A *message authentication scheme* $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ consists of three algorithms. The randomized *key generation* algorithm $\mathcal{K}$ takes input a security parameter $k \in \mathsf{N}$ and returns a key K; we write $K \xleftarrow{R} \mathcal{K}(k)$. The *tagging* algorithm $\mathcal{T}$ could be either randomized or stateful. It takes the key K and a message M to return a *tag* σ; we write $\sigma \xleftarrow{R} \mathcal{T}_K(M)$. The *verification* algorithm $\mathcal{V}$ is deterministic. It takes the key K, a message M, and a candidate tag σ for M to return a bit v; we write $v \leftarrow \mathcal{V}_K(M, \sigma)$. We require that $\mathcal{V}_K(M, \mathcal{T}_K(M)) = 1$ for all $M \in \{0,1\}^*$. The scheme is said to be deterministic if the tagging algorithm is deterministic and verification is done via tag re-computation. We sometimes call a message authentication scheme a MAC, and also sometimes call the tag σ a MAC.

Security for message authentication considers an adversary F who is allowed a chosen-message attack, modeled by allowing it access to an oracle for $\mathcal{T}_K(\cdot)$. F is "successful" if it can make the verifying oracle $\mathcal{V}_K(\cdot,\cdot)$ accept a pair (M, σ) that was not "legitimately produced." There are two possible conventions with regard to what "legitimately produced" can mean, leading to two measures of advantage. The "standard" measure is that the message M is "new," meaning F never made query M of its tagging oracle. We call this type of forgery a *weak forgery*. (This is the measure of [4] which in turn is an adaptation to the symmetric case of the notion of security for digital signatures of [13].) A more stringent measure considers the adversary successful even if the message is not new, as long as the tag is new. This type of *strong forgery* means that the adversary wins as long as σ was never returned by the tagging oracle in response to query M. In the following definition, we use the acronyms WUF-CMA and SUF-CMA respectively for weak and strong unforgeability against chosen-message attacks.

Definition 2.4 (Message Authentication Scheme Security) Let $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ be a message authentication scheme. Let $k \in \mathsf{N}$, and let F_{w} and F_{s} be adversaries that have access to two oracles. Consider the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{wuf-cma}}_{\mathcal{MA},F_{\mathrm{w}}}(k)$
$K \xleftarrow{R} \mathcal{K}(k)$
If $F_{\mathrm{w}}^{\mathcal{T}_K(\cdot),\mathcal{V}_K(\cdot,\cdot)}(k)$ makes a query (M,σ) to the oracle $\mathcal{V}_K(\cdot,\cdot)$ such that
– $\mathcal{V}_K(M,\sigma)$ returns 1, and
– M was never queried to the oracle $\mathcal{T}_K(\cdot)$,
`then` return 1 `else` return 0.

Experiment $\mathbf{Exp}^{\mathrm{suf-cma}}_{\mathcal{MA},F_{\mathrm{s}}}(k)$
$K \xleftarrow{R} \mathcal{K}(k)$
If $F_{\mathrm{s}}^{\mathcal{T}_K(\cdot),\mathcal{V}_K(\cdot,\cdot)}(k)$ makes a query (M,σ) to the oracle $\mathcal{V}_K(\cdot,\cdot)$ such that
– $\mathcal{V}_K(M,\sigma)$ returns 1, and
– σ was never returned by the oracle $\mathcal{T}_K(\cdot)$ in response to query M,
`then` return 1 `else` return 0.

We define the *advantages* of the forgers via

$$\begin{aligned}\mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA},F_{\mathrm{w}}}(k) &= \Pr\left[\,\mathbf{Exp}^{\mathrm{wuf-cma}}_{\mathcal{MA},F_{\mathrm{w}}}(k) = 1\,\right]\\ \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA},F_{\mathrm{s}}}(k) &= \Pr\left[\,\mathbf{Exp}^{\mathrm{suf-cma}}_{\mathcal{MA},F_{\mathrm{s}}}(k) = 1\,\right]\end{aligned}$$

We define the *advantage functions of the scheme* as follows. For any integers $t, q_t, q_v, \mu_t, \mu_v$,

$$\begin{aligned}\mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA}}(k,t,q_t,q_v,\mu_t,\mu_v) &= \max_{F_{\mathrm{w}}}\{\mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA},F_{\mathrm{w}}}(k)\}\\ \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(k,t,q_t,q_v,\mu_t,\mu_v) &= \max_{F_{\mathrm{s}}}\{\mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA},F_{\mathrm{s}}}(k)\}\end{aligned}$$

where the maximum is over all F_{w}, F_{s} with time complexity t, making at most q_t oracle queries to $\mathcal{T}_K(\cdot)$ the sum of whose lengths is at most μ_t bits, and making at most q_v oracle queries to $\mathcal{V}_K(\cdot,\cdot)$

the sum of whose lengths is at most μ_v bits. The scheme $\mathcal{MA}$ is said to be *WUF-CMA secure* —resp. *SUF-CMA secure*— if the function $\mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA},F}(\cdot)$ —resp. $\mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA},F}(\cdot)$— is negligible for any forger F whose time complexity is polynomial in k. ▮

It is easy to show that any pseudorandom function (PRF) is a SUF-CMA-secure (determinstic) message authentication scheme. Assuming the underlying block cipher is a PRF, the CBC-MAC based on it is known to be a PRF [4, 19] hence is a SUF-CMA-secure MAC. Many practical MACs such as HMAC [1] also seem to be SUF-CMA-secure. For the sequel it is useful to note that any scheme which is SUF-CMA-secure is also WUF-CMA-secure.

Theorem 2.5 (SUF-CMA → WUF-CMA) Let $\mathcal{MA}$ be a message authentication scheme. If $\mathcal{MA}$ is SUF-CMA secure, then it is WUF-CMA secure as well. Concretely,

$$\mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA}}(k, t, q_t, q_v, \mu_t, \mu_v) \leq \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(k, t, q_t, q_v, \mu_t, \mu_v) \; . \; ▮$$

Proof of Theorem 2.5: This is true because a valid weak forgery is also a valid strong forgery. In particular, a tag corresponding to a new message is clearly a new tag for that message. Here are the details.

We associate with any forger F_{w} mounting an attack against the scheme $\mathcal{MA}$ under the WUF-CMA notion a forger F_{s} mounting an attack against the scheme under the SUF-CMA notion such that

$$\mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA},F_{\text{w}}}(k) \leq \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA},F_{\text{s}}}(k)$$

and F_{s} uses the same amount of resources as F_{s} does. Then, Theorem 2.5 follows.

In this case, we simply set the adversary F_{s} to be exactly the same as F_{w}. To see why this works, let (M, σ) be the winning forgery made by F_{w}, namely a query to the verification oracle $\mathcal{V}_K(\cdot, \cdot)$ such that this oracle returns 1 and M was never queried to the tagging oracle $\mathcal{T}_K(\cdot)$. Clearly, since the tagging oracle never receives a query M, it cannot have returned σ as a response to a query M. So (M, σ) is a valid strong forgery, and thus, F_{s} achieves its goal. ▮

2.5 Notation for adversary execution

In reductions we will often have one adversary A' executing another adversary A. Adversary A' will maintain the execution state of A. Whenever A makes an oracle query, A' will stop A, itself return a reply to this oracle query, and then continue running A. We will write code for A' which will contain things of the form:

```
For i = 1, ..., q_e do
    When A makes oracle query x_i
        [ Some code computing a value y_i ]
        A ⇐ y_i
    EndWhile
A ⇒ b
```

The notation $A \Leftarrow y_i$ means that A is being provided the value y_i in response to its oracle query x_i. It is assumed here that A makes a total of q_e queries. The notation $A \Rightarrow b$ means that A is returning a value b.

3 Relations among notions of symmetric encryption

In this section, we state the formal versions of the results summarized in Figure 1 and provide proofs. We begin with the implications and then move to the separations. The first implication, below, is a triviality.

Theorem 3.1 (INT-CTXT → INT-PTXT) Let $\mathcal{SE}$ be an encryption scheme. If $\mathcal{SE}$ is INT-CTXT secure, then it is INT-PTXT secure as well. Concretely,

$$\mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{SE}}(k,t,q_e,q_d,\mu_e,\mu_d) \le \mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE}}(k,t,q_e,q_d,\mu_e,\mu_d)\ . \blacksquare$$

Proof of Theorem 3.1: This is true because an adversary that violates integrity of plaintexts of a scheme $\mathcal{SE} = (\mathcal{K},\mathcal{E},\mathcal{D})$ also violates integrity of ciphertexts of the same scheme. Here are the details.

We associate with any adversary A mounting an attack against integrity of plaintexts of $\mathcal{SE}$ an adversary A' mounting an attack against integrity of ciphertexts of the scheme such that

$$\mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{SE},A}(k) \le \mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},A'}(k)$$

and A' uses the same amount of resources as A does. Then, Theorem 3.1 follows.

In this case, we simply set the adversary A' to be exactly the same as A. To see why this works, let C be a winning query made by A in $\mathbf{Exp}^{\text{int-ptxt}}_{\mathcal{SE},A}(k)$, namely a query to $\mathcal{D}^*_K(\cdot)$ such that this oracle returns 1 but

$$M \stackrel{\text{def}}{=} \mathcal{D}_K(C)$$

was never queried to the oracle $\mathcal{E}_K(\cdot)$. We claim that C was never an output of the oracle $\mathcal{E}_K(\cdot)$ in $\mathbf{Exp}^{\text{int-ctxt}}_{\mathcal{SE},A}(k)$. This is true by the unique decryptablity of a symmetric encryption scheme: the only possible message that could result in an output of C from $\mathcal{E}_K(\cdot)$ is M. So A' achieves its goal with the same probability that A achieves its goal. $\blacksquare$

The next implication is more interesting.

Theorem 3.2 (INT-CTXT ∧ IND-CPA → IND-CCA) Let $\mathcal{SE}$ be an encryption scheme. If $\mathcal{SE}$ is INT-CTXT secure and IND-CPA secure, then it is IND-CCA secure. Concretely,

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE}}(k,t,q_e,q_d,\mu_e,\mu_d) \le 2\cdot\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE}}(k,t,q_e,q_d,\mu_e,\mu_d) + \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e)\ . \blacksquare$$

Proof of Theorem 3.2: Let $\mathcal{SE} = (\mathcal{K},\mathcal{E},\mathcal{D})$ be a symmetric encryption scheme. To any adversary A attacking the scheme in the IND-CCA sense we associate two adversaries, A_c which attacks $\mathcal{SE}$ in the INT-CTXT sense, and A_p which attacks $\mathcal{SE}$ in the IND-CPA sense, so that

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE},A}(k) \;\;\le\;\; 2\cdot\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},A_c}(k) + \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},A_p}(k)\ , \tag{1}$$

and furthermore, if A runs in time t using q_e encryption and q_d decryption queries totaling μ_e,μ_d bits respectively, then A_c runs in time t using q_e encryption and q_d verification queries totaling μ_e,μ_d bits respectively, and A_p runs in time t using q_e encryption queries tataling μ_e bits. Then, Theorem 3.2 follows.

The two adversaries A_c and A_p will use A to achieve their goals. Specifically, A_c whose goal is to submit a new valid ciphertext query to the oracle $\mathcal{D}^*$ will simply use A's query to the oracle $\mathcal{D}$ as

its own. Thus, if A can form a valid ciphertext query, so will A_c. Similarly, A_p whose goal is to figure out whether the left or the right message has been encrypted will directly use A's ability to do so. Lacking access to a decryption oracle, however, it will simply return $\perp$ when A asks for a decryption. This strategy works overall mainly because, regardless of whether A can form a valid ciphertext, at least one of the two adversaries will benefit.

The constructions for A_c and A_p are as follows. Refer to the end of Section 2 for the notation $A \Leftarrow \cdot$ and $A \Rightarrow \cdot$.

Adversary $A_c^{\mathcal{E}_K(\cdot),\mathcal{D}_K^*(\cdot)}(k)$
  $b' \xleftarrow{R} \{0,1\}$
  For $i = 1, \ldots, q_e + q_d$ do
    When A makes a query $M_{i,0}, M_{i,1}$
    to its left-or-right encryption oracle do
      $A \Leftarrow \mathcal{E}_K(M_{i,b'})$.
    When A makes a query C_i
    to its decryption oracle do
      $v \leftarrow \mathcal{D}_K^*(C_i)$
      If $v = 0$,
        then $A \Leftarrow \perp$,
        else stop.

Adversary $A_p^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))}(k)$
  For $i = 1, \ldots, q_e + q_d$ do
    When A makes a query $M_{i,0}, M_{i,1}$
    to its left-or-right encryption oracle do
      $A \Leftarrow \mathcal{E}_K(\mathcal{LR}(M_{i,0}, M_{i,1}, b))$
    When A makes a query C_i
    to its decryption oracle do
      $A \Leftarrow \perp$
  $A \Rightarrow b'$
  Return b'

We will now prove Equation (1). Let $\Pr[\,\cdot\,]$ denote the probability in $\mathbf{Exp}^{\text{ind-cca-}b}_{\mathcal{SE},A}(k)$ where $b \in \{0,1\}$ and let b' denote the bit output by A in this experiment. Let E denote the event that A makes at least one valid decryption oracle query, i.e. a query C such that $\mathcal{D}_K(C) \neq \perp$. Let $\Pr_p[\,\cdot\,]$ denote the probability in $\mathbf{Exp}^{\text{ind-cpa-}b}_{\mathcal{SE},A_p}(k)$ and let $\Pr_c[\,\cdot\,]$ denote the probability in $\mathbf{Exp}^{\text{int-ctxt}}_{\mathcal{SE},A_c}(k)$. We claim

$$\begin{aligned} \Pr[\,b' = b \,\wedge\, E\,] &\leq \Pr[\,E\,] \\ &= \Pr_c[\,A_c \text{ succeeds}\,] \\ &= \mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},A_c}(k) \end{aligned} \tag{2}$$

and

$$\begin{aligned} \Pr[\,b' = b \,\wedge\, \neg E\,] &\leq \Pr_p[\,b' = b\,] \\ &= \frac{1}{2}\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},A_p}(k) + \frac{1}{2}\,. \end{aligned} \tag{3}$$

We finish the proof given this and then return to the justification. We have

$$\begin{aligned} \frac{1}{2}\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE},A}(k) + \frac{1}{2} &= \Pr[\,b' = b\,] \\ &= \Pr[\,b' = b \,\wedge\, E\,] + \Pr[\,b' = b \,\wedge\, \neg E\,] \\ &\leq \mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},A_c}(k) + \frac{1}{2}\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},A_p}(k) + \frac{1}{2}\,. \end{aligned}$$

Some algebraic manipulation leads to Equation (1).

We now justify the claimed inequalities (2) and (3) by analyzing each of them in turn.

To justify the inequality (2), we observe that A_c simulates A in the exact same environment as that of the experiment $\mathbf{Exp}^{\text{ind-cca-}b}_{\mathcal{SE},A}(k)$. Therefore, if A submits a valid ciphertext as a decryption

query (i.e. the event E occurs), A_c uses this ciphertext as a query to its verification oracle, and so Equation (2) follows. (Once this ciphertext has been submitted A_c stops and of course the simulation is then no longer accurate but that doesn't matter.) Similarly, for the inequality (3), when event E does not occur, A_p simulates A in the exact same environment as that of the experiment $\mathbf{Exp}^{\text{ind-cca-}b}_{\mathcal{SE},A}(k)$. Therefore, if A is able to guess the correct bit $b' = b$, so will A_p, and Equation (3) follows. This concludes the proof for Equation (1).

Note that here we rely on the assumption that A never queries its decryption oracle on an output of its $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ oracle. Otherwise, A could query its decryption oracle with a valid ciphertext C, meaning event E would occur, yet there would be no win for A_c because the simulation would have lead it to query its own encryption oracle on the message which is the decryption of C.

To justify the claimed resource complexities of A_c and A_p, we note that each of A_c and A_p uses the same number of queries as that of A (A_c to its $\mathcal{E}_K(\cdot)$ and $\mathcal{D}^*_K(\cdot)$ oracles and A_p to its $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ oracle). For time complexity, we simply note that we measure the time for each *entire* experiment. Therefore, Equation (1) leads to Theorem 3.2. We omit details. ▌

We use the approach of [3] to show separations. Namely, to show that a security notion **A** does not imply a security notion **B**, we construct a scheme $\overline{\mathcal{SE}}$ that meets notion **A** but for which we can exhibit an attack showing that it does not meet notion **B**. Of course, the statement that $\mathbf{A} \not\rightarrow \mathbf{B}$ is vacuously and un-interestingly true if there does not exist any scheme secure under the notion **A** in the first place. So we make the minimal assumption whenever we show a separation $\mathbf{A} \not\rightarrow \mathbf{B}$ that there exists some scheme secure under the notion **A**, and obtain $\overline{\mathcal{SE}}$ by modifying this given scheme.

We note that the scheme $\overline{\mathcal{SE}}$ may be artificial. But the point we are making is that it is not possible to prove $\mathbf{A} \rightarrow \mathbf{B}$, and even an artificial example is enough for that.

Proposition 3.3 (IND-CCA $\not\rightarrow$ INT-PTXT) Given a symmetric encryption scheme $\mathcal{SE}$ which is IND-CCA secure, we can construct a symmetric encryption scheme $\overline{\mathcal{SE}}$ which is also IND-CCA secure but is *not* INT-PTXT secure. ▌

Proof of Proposition 3.3: Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the given symmetric encryption scheme. We define the scheme $\overline{\mathcal{SE}}$ such that $\overline{\mathcal{SE}}$ is IND-CCA secure but is not INT-PTXT secure. The idea is simple. A certain known string (or strings) will be viewed by $\overline{\mathcal{D}}$ as valid and decrypted to certain known messages, so that forgery is easy. But these "ciphertexts" will never be produced by the encryption algorithm so privacy will not be affected. Here are the details.

The new scheme $\overline{\mathcal{SE}} = (\mathcal{K}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ has the same key generation algorithm as the old scheme and the following modified encryption and decryption algorithms:

Algorithm $\overline{\mathcal{E}}_K(M)$	Algorithm $\overline{\mathcal{D}}_K(C)$
$C' \leftarrow \mathcal{E}_K(M)$	Parse C as $b\|C'$ where b is a bit
$C \leftarrow 0\|C'$	If $b = 0$ `then` $M \leftarrow \mathcal{D}_K(C')$; `return` M
`Return` C	`Else return` 0

We present an attack on $\overline{\mathcal{SE}}$, in the form of an adversary A who defeats the integrity of plaintexts with probability one using resources polynomial in the security parameter k. It works as follows:

Adversary $A^{\overline{\mathcal{E}}_K(\cdot),\overline{\mathcal{D}}^*_K(\cdot)}(k)$
  Submit query 10 to oracle $\overline{\mathcal{D}}^*_K(\cdot)$.

We observe that $\overline{\mathcal{D}}_K(10) = 0$, meaning 10 is a valid ciphertext, and it decrypts to a message (namely 0) that the adversary has not queried of its oracle. So

$$\mathbf{Adv}^{\text{int-ptxt}}_{\overline{\mathcal{SE}},A}(k) = 1 \ .$$

Also, A makes zero queries to $\overline{\mathcal{E}}_K(\cdot)$ and one query to $\overline{\mathcal{D}}^*_K(\cdot)$ totalling 2 bits, and is certainly poly(k)-time.

To prove that $\overline{\mathcal{SE}}$ is IND-CCA secure, it suffices to associate with any poly(k)-time adversary A attacking $\overline{\mathcal{SE}}$ in the IND-CCA sense a poly(k)-time adversary B attacking $\mathcal{SE}$ in the IND-CCA sense such that

$$\mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{SE}},A}(k) \leq \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{SE},B}(k) \ .$$

Adversary B simply simulates A and uses its oracles to answer A's oracle queries in a straightforward manner as follows:

Adversary $B^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b)),\mathcal{D}_K(\cdot)}(k)$
  `For` $i = 1, \ldots, q_e + q_d$ `do`
    When A makes a query $M_{i,0}, M_{i,1}$ to its left-or-right encryption oracle do
      $A \Leftarrow 0 \| \mathcal{E}_K(\mathcal{LR}(M_{i,0}, M_{i,1}, b))$
    When A makes a query C_i to its decryption oracle do
      Parse C as $b_i \| C'_i$ where b_i is a bit
      `If` $b = 0$ then $A \Leftarrow \mathcal{D}_K(C'_i)$
      `Else` $A \Leftarrow 0$
  `Return` whatever A returns

As the code shows, it is easy for B to break the scheme if A can. Furthermore, the resource usage of both adversaries are clearly the same. ▌

Proposition 3.4 (INT-PTXT∧IND-CPA $\not\rightarrow$ NM-CPA) Given a symmetric encryption scheme $\mathcal{SE}$ which is both INT-PTXT secure and IND-CPA secure, we can construct a symmetric encryption scheme $\overline{\mathcal{SE}}$ which is also both INT-PTXT secure and IND-CPA secure but is *not* NM-CPA secure. ▌

Proof of Proposition 3.4: Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the given symmetric encryption scheme. We define the scheme $\overline{\mathcal{SE}}$ such that $\overline{\mathcal{SE}}$ is INT-PTXT and IND-CPA secure but is not NM-CPA secure. The idea is to prepend a redundant bit to ciphertexts. This bit is ignored by $\overline{\mathcal{D}}$, resulting in the ability to create two different ciphertexts of the same message, which defeats the non-malleability. Here are the details.

The new scheme $\overline{\mathcal{SE}} = (\mathcal{K}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ has the same key generation algorithm as the old scheme and the following modified encryption and decryption algorithms:

Algorithm $\overline{\mathcal{E}}_K(M)$	Algorithm $\overline{\mathcal{D}}_K(C)$
$C \leftarrow \mathcal{E}_K(M)$	Parse C as $b \| C'$ where b is a bit
`Return` $0 \| C$	$M \leftarrow \mathcal{D}_K(C')$; `return` M

To prove that $\overline{\mathcal{SE}}$ is not NM-CPA secure, we present an attack on $\overline{\mathcal{SE}}$ in the form of an adversary $A = (A_1, A_2)$ who violates its non-malleability with probability one using resources polynomial in the security parameter k. It works as follows:

Adversary $A_1^{\overline{\mathcal{E}}_K(\mathcal{LR}(\cdot,\cdot,b))}(k)$
  $C \leftarrow \overline{\mathcal{E}}_K(\mathcal{LR}(0,1,b))$
    Parse C as $x\|C'$ where x is a bit.
    $x' \leftarrow x \oplus 1$
    $\vec{c}[1] \leftarrow x'\|C'$
  Return $(\vec{c}, \varepsilon)$

Adversary $A_2(\vec{p}, \vec{c}, s)$
  If $\vec{p}[1] = 0$
    then return 0
    else return 1

A_1 queries its left-or-right encryption oracle with the messages 0 and 1 to get a ciphertext $C = x\|C'$. It then creates a vector $\vec{c}$ which has only one component, this being the ciphertext formed by flipping the first bit of $x\|C'$. It outputs $\vec{c}$ together with the empty string ε for state information. A_2 has as input $\vec{p}$ with $\vec{p}[1]$ being $\overline{\mathcal{D}}_K(\vec{c}[1]) = \mathcal{D}_K(C)$. It need only see which of the values 0 or 1 the plaintext $\vec{p}[1]$ equals. The adversary is valid because $\vec{c}[1]$ was not an output of the left-or-right encryption oracle. Clearly this adversary has time-complexity poly(k) and

$$\mathbf{Adv}^{\text{nm-cpa}}_{\mathcal{SE},A}(k) \;=\; 1\,.$$

To prove that $\overline{\mathcal{SE}}$ is indeed IND-CPA (resp. INT-PTXT) secure, it suffices to associate with any poly(k)-time adversary A_p (resp. A_c) attacking $\overline{\mathcal{SE}}$ in the IND-CPA (resp. INT-PTXT) sense, a poly(k)-time adversary B_p (resp. B_c) attacking the $\mathcal{SE}$ in the IND-CPA (resp. INT-PTXT) sense, such that

$$\begin{aligned}\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{SE}},A_p}(k) &\le \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},B_p}(k)\\ \mathbf{Adv}^{\text{int-ptxt}}_{\overline{\mathcal{SE}},A_c}(k) &\le \mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{SE},B_i}(k)\,.\end{aligned}$$

The adversaries B_p and B_c work as follows:

Adversary $B_p^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))}(k)$
  For $i = 1, \ldots, q_e + q_d$ do
    When A_p makes a query $(M_{i,0}, M_{i,1})$
    to its left-or-right encryption oracle do
      $A_p \Leftarrow 0\|\mathcal{E}_K(\mathcal{LR}(M_{i,0}, M_{i,1}, b))$
  Return whatever A_p returns.

Adversary $B_c^{\mathcal{E}_K(\cdot),\mathcal{D}_K^*(\cdot)}(k)$
  For $i = 1, \ldots, q_t + q_v(k)$ do
    When A_c makes a query M_i
    to its encryption oracle do
      $A_c \Leftarrow 0\|\mathcal{E}_K(M_i)$
    When A_c makes a query C_i
    to its verification oracle do
      Parse C_i as $b\|C_i'$ where b is a bit.
      $A_c \Leftarrow \mathcal{D}_K^*(C_i')$.

As the code shows, it is easy for B_p (resp. B_c) to use its own oracle to provide A_p (resp. A_c) with the answers to the latter's oracle queries. Thus, B_p (resp. B_c) is successful with the same probability as A_p (resp. A_c) and furthermore, the resource usage of B_p (resp. A_p) and that of B_c (resp. A_c) is the same. Therefore, $\overline{\mathcal{SE}}$ is IND-CPA and INT-PTXT secure. ▌

4 Security of the Composite Schemes

We now present the formal security results for the composite schemes as summarized in Figure 2 and Figure 3. Throughout this section, $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ is a given symmetric encryption scheme which is IND-CPA secure, $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ is a given message authentication scheme which is WUF-CMA or SUF-CMA secure, and $\overline{\mathcal{SE}} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ is a composite scheme according to one of

Security		Weak MAC		Strong MAC	
		Result	Reason	Result	Reason
Privacy	IND-CPA	Insecure	Proposition 4.1	Insecure	Proposition 4.1
	IND-CCA	Insecure	IND-CPA insecure and IND-CCA → IND-CPA	Insecure	IND-CPA insecure and IND-CCA → IND-CPA
	NM-CPA	Insecure	IND-CPA insecure and NM-CPA → IND-CPA	Insecure	IND-CPA insecure and NM-CPA → IND-CPA
Integrity	INT-PTXT	Secure	Theorem 4.3	Secure	Theorems 4.3 and 2.5
	INT-CTXT	Insecure	Proposition 4.4	Insecure	Proposition 4.4

Figure 4: Summary of results for the *Encrypt-and-MAC* composition method.

the three methods we are considering. The presentation below is method by method, and in each case we begin by specifying the method in more detail. We then provide a table which summarizes the results and reasons for them. A reason is either a reference to a theorem, a reference to a proposition, or a brief line of reasoning saying how the result in question can be derived from already proved entries of the same table in combination with results from Section 3.

We make the simplifying assumption that $\mathcal{D}$ never returns $\perp$. It can take any string as input, and the output is always some string. (This is without loss of generality because we can modify $\mathcal{D}$ so that instead of returning $\perp$ it just returns some default message. Security under chosen-plaintext attack is unaffected.) However, $\overline{\mathcal{D}}$ can and will return $\perp$ at times, and this is crucial for integrity.

In presenting a counter-example (meaning a claim that a certain composition method is insecure under some notion of security **A**) we use the following paradigm. We present a symmetric encryption scheme $\mathcal{SE}'$ and a MAC $\mathcal{MA}'$ such that $\mathcal{SE}'$ is IND-CPA secure and $\mathcal{MA}'$ is WUF-CMA or SUF-CMA secure but we can present an attack on the composite scheme based on them showing that the composite scheme does not meet notion **A**. Of course, we make the minimal assumptions that some scheme $\mathcal{SE}$ that is IND-CPA secure, and some scheme $\mathcal{MA}$ that is WUF-CMA or SUF-CMA secure, exist, since otherwise the claim is vacuous. We construct $\mathcal{SE}'$ from $\mathcal{SE}$ and $\mathcal{MA}'$ from $\mathcal{MA}$.

In some cases the constructions are artificial. But what we want to assess is whether it is possible to prove that the composite scheme meets notion **A** assuming *only* that the constituent encryption scheme is IND-CPA secure and the constituent MAC scheme is WUF-CMA or SUF-CMA secure, and a result of the type just explained shows that such a proof is not possible.

4.1 Encrypt-and-MAC

The composite scheme is defined as follows:

Algorithm $\overline{\mathcal{K}}(k)$
$K_e \xleftarrow{R} \mathcal{K}_e(k)$
$K_m \xleftarrow{R} \mathcal{K}_m(k)$
`Return` $\langle K_e, K_m \rangle$

Algorithm $\overline{\mathcal{E}}_{\langle K_e, K_m \rangle}(M)$
$C' \leftarrow \mathcal{E}_{K_e}(M)$
$\tau \leftarrow \mathcal{T}_{K_m}(M)$
$C \leftarrow C' \| \tau$
`Return` C

Algorithm $\overline{\mathcal{D}}_{\langle K_e, K_m \rangle}(C)$
Parse C as $C' \| \tau$
$M \leftarrow \mathcal{D}_{K_e}(C')$
$v \leftarrow \mathcal{V}_{K_m}(M, \tau)$
`If` $v = 1$, `return` M
`else return` $\perp$.

The results about it are summarized in Figure 4. We now proceed to the theorems and propositions mentioned there.

The *Encrypt-and-MAC* composition method does not preserve privacy because the MAC could reveal information about the plaintext. The following makes this precise.

Proposition 4.1 (Encrypt-and-MAC method is not IND-CPA secure) Given a IND-CPA secure symmetric encryption scheme $\mathcal{SE}$ and a WUF-CMA (resp. SUF-CMA) secure message authentication scheme $\mathcal{MA}$, we can construct a message authentication scheme $\mathcal{MA}'$ such that $\mathcal{MA}'$ is WUF-CMA (resp. SUF-CMA) secure, but the composite scheme $\overline{\mathcal{SE}}$ formed by the *Encrypt-and-MAC* composition method based on $\mathcal{SE}$ and $\mathcal{MA}'$ is *not* IND-CPA secure. ▌

Proof of Proposition 4.1: Let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be the given MAC scheme. We define a MAC scheme $\mathcal{MA}'$ which is the same as the given one except that it prepends the first bit of the message to the tag. Formally $\mathcal{MA}' = (\mathcal{K}_m, \mathcal{T}', \mathcal{V}')$ has the same key generation algorithm as the given MAC scheme and the following tagging and verification algorithms:

Algorithm $\mathcal{T}'_K(M)$	Algorithm $\mathcal{V}'_K(M, \sigma)$
Parse M as $x\|M'$ where x is a bit	Parse M as $x\|M'$ where x is a bit
`Return` $x\|\mathcal{T}_K(M)$	Parse σ as $s\|\sigma'$ where s is a bit
	`If` $x = s$ and $\mathcal{V}_K(M, \sigma') = 1$
	`Then return` 1 `else return` 0

It is easy to see that if $\mathcal{MA}$ is WUF-CMA —resp. SUF-CMA— secure then $\mathcal{MA}'$ is WUF-CMA —resp. SUF-CMA— secure. (The formal proof is omitted.) However if $\mathcal{MA}'$ is used as the base message authentication scheme in the *Encrypt-and-MAC* composition method, the resulting symmetric encryption scheme will fail to achieve IND-CPA because the first bit of the message is provided to the adversary via the MAC. The adversary can use this to break the scheme in the IND-CPA sense as follows. It queries its left-or-right encryption oracle $\overline{\mathcal{E}}_{\langle K_e, K_m\rangle}(\mathcal{LR}(\cdot,\cdot,b))$ with two messages M_0, M_1 such that the first bit of M_0 is 0 and the first bit of M_1 is 1. It gets back ciphertext $C = C'\|\tau$ where $\tau = \mathcal{T}'_{K_m}(M_b)$ and $C' = \mathcal{E}_{K_e}(x_b)$. It lets s be the first bit of τ. As per our construction above, s is the first bit of M_b and hence $s = b$, so the adversary returns s. The advantage of this adversary is one. ▌

Since both IND-CCA and NM-CPA imply IND-CPA, this means that this composition method is also *neither* IND-CCA *nor* NM-CPA secure.

The next proposition makes a somewhat stronger statement. Not only do there exists schemes for which the *Encrypt-and-MAC* method fails to provide IND-CPA, but it will fail to be so for most of the commonly defined MACs, including CBC-MAC and HMAC, because the latter are deterministic. When the MAC is deterministic, an adversary can use the MAC present in the ciphertext of the composite scheme to see whether the same message has been encrypted twice, something which should not be possible if the scheme is to meet a strong notion of privacy like IND-CPA.

Proposition 4.2 (Encrypt-and-MAC method is IND-CPA insecure for any deterministic MAC) Let $\mathcal{SE}$ be a IND-CPA secure symmetric encryption scheme, and let $\mathcal{MA}$ be a deterministic WUF-CMA or SUF-CMA secure message authentication scheme. Then, the composite scheme $\overline{\mathcal{SE}}$ obtained from $\mathcal{SE}$ and $\mathcal{MA}$ by the *Encrypt-and-MAC* composition method is *not* IND-CPA secure. ▌

Proof of Proposition 4.2: We describe an attack on the privacy of $\overline{\mathcal{SE}}$. Recall that as per Definition 2.1 the adversary has access to the left-or-right encryption oracle $\overline{\mathcal{E}}_{\langle K_e, K_m\rangle}(\mathcal{LR}(\cdot,\cdot,b))$. In this case, given messages M_0, M_1, the oracle returns $\mathcal{E}_{K_e}(m_b)\|\mathcal{T}_{K_m}(M_b)$. The attack is described by the following adversary:

Adversary $A^{\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\mathcal{LR}(\cdot,\cdot,b))}(k)$
$C_0\|\sigma_0 \leftarrow \overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\mathcal{LR}(0,0,b))$
$C_1\|\sigma_1 \leftarrow \overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\mathcal{LR}(0,1,b))$
`If` $\sigma_0 = \sigma_1$ `then return` 0 `else return` 1

If $b = 0$, then the determinism of the $\mathcal{T}$ function means that $\sigma_0 = \sigma_1$ so the output of A is 0. If $b = 1$, then A outputs 1 unless it happens that the messages 0 and 1 have the same MAC, namely $\mathcal{T}_{K_m}(0) = \mathcal{T}_{K_m}(1)$. But if the latter were true, the message authentication scheme is clearly insecure: we could query the tagging function at 0 and then forge the MAC of 1. So assuming the MAC is WUF-CMA-secure we have that $\overline{\mathcal{SE}}$ is not IND-CPA secure. ▌

The *Encrypt-and-MAC* composition method does preserve integrity of plaintexts. It inherits the integrity of the MAC in a direct way, with no degradation in security. This is independent of the symmetric encryption scheme: whether the latter is secure or not does not affect the integrity of the composite scheme.

Theorem 4.3 (Encrypt-and-MAC method is INT-PTXT secure) Let $\mathcal{SE}$ be a symmetric encryption scheme, let $\mathcal{MA}$ be a message authentication scheme, and let $\overline{\mathcal{SE}}$ be the encryption scheme obtained from $\mathcal{SE}$ and $\mathcal{MA}$ via the *Encrypt-and-MAC* composition method. Then, if $\mathcal{MA}$ is WUF-CMA-secure, then $\overline{\mathcal{SE}}$ is INT-PTXT secure. Concretely,

$$\mathbf{Adv}^{\text{int-ptxt}}_{\overline{\mathcal{SE}}}(k,t,q_e,q_d,\mu_e,\mu_d) \;\le\; \mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA}}(k,t,q_e,q_d,\mu_e,\mu_d)\ .\ \blacksquare$$

The same is true if the MAC is SUF-CMA-secure, by Theorem 2.5.

Proof of Theorem 4.3: Let $\overline{\mathcal{SE}} = (\overline{\mathcal{K}},\overline{\mathcal{E}},\overline{\mathcal{D}})$ be the composite encryption scheme constructed via the *Encrypt-and-MAC* method from the encryption scheme $\mathcal{SE} = (\mathcal{K}_e,\mathcal{E},\mathcal{D})$ and the MAC scheme $\mathcal{MA} = (\mathcal{K}_m,\mathcal{T},\mathcal{V})$. We associate with any adversary A attacking $\overline{\mathcal{SE}}$ in the INT-PTXT-sense a forger F attacking $\mathcal{MA}$ in the WUF-CMA-sense such that

$$\mathbf{Adv}^{\text{int-ptxt}}_{\overline{\mathcal{SE}},A}(k) \;\le\; \mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA},F}(k)$$

and F uses the same resources as A does. This implies Theorem 4.3.

The forger F uses the adversary A to achieve its goal. It has access to the oracles $\mathcal{T}_{K_m}(\cdot)$ and $\mathcal{V}_{K_m}(\cdot,\cdot)$ where K_m is a random key for $\mathcal{MA}$ and will pick a key K_e for the encryption algorithm $\mathcal{E}$. Using this key and its own oracles, it can simulate the encryption oracle $\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot)$ and verification oracle $\overline{\mathcal{D}}^{*}_{\langle K_e,K_m\rangle}(\cdot)$ that A needs, and thus answer A's oracle queries. In more detail, it works as follows:

Adversary $F^{\mathcal{T}_{K_m}(\cdot),\mathcal{V}_{K_m}(\cdot,\cdot)}(k)$
$K_e \xleftarrow{R} \mathcal{K}_e(k)$
`For` $i = 1,\ldots,q_e+q_d$ `do`
When A makes a query M_i to its encryption oracle `do`
$C'_i \leftarrow \mathcal{E}_{K_e}(M_i)$; $\tau_i \leftarrow \mathcal{T}_{K_m}(M_i)$; $A \Leftarrow C'_i\|\tau_i$
When A makes a query C_i to its verification oracle `do`
Parse C_i as $C'_i\|\tau_i$; $M_i \leftarrow \mathcal{D}_{K_e}(C'_i)$; $v_i \leftarrow \mathcal{V}_{K_m}(M_i,\tau_i)$; $A \Leftarrow v_i$

Consider a ciphertext $C_i = C'_i\|\tau_i$ that yields a successful forgery of a new plaintext M_i. This means that M_i was never queried to $\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot)$, which implies that F never queried it to $\mathcal{T}_{K_m}(\cdot)$ either.

Therefore, the pair (M_i, τ_i) is a valid weak forgery, and the above equation is justified. It remains to justify the claims about the resource parameters used by F. The key thing to remember is that, as per our definitions, the resources for both adversaries pertain to the entire experiment which measures their success. ▌

However, the *Encrypt-and-MAC* composition method fails to provide integrity of ciphertexts. This is because there are secure encryption schemes with the property that a ciphertext can be modified without changing its decryption. When such an encryption scheme is used as the base symmetric encryption scheme, an adversary can query the encryption oracle, modify part of the response, and still submit the result to the verification oracle as a valid ciphertext. The following proposition states this result.

Proposition 4.4 (Encrypt-and-MAC method is not INT-CTXT secure) Given a IND-CPA secure symmetric encryption scheme $\mathcal{SE}$ and a WUF-CMA or SUF-CMA secure message authentication scheme $\mathcal{MA}$, we can construct a symmetric encryption scheme $\mathcal{SE}'$ such that $\mathcal{SE}'$ is IND-CPA secure, but the composite scheme $\overline{\mathcal{SE}}$ formed by the *Encrypt-and-MAC* composition method based on $\mathcal{SE}'$ and $\mathcal{MA}$ is *not* INT-CTXT secure. ▌

Proof of Proposition 4.4: Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the given symmetric encryption scheme. We define the scheme $\mathcal{SE}'$ such that $\mathcal{SE}'$ is IND-CPA secure, but the composite scheme $\overline{\mathcal{SE}}$ is not INT-CTXT secure. The idea is similar to that in the proof of Proposition 3.4. A redundant bit prepended to ciphertexts is ignored by $\overline{\mathcal{D}}$, allowing the adversary to form a new valid ciphertext. Here are the details.

The new scheme $\mathcal{SE}' = (\mathcal{K}_e, \mathcal{E}', \mathcal{D}')$ has the same key generation algorithm as that of $\mathcal{SE}$ and the same encryption and decryption algorithms as those of the scheme $\mathcal{SE}'$ defined in the proof of Proposition 3.4. Then, we provide the following adversary A attacking the composite scheme $\overline{\mathcal{SE}}$ constructed based on the schemes $\mathcal{SE}'$ and $\mathcal{MA}$:

Adversary $A^{\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot),\overline{\mathcal{D}}^*_{\langle K_e,K_m\rangle}(\cdot)}(k)$
  $\overline{C} \leftarrow \overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(0)$
  Parse $\overline{C}$ as $0\|C$
  Submit $1\|C$ as a query to the oracle $\overline{\mathcal{D}}^*_{\langle K_e,K_m\rangle}(\cdot)$.

The ciphertext submitted to $\overline{\mathcal{D}}^*_{\langle K_e,K_m\rangle}(\cdot)$ is new, meaning was never output by the encryption oracle $\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot)$, because it begins with a 1 while all outputs of $\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot)$ begin with a 0. Furthermore, it is valid because the decryption algorithm $\mathcal{D}'$ by definition ignores the first bit of any ciphertext it is given. Therefore, A violates the integrity of ciphertexts of $\overline{\mathcal{SE}}$ with probability 1. (Note that this does not violate integrity of plaintexts because the plaintexts underlying ciphertexts $0\|C$ and $1\|C$ are the same.) Finally, we note that the proof that the modified scheme $\mathcal{SE}'$ is still secure against chosen-plaintext attack is easy and is omitted. ▌

4.2 MAC-then-Encrypt

The composite scheme is defined as follows:

Security		Weak MAC		Strong MAC	
		Result	Reason	Result	Reason
Privacy	IND-CPA	Secure	Theorem 4.5	Secure	Theorem 4.5
	IND-CCA	Insecure	NM-CPA insecure and NM-CPA → IND-CCA	Insecure	NM-CPA insecure and NM-CPA → IND-CCA
	NM-CPA	Insecure	Proposition 4.6	Insecure	Proposition 4.6
Integrity	INT-PTXT	Secure	Theorem 4.5	Secure	Theorems 4.5 and 2.5
	INT-CTXT	Insecure	IND-CPA secure and NM-CPA insecure and INT-CTXT∧IND-CPA → NM-CPA	Insecure	IND-CPA secure and NM-CPA insecure and INT-CTXT∧IND-CPA → NM-CPA

Figure 5: Summary of results for the *MAC-then-encrypt* composition method

---

Algorithm $\overline{\mathcal{K}}(k)$
$K_e \xleftarrow{R} \mathcal{K}_e(k)$
$K_m \xleftarrow{R} \mathcal{K}_m(k)$
`Return` $\langle K_e, K_m \rangle$

Algorithm $\overline{\mathcal{E}}_{\langle K_e, K_m \rangle}(M)$
$\tau \leftarrow \mathcal{T}_{K_m}(M)$
$C \leftarrow \mathcal{E}_{K_e}(M \| \tau)$
`Return` C

Algorithm $\overline{\mathcal{D}}_{\langle K_e, K_m \rangle}(C)$
$M' \leftarrow \mathcal{D}_{K_e}(C)$
Parse M' as $M \| \tau$
$v \leftarrow \mathcal{V}_{K_m}(M, \tau)$
`If` $v = 1$, `return` M
`else return` $\perp$.

The results about it are summarized in Figure 5. We now proceed to the theorems and propositions mentioned there.

The *MAC-then-encrypt* composition method preserves both privacy against chosen-plaintext attack and integrity of plaintexts, as stated in the following theorem.

Theorem 4.5 (MAC-then-encrypt method is both INT-PTXT and IND-CPA secure) Let $\mathcal{MA}$ be a message authentication scheme, and let $\mathcal{SE}$ be a symmetric encryption scheme secure against chosen-plaintext attacks. Let $\overline{\mathcal{SE}}$ be the encryption scheme obtained from $\mathcal{SE}$ and $\mathcal{MA}$ via the *MAC-then-encrypt* composition method. Then, if $\mathcal{MA}$ is WUF-CMA secure, then $\overline{\mathcal{SE}}$ is INT-PTXT secure. Furthermore, if $\mathcal{SE}$ is IND-CPA secure, then so is $\overline{\mathcal{SE}}$. Concretely,

$$\begin{aligned} \mathbf{Adv}^{\text{int-ptxt}}_{\overline{\mathcal{SE}}}(k, t, q_e, q_d, \mu_e, \mu_d) &\leq \mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA}}(k, t, q_e, q_d, \mu_e, \mu_d) \\ \mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{SE}}}(k, t, q, \mu) &\leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(k, t, q, \mu + ql) \end{aligned}$$

where we are assuming that the length of a tag in the scheme $\mathcal{MA}$ is l bits. ∎

Proof of Theorem 4.5: Let $\overline{\mathcal{SE}} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ be the composite encryption scheme constructed via the *MAC-then-encrypt* method from the encryption scheme $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ and the MAC scheme $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$. We associate with any adversary A attacking $\overline{\mathcal{SE}}$ in the INT-PTXT-sense a forger F attacking $\mathcal{MA}$ in the WUF-CMA-sense such that

$$\mathbf{Adv}^{\text{int-ptxt}}_{\overline{\mathcal{SE}}, A}(k) \leq \mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA}, F}(k)$$

and F uses the same resources as A does. Then, the first equation of Theorem 4.5 follows.

The forger F uses the adversary A to achieve its goal. It has access to the oracles $\mathcal{T}_{K_m}(\cdot)$ and $\mathcal{V}_{K_m}(\cdot, \cdot)$ where K_m is a random key for $\mathcal{MA}$, and will pick a key K_e for the encryption algorithm

$\mathcal{E}$. Using this key and its own oracles, it can simulate the encryption oracle $\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot)$ and the verification oracle $\overline{\mathcal{D}}^*_{\langle K_e,K_m\rangle}(\cdot)$ that A needs, and thus answer A's oracle queries. In more detail, it works as follows:

Adversary $F^{\mathcal{T}_{K_m}(\cdot),\mathcal{V}_{K_m}(\cdot,\cdot)}(k)$
  $K_e \xleftarrow{R} \mathcal{K}_e(k)$
  `For` $i = 1,\ldots,q_e + q_d$ `do`
    When A makes a query M_i to its encryption oracle `do`
      $M_i' \leftarrow M_i \| \mathcal{T}_{K_m}(M_i)$; $C_i' \leftarrow \mathcal{E}_{K_e}(M_i')$; $A \Leftarrow C_i'$
    When A makes a query C_i to its verification oracle `do`
      $M_i' \leftarrow \mathcal{D}_{K_e}(C_i)$; Parse M_i' as $M_i \| \tau_i$; $v_i \leftarrow \mathcal{V}_{K_m}(M_i,\tau_i)$; $A \Leftarrow v_i$

Consider a ciphertext C_i that yields a successful forgery of a new plaintext M_i. Since M_i is new, the pair (M_i,τ_i) where τ_i is obtained from appropriately parsing $\mathcal{D}_{K_e}(C_i)$ as described in the above algorithm is a valid weak forgery. Thus, the above equation follows. It remains to justify the claims about the resource parameters used by F. Note that the queries made by F to its tag oracle are exactly those made by A to its encryption oracle. On the other hand, since the length of a plaintext is always at most the length of a corresponding ciphertext, the length of a query M_i,τ_i made by F to its verification oracle is at most the length C_i of the corresponding query made by A to its verification oracle.

We now proceed to the proof of the second claim. We associate with any adversary A attacking $\overline{\mathcal{SE}}$ in the IND-CPA-sense an adversary A_p such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{SE}},A}(k) \;\le\; \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},A_p}(k)$$

and A_p uses the same resources as A except for an extra ql bits in the total length of its oracle queries. Then, the second equation of Theorem 4.5 follows.

The adversary A_p uses the adversary A to achieve its goal. It has access to the oracle $\mathcal{E}_{K_e}(\mathcal{LR}(\cdot,\cdot,b))$ where K_e is a random key for $\mathcal{SE}$, and will pick a key K_m for the tagging algorithm $\mathcal{T}$. Using this key and its own oracle, it can simulate the left-or-right encryption oracle $\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot)$ that A needs, and thus answer A's oracle queries. In more detail, it works as follows:

Adversary $A_p^{\mathcal{E}_{K_e}(\mathcal{LR}(\cdot,\cdot,b))}(k)$
  $K_m \xleftarrow{R} \mathcal{K}_m(k)$
  `For` $i = 1,\ldots,q$ `do`
    When A makes a query $(M_{i,0},M_{i,1})$ to its left-or-right encryption oracle `do`
      $\tau_0 \leftarrow \mathcal{T}_{K_m}(M_{i,0})$; $\tau_1 \leftarrow \mathcal{T}_{K_m}(M_{i,1})$
      $M_0 \leftarrow M_{i,0} \| \tau_0$; $M_1 \leftarrow M_{i,1} \| \tau_1$
      $C_i \leftarrow \mathcal{E}_{K_e}(\mathcal{LR}(M_0,M_1,b))$
      $A \Leftarrow C_i$
  $A \Rightarrow b'$
  `Return` b'

For each query, A_p computes the tags of both messages queried by A to generate inputs to its oracle and then lets its oracle decide which input to encrypt. It then outputs A's guess as its own. The advantages of A_p and A are the same. It remains to justify the claims about the resource parameters used by A_p. The length of a query made by A_p to its left-or-right encryption oracle is

greater than the length of the corresponding query made by A by the length of the added tag, and so ql bits are added to the total length of the queries. ▌

The base encryption scheme might be malleable, and this will be inherited by the composite scheme.

Proposition 4.6 (MAC-then-encrypt method is not NM-CPA secure) Given a IND-CPA secure symmetric encryption scheme $\mathcal{SE}$ and a WUF-CMA or SUF-CMA secure message authentication scheme $\mathcal{MA}$, we can construct a symmetric encryption scheme $\mathcal{SE}'$ such that $\mathcal{SE}'$ is IND-CPA secure, but the composite scheme $\overline{\mathcal{SE}}$ formed by the *MAC-then-encrypt* composition method based on $\mathcal{SE}'$ and $\mathcal{MA}$ is *not* NM-CPA secure. ▌

Proof of Proposition 4.6: Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be the given symmetric encryption scheme. We define the scheme $\mathcal{SE}' = (\mathcal{K}_e, \mathcal{E}', \mathcal{D}')$ as in the proof of Proposition 3.4, namely it has the same key generation algorithm as $\mathcal{SE}$ and the following encryption and decryption algorithms

Algorithm $\mathcal{E}'_K(M)$	Algorithm $\mathcal{D}'_K(C)$
$C \leftarrow \mathcal{E}_K(M)$	Parse C as $b\|C'$ where b is a bit
Return $0\|C$	$M \leftarrow \mathcal{D}_K(C')$; return M

Let $\overline{\mathcal{SE}}$ be the scheme obtained by the *MAC-then-encrypt* composition method based on $\mathcal{SE}'$ and $\mathcal{MA}$. It is easy to see that the attack of the proof of Proposition 3.4 applies again to show that $\overline{\mathcal{SE}}$ is insecure in the NM-CPA sense. Similarly the proof that the $\mathcal{SE}'$ is IND-CPA is also the same as in the proof of Proposition 3.4. ▌

Since IND-CCA implies NM-CPA, this composition method is also *not* IND-CCA secure. Furthermore, the fact that it is IND-CPA secure but not NM-CPA secure implies that it is not INT-CTXT secure.

4.3 Encrypt-then-MAC

The composite scheme is defined as follows:

Algorithm $\overline{\mathcal{K}}(k)$	Algorithm $\overline{\mathcal{E}}_{\langle K_e, K_m\rangle}(M)$	Algorithm $\overline{\mathcal{D}}_{\langle K_e, K_m\rangle}(C)$
$K_e \xleftarrow{R} \mathcal{K}_e(k)$	$C' \leftarrow \mathcal{E}_{K_e}(M)$	Parse C as $C'\|\tau'$
$K_m \xleftarrow{R} \mathcal{K}_m(k)$	$\tau' \leftarrow \mathcal{T}_{K_m}(C')$	$M \leftarrow \mathcal{D}_{K_e}(C')$
Return $\langle K_e, K_m\rangle$	$C \leftarrow C'\|\tau'$	$v \leftarrow \mathcal{V}_{K_m}(C', \tau')$
	Return C	If $v = 1$, return M
		else return $\perp$.

The results about it are summarized in Figure 6. We now proceed to the theorems and propositions mentioned there.

The security results for the two composition methods we have covered so far, i.e. *Encrypt-and-MAC* and *MAC-then-encrypt*, hold whether or not we assume the base MAC scheme to be weakly or strongly unforgeable. For the *Encrypt-and-MAC* composition method, however, we have different security results depending on our assumption about the MAC as indicated in Figure 2 and Figure 3. For clarity, we seperate the results accordingly here.

The following theorem states that the *Encrypt-and-MAC* composition method is IND-CPA and INT-PTXT secure assuming that the base MAC scheme is weakly unforgeable.

Security		Weak MAC		Strong MAC	
		Result	Reason	Result	Reason
Privacy	IND-CPA	Secure	Theorem 4.7	Secure	Theorem 4.9
	IND-CCA	Insecure	NM-CPA insecure and NM-CPA → IND-CCA	Secure	Theorem 4.9
	NM-CPA	Insecure	Proposition 4.6	Secure	IND-CCA secure and IND-CCA → NM-CPA
Integrity	INT-PTXT	Secure	Theorem 4.7	Secure	INT-CTXT secure and INT-CTXT → INT-PTXT
	INT-CTXT	Insecure	IND-CPA secure and NM-CPA insecure and INT-CTXT∧IND-CPA → NM-CPA	Secure	Theorem 4.9

Figure 6: Summary of results for the *encrypt-then-MAC* composition method

---

Theorem 4.7 (Encrypt-then-MAC method is IND-CPA and INT-PTXT secure) Let $\mathcal{SE}$ be a symmetric encryption scheme, and let $\mathcal{MA}$ be a message authentication scheme. Let $\overline{\mathcal{SE}}$ be the authenticated encryption scheme obtained from $\mathcal{SE}$ and $\mathcal{MA}$ via the *encrypt-then-MAC* composition method. Then, if $\mathcal{MA}$ is WUF-CMA secure, then $\overline{\mathcal{SE}}$ is INT-PTXT secure. And if $\mathcal{SE}$ is IND-CPA secure, then so is $\overline{\mathcal{SE}}$. Concretely,

$$\begin{aligned}\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{SE}}}(k,t,q,\mu) &\leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(k,t,q,\mu)\\ \mathbf{Adv}^{\text{int-ptxt}}_{\overline{\mathcal{SE}}}(k,t,q_e,q_d,\mu_e,\mu_d) &\leq \mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA}}(k,t,q_e,q_d,\mu_e+q_e l,\mu_d)\end{aligned}$$

where we are assuming that the length of a ciphertext in the scheme $\mathcal{SE}$ is l bits more than the length of the corresponding plaintext. ▌

Proof of Theorem 4.7: Let $\overline{\mathcal{SE}} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ be a composite encryption scheme constructed via the encrypt-then-MAC method from the encryption scheme $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ and the MAC scheme $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$. We associate with any adversary A attacking $\overline{\mathcal{SE}}$ in the IND-CPA-sense an adversary A_p such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{SE}},A}(k) \;\leq\; \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE},A_p}(k)$$

and A_p uses the same resources as A does. Then, the first equation of Theorem 4.7 follows.

The adversary A_p uses the adversary A to achieve its goal. It has access to the oracle $\mathcal{E}_{K_e}(\mathcal{LR}(\cdot,\cdot,b))$ where K_e is a random key for $\mathcal{SE}$, and will pick a key K_m for the tagging algorithm $\mathcal{T}$. Using this key and its own oracle, it can simulate the left-or-right encryption oracle $\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot)$ that A needs, and thus answer A's oracle queries. In more detail, it works as follows:

Adversary $A_p^{\mathcal{E}_{K_e}(\mathcal{LR}(\cdot,\cdot,b))}(k)$
  $K_m \xleftarrow{R} \mathcal{K}_m(k)$
  `For` $i = 1,\ldots,q$ `do`
    When A makes a query $(M_{i,0}, M_{i,1})$ to its left-or-right encryption oracle `do`
      $C_i \leftarrow \mathcal{E}_{K_e}(\mathcal{LR}(M_{i,0}, M_{i,1}, b))$; $\tau_i \leftarrow \mathcal{T}_{K_m}(C_i)$; $A \Leftarrow C_i \| \tau_i$

$A \Rightarrow b'$
`Return` b'

Clearly, if A can successfully determine the bit b, so can A_p. Thus, the equation above follows. The claims about the resource parameters used by A_p are easily checked.

We proceed to the proof of the second claim. We associate with any adversary A attacking integrity of plaintexts against $\overline{\mathcal{SE}}$ a forger F such that

$$\mathbf{Adv}^{\text{int-ptxt}}_{\overline{\mathcal{SE}},A}(k) \;\leq\; \mathbf{Adv}^{\text{wuf-cma}}_{\mathcal{MA},F}(k)$$

and F uses the same resources as A does except for an extra $q_e l$ bits in the total length of queries to the tagging oracle. Then, the second equation of Theorem 4.7 follows.

The forger F uses the adverary A to achieve its goal. It has access to the oracles $\mathcal{T}_{K_m}(\cdot)$ and $\mathcal{V}_{K_m}(\cdot,\cdot)$, and will pick a key K_e for the encryption algorithm. Using this key and its own oracles, it can simulate the encryption oracle $\overline{\mathcal{E}}_{\langle K_e,K_m\rangle}(\cdot)$ and verification oracle $\overline{\mathcal{D}}^{*}_{\langle K_e,K_m\rangle}(\cdot)$ that A needs, and thus answer A's oracle queries. In more details, it works as follows:

Adversary $F^{\mathcal{T}_{K_m}(\cdot),\mathcal{V}_{K_m}(\cdot,\cdot)}(k)$
  $K_e \xleftarrow{R} \mathcal{K}_e(k)$
  `For` $i = 1,\ldots,q_e + q_d$ `do`
    When A makes a query M_i to its encryption oracle `do`
      $C'_i \leftarrow \mathcal{E}_{K_e}(M_i)$; $\tau_i \leftarrow \mathcal{T}_{K_m}(C'_i)$; $A \Leftarrow C'_i \| \tau_i$
    When A makes a query C_i to its verification oracle `do`
      Parse C_i as $C'_i \| \tau'_i$; $v_i \leftarrow \mathcal{V}_{K_m}(C'_i, \tau'_i)$; $A \Leftarrow v_i$

Let $C_i = C'_i \| \tau'_i$ be a ciphertext submitted by A to its verification oracle that leads to A's violating the integrity of plaintexts of $\overline{\mathcal{SE}}$ and let $M_i = \mathcal{D}_K(C'_i)$. Then $\mathcal{V}_{K_m}(C'_i, \tau'_i) = 1$ and M_i was not a query of A to its encryption oracle. The unique decryptability of $\mathcal{SE}$ means that C'_i was not a query of F to its tagging oracle, so the query C'_i, τ'_i made by F to its verification oracle leads to F's making a successful weak forgery. Thus, the equation above follows. It remains to justify the claims about the resource parameters used by F. When A makes a query M to its encryption oracle, F queries its tagging oracle on the corresponding ciphertext C and by assumption $|C| = l + |M|$, and this accounts for the extra $q_e l$ bits in the total length of queries to the tagging oracle. ▮

However a weakly unforgeable base MAC scheme is not enough to obtain a NM-CPA secure composite scheme under this composition method.

Proposition 4.8 (Encrypt-then-MAC method with a WUF-CMA-secure MAC is not NM-CPA secure) Given a IND-CPA secure symmetric encryption scheme $\mathcal{SE}$ and a WUF-CMA secure message authentication scheme $\mathcal{MA}$, we can construct a message authentication scheme $\mathcal{MA}'$ such that $\mathcal{MA}'$ is WUF-CMA secure, but the composite scheme $\overline{\mathcal{SE}}$ formed by the *encrypt-then-MAC* composition method based on $\mathcal{SE}$ and $\mathcal{MA}'$ is *not* NM-CPA secure. ▮

Proof of Proposition 4.8: Let $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ be the given MAC scheme. We define the scheme $\mathcal{MA}'$ such that $\mathcal{MA}'$ is WUF-CMA secure, but the composite scheme $\overline{\mathcal{SE}}$ formed by the *encrypt-then-MAC* composition method based on $\mathcal{SE}$ and $\mathcal{MA}'$ is not NM-CPA secure. The idea is that a redundant bit appended to the tag that the tagging algorithm generates is ignored by the verification algorithm $\mathcal{V}$. The resulting MAC scheme will still be WUF-CMA secure, but the composite encryption scheme will become malleable. Here are the details.

The new MAC scheme $\mathcal{MA}' = (\mathcal{K}, \mathcal{T}', \mathcal{V}')$ has the same key generation algorithm as that of the original scheme, but its tagging and verifying algorithms are as follows:

Algorithm $\mathcal{T}'_K(M)$	Algorithm $\mathcal{V}'_K(M, \tau)$
$\tau \leftarrow \mathcal{T}_K(M)$	Parse τ as $\tau' \| b$ where b is a bit.
Return $\tau \| 0$	Return $\mathcal{V}_K(M, \tau')$

To prove that the composite scheme $\overline{\mathcal{SE}}$ constructed from $\mathcal{SE}$ and $\mathcal{MA}'$ using encrypt-then-MAC method is not NM-CPA secure, we present an attack on $\overline{\mathcal{SE}}$ in the form of an adversary $A = (A_1, A_2)$ who violates the non-malleability of $\overline{\mathcal{SE}}$ with high probability. It works as follows:

Adversary $A_1^{\overline{\mathcal{E}}_{\langle K_e, K_m\rangle}(\mathcal{LR}(\cdot,\cdot,b))}(k)$	Adversary $A_2(\vec{p}, \vec{c}, s)$
$C \leftarrow \overline{\mathcal{E}}_{\langle K_e, K_m\rangle}(\mathcal{LR}(0, 1, b))$	If $\vec{p}[1] = 0$
Parse C as $C' \| \tau \| x$ where x is a bit.	then return 0
$x' \leftarrow x \oplus 1$	else return 1.
$\vec{c}[1] \leftarrow C' \| \tau \| x'$	
Return $(\vec{c}, \varepsilon)$	

A_1 queries its left-or-right encryption oracle with the messages 0 and 1 to get a ciphertext $C = C' \| \tau \| x$. It then creates a vector $\vec{c}$ which has only one component, this being the ciphertext formed by flipping the last bit of $C' \| \tau \| x$. It outputs $\vec{c}$ together with the empty string ε for state information. A_2 has as input $\vec{p}$ with $\vec{p}[1]$ being $\overline{\mathcal{D}}_{\langle K_e, K_m\rangle}(\vec{c}[1]) = \mathcal{D}_{K_e}(C')$. It need only see which of the values 0 or 1 the plaintext $\vec{p}[1]$ equals. The adversary is valid because $\vec{c}[1]$ was not an output of the left-or-right encryption oracle. Clearly this adversary has time-complexity poly(k) and

$$\mathbf{Adv}^{\text{nm-cpa}}_{\overline{\mathcal{SE}},A}(k) \;=\; 1\,.$$

The proof that $\mathcal{MA}'$ is WUF-CMA-secure is easy and is omitted. ▌

Furthermore, since IND-CCA and INT-CTXT $\wedge$ IND-CPA imply NM-CPA, this composition method is *neither* IND-CCA *nor* INT-CTXT secure when the base MAC scheme is only assumed to be weakly unforgeable.

The following theorem implies that the *encrypt-then-MAC* composition method is IND-CPA, IND-CCA, NM-CPA, INT-PTXT and INT-CTXT secure assuming a strongly unforgeable base MAC scheme. For brevity, we do not state explicitly in the theorem that this composition method is also NM-CPA and NM-CCA secure because it follows directly from the results proven in [3], i.e. that IND-CCA security implies NM-CPA and NM-CCA security. Also, we do not state explicitly here that the composition method is INT-PTXT secure since INT-CTXT security implies INT-PTXT security.

Theorem 4.9 (Encrypt-then-MAC method with a SUF-CMA-secure MAC is INT-CTXT, IND-CPA, and IND-CCA secure) Let $\mathcal{SE}$ be a symmetric encryption scheme, and let $\mathcal{MA}$ be a message authentication scheme. Let $\overline{\mathcal{SE}}$ be the authenticated encryption scheme obtained from $\mathcal{SE}$ and $\mathcal{MA}$ via the *encrypt-then-MAC* composition method. Then, if $\mathcal{MA}$ is SUF-CMA secure, then $\overline{\mathcal{SE}}$ is INT-CTXT secure. If $\mathcal{SE}$ is IND-CPA secure, then so is $\overline{\mathcal{SE}}$. And if we have both of the previous conditions, then $\overline{\mathcal{SE}}$ is IND-CCA secure. Concretely,

$$\begin{aligned}
\mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathcal{SE}}}(k,t,q,\mu) &\le \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(k,t,q,\mu)\\
\mathbf{Adv}^{\text{int-ctxt}}_{\overline{\mathcal{SE}}}(k,t,q_e,q_d,\mu_e,\mu_d) &\le \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(k,t,q_e,q_d,\mu_e+q_e l,\mu_d)\\
\mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{SE}}}(k,t,q_e,q_d,\mu_e,\mu_d) &\le 2\cdot\mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(k,t,q_e,q_d,\mu_e+q_e l,\mu_d) + \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e)
\end{aligned}$$

where we are assuming that the length of a ciphertext in the scheme $\mathcal{SE}$ is l bits more than the length of the corresponding plaintext. ▌

Proof of Theorem 4.9: The proof of the first claim is the same as the proof of the first claim in Theorem 4.7 and we omit the details. The third claim is a corollary of Theorem 3.2 and the first and second claims of Theorem 4.9. It remains to prove the second claim.

We associate with any adversary A attacking $\overline{\mathcal{SE}}$ in the INT-CTXT-sense a forger F such that

$$\mathbf{Adv}^{\text{int-ctxt}}_{\overline{\mathcal{SE}},A}(k) \;\leq\; \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA},F}(k)$$

and F uses the same resources as A does. Then, the second equation of Theorem 4.7 follows.

The forger F is the same as the one used to prove that the composite scheme is INT-PTXT secure assuming that the base MAC scheme is weakly unforgeble, namely the one described in the proof of the second claim of Theorem 4.7. We need to check that it works here as well. Let $C_i = C_i' \| \tau_i'$ be a ciphertext submitted by A to its verification oracle that leads to A's violating the integrity of ciphertexts of $\overline{\mathcal{SE}}$. Then $\mathcal{V}_{K_m}(C_i', \tau_i') = 1$, and $C_i = C_i' \| \tau_i'$ was not a reply of A's encryption oracle. We claim that τ_i' was not a reply to a query C_i' made by F to its tagging oracle. (This is true because F invokes its tagging oracle to compute replies to encryption oracle queries of A, and had τ_i' been a reply of the tagging oracle to query C_i' then $C_i = C_i' \| \tau_i'$ would have been sent as a reply to the corresponding encryption oracle query.) This means that C_i', τ_i' is a valid strong forgery. The claims about resources are justified in the same manner as in the proof of the second claim in Theorem 4.7. ▌

References

[1] M. Bellare, R. Canetti and H. Krawczyk, "Keying hash functions for message authentication," *Advances in Cryptology – Crypto '96*, Lecture Notes in Computer ScienceVol. 1109, N. Koblitz ed., Springer-Verlag, 1996.

[2] M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation," *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.

[3] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology – Crypto '98*, Lecture Notes in Computer ScienceVol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[4] M. Bellare, J. Kilian, P. Rogaway, "The security of the cipher block chaining message authentication code," *Advances in Cryptology – Crypto '94*, Lecture Notes in Computer ScienceVol. 839, Y. Desmedt ed., Springer-Verlag, 1994.

[5] M. Bellare and P. Rogaway, "Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography," *Advances in Cryptology – ASIACRYPT '00*, Lecture Notes in Computer ScienceVol. ??, T. Okamoto ed., Springer-Verlag, 2000.

[6] M. Bellare and A. Sahai, "Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization," *Advances in Cryptology – Crypto '99*, Lecture Notes in Computer ScienceVol. 1666, M. Wiener ed., Springer-Verlag, 1999.

[7] J. Black, S. Halevi, H. Krawczyk, T. Krovetz and P. Rogaway, "UMAC: Fast and secure message authentication," *Advances in Cryptology – Crypto '99*, Lecture Notes in Computer ScienceVol. 1666, M. Wiener ed., Springer-Verlag, 1999.

[8] J. Black, S. Halevi, H. Krawczyk, T. Krovetz and P. Rogaway, "Update on UMAC Fast message authentication," Manuscript, May 2000. Available at `http://www.cs.ucdavis.edu/~rogaway/umac/`.

[9] A. Desai, "New paradigms for constructing symmetric encryption schemes secure against chosen ciphertext attack," *Advances in Cryptology – Crypto '00*, Lecture Notes in Computer ScienceVol. 1880, M. Bellare ed., Springer-Verlag, 2000.

[10] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," *Proceedings of the 23rd Annual Symposium on the Theory of Computing*, ACM, 1991.

[11] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," to appear in *SIAM J. Comput.*

[12] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Science*, Vol. 28, 1984, pp. 270-299.

[13] S. Goldwasser, S. Micali and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal of Computing*, Vol. 17, No. 2, pp. 281–308, April 1988.

[14] C. Jutla, "Encryption modes with almost free message integrity," Report 2000/039, *Cryptology ePrint Archive,* `http://eprint.iacr.org/`, August 2000.

[15] J. Katz and M. Yung, "Complete characterization of security notions for probabilistic private-key encryption," *Proceedings of the 32nd Annual Symposium on the Theory of Computing*, ACM, 2000.

[16] J. Katz and M. Yung, "Unforgeable Encryption and Adaptively Secure Modes of Operation," *Fast Software Encryption '00*, Lecture Notes in Computer ScienceVol. ??, B. Schneier ed., Springer-Verlag, 2000.

[17] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," Request for Comments 2406, November 1998.

[18] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," *Proceedings of the 22nd Annual Symposium on the Theory of Computing*, ACM, 1990.

[19] E. Petrank and C. Rackoff, "CBC MAC for real time data sources," *Journal of Cryptology*, Vol. 13, No. 3, 2000, pp. 315–338.

[20] C. Rackoff and D. Simon, "Non-Interactive zero-knowledge proof of knowledge and chosen ciphertext attack," *Advances in Cryptology – Crypto '91*, Lecture Notes in Computer ScienceVol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.