Reconsidering Generic Composition: the Tag-then-Encrypt case

Francesco Berti, Olivier Pereira, Thomas Peters

ICTEAM/ELEN/Crypto Group
Université catholique de Louvain
B-1348 Louvain-la-Neuve, Belgium
emails: {francesco.berti,thomas.peters,olivier.pereira}@uclouvain.be

Abstract. Authenticated Encryption (AE) achieves confidentiality and authenticity, the two most fundamental goals of cryptography, in a single scheme. A common strategy to obtain AE is to combine a Message Authentication Code (MAC) and an encryption scheme, either nonce-based or *iv*-based. Out of the 180 possible combinations, Namprempre et al. [20] proved that 12 were secure, 164 insecure and 4 were left unresolved: A10, A11 and A12 which use an *iv*-based encryption scheme and N4 which uses a nonce-based one. The question of the security of these composition modes is particularly intriguing as N4, A11, and A12 are more efficient than the 12 composition modes that are known to be provably secure.
We prove that: (*i*) N4 is not secure in general, (*ii*) A10, A11 and A12 have equivalent security, (*iii*) A10, A11, A12 and N4 are secure if the underlying encryption scheme is either misuse-resistant or "message malleable", a property that is satisfied by many classical encryption modes, (*iv*) A10, A11 and A12 are insecure if the underlying encryption scheme is stateful or untidy. All the results are quantitative.

1 Introduction

Authenticated encryption and generic composition. From its start, the goal of cryptography is to prevent that anyone but the intended receiver can read a message (privacy) and that anyone can send a message impersonating someone else (authenticity). In order to answer this privacy (resp. authenticity) requirement, encryption schemes (resp. Message Authentication Codes (MACs)) were designed independently. When there is a need for both privacy and authenticity, Authenticated Encryption (AE) can be used [6,15,17,5]. Moreover, AE may be used to authenticate associated data (AD), which are data attached to a message which do not need to be private, but do need to be authenticated (e.g., message header [25]). We suppose that both the sender and the receiver share the same private key (symmetric scenario).
There are two possible ways to create an AE scheme: the first is to design it from scratch, using a single key, and the second is to combine an Encryption scheme with a MAC. Examples of the first path are AES-GCM [12], AES-CCM [19], CHACHA20_POLY305 [21] (used in TLS 1.3 [13]), SCT [24] and the

CAESAR candidates [7]. When following the second path, the problem is to decide how to compose the ingredients. This problem is called *generic composition* and was introduced and studied first by Bellare and Namprempre [5]. They and Krawczyk proved the well-known result that *Encrypt-then*-MAC is secure [6,17]. Namprempre et al. have made a deeper analysis [20], which considered in detail the assumptions on the Encryption scheme, whether it is iv-based (ivE [with the iv randomly picked]) or nonce-based (nE [with the nonce n never repeated]) and assumed that the MACs are PRFs. Out of all the possible composition modes, 12 (9 with ivE, 3 with nE) were proved to be secure, 164 to be insecure and 4 were unresolved: N4 which uses a nE and A10, A11, A12 which use an ivE. These four modes, which are depicted in Fig. 1, are based on the Tag-then-Encrypt paradigm: given a nonce n, an associated data a and a message m, the resulting AEs simply output $c = \mathsf{Enc}^{n}_{k_E}(m\|\tau)$ or $c = \mathsf{Enc}^{iv}_{k_E}(m\|\tau)$ for some n/iv, where τ is the tag provided by the MAC, and is computed either as $\mathsf{Mac}_{k_M}(a, m)$ or as $\mathsf{Mac}_{k_M}(m)$ depending on the mode. When an ivE scheme is used, the iv is computed using a PRF MAC that takes as input either n or (n, a). Interestingly three of these modes (N4, A11 and A12) use the n, a and m only once in total during both the computation of iv ($\mathsf{Mac}^{\mathsf{IV}}_{k_M}$) and τ ($\mathsf{Mac}^{\mathsf{Tag}}_{k_M}$), which makes them the most efficient among all Tag-then-Encrypt schemes. In this paper, we investigate the security of these four composition modes, focusing on ciphertext integrity, as Namprempre et al. already established the expected confidentiality guarantees.

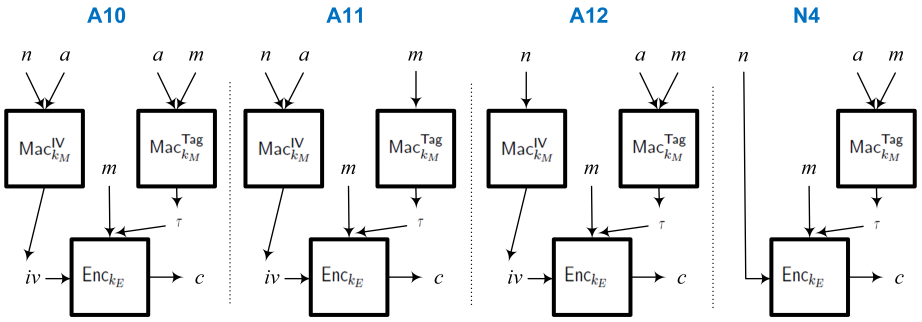

Fig. 1. The four modes A10, A11, A12, and N4.

Our contribution Our investigation gives several new results.

First, the mode N4 does not guarantee ciphertext integrity in general, and we offer a counterexample. The idea of this counterexample is to carefully inject a kind of Trojanin the nE encryption scheme, which can only be activated during the decryption queries using well-crafted ciphertext. The Trojan is triggered through the nonce and a block of the message.

Second, we show that A10, A11 and A12 have equivalent security, by offering security reductions between these three modes. Different techniques are used in

these reductions, which are based on the uniqueness of the nonce and, in other cases, recrafting these nonces.

Third, we push our analysis further, by investigating the security of these 3 modes by making some additional hypothesis on the ivE scheme. We found that these modes are secure if the ivE scheme is either misuse-resistant (that is, repeated nonces can only lead to repeated ciphertexts without further security degradation) or "message-malleable" (that is, given a triple (iv, m, c) with $c = \mathsf{Enc}^{iv}(m)$, it is possible to compute correctly every other triple (iv, m', c') with $c' = \mathsf{Enc}^{iv}(m')$ [resp. $m' = \mathsf{Dec}^{iv}(c')$] for the same iv from m' [resp. c']). Many common schemes, like CTR and OFB [14], CHACHA20 [21] or any other stream ciphers, are "message-malleable", thus we have proved that the three composition modes are secure if implemented with these encryption schemes. This is another evidence of the "*generic composition's sensitivity to definitional and algorithmic adjustments*" [20]. While the proof for misuse-resistant ivE-schemes is relatively straightforward, the proof for "message-malleable" ivE is more interesting as it uses a reduction of a INT-CTXT (ciphertext integrity) adversary to a CPA (Chosen Plaintext Attack) adversary and not only to the properties of the MAC schemes. Interestingly, the N4 mode also becomes secure when the same extra requirements are made for the nE encryption scheme. With respect to the Namprempre et al. [20], we have still to use an additional hypothesis (they used Knowledge of Tag [KoT]), but ours are much easier to prove although they are less general.

Fourth, we find two insecure variants for all three modes, one if the ivE encryption scheme is not tidy, the other if it is stateful. Although Namprempre et al. [20] already used tidiness in security proofs, our ivE scheme correctly encrypts the tag and it decrypts in the "natural" way. Thus, our analysis supports the idea that tidiness is also a security property (already present in Namprempre et al. [20] and in Paterson et al. [23], with respect to CRD). Concerning the attack using a secure stateful scheme (AE stateful schemes were defined by Bellare et al. [3] and their security redefined by Rogaway and Zhang [28]), the idea is to use the state in order to emulate the trojan approach that was used in our attack against mode N4. Namprempre et al. considered only stateless schemes, but it is interesting to see how the security of a mode may depend on the fact of being stateful or stateless. Moreover, stateful AE schemes are an interesting subject of studies [23,4,16,11].

There is an extended version of this paper [9] containing a more detailed background and all the proofs.

Structure of the paper We give a section introducing all the notions we need (Sec. 2); after that we present the four modes N4, A10, A11 and A12 which we investigate (Sec. 3). Then, we show the proof that mode N4 is not secure (Sec. 4) and the security relations among modes A10, A11 and A12 (Sec. 5). After that, we prove that these modes are secure if we add some hypothesis on the ivE scheme (Sec. 6) and we end analyzing our insecure variants of modes A10, A11, and A12 (Sec. 7).

2 Background

2.1 Notations

We use finite binary strings. The length of the string x is denoted by $|x|$ and the concatenation of the strings x and y is denoted by $x\|y$. The set of all finite strings is denoted by $\{0,1\}^*$. We denote the set of all n-bit strings as $\{0,1\}^n$ and the set of strings of at most n bits as $\{0,1\}^{\leq n}$. Given a string $x = (x_1, x_2, ..., x_l)$ of l bits, we denote with $\pi_t(x)$ the string $(x_1, ..., x_T)$ where $T = min(|x|, t)$.
We reserve calligraphic notation for sets. In particular we denote with $\mathcal{K}, \mathcal{N}$, $\mathcal{IV}$, $\mathcal{A}$, $\mathcal{M}$, $\mathcal{TW}$, $\mathcal{T}$, $\mathcal{X}$ and $\mathcal{C}$ respectively the *key space, nonce space, iv-space, associated data space, message space, tweak space, tag space, input space of the* MAC and the *ciphertext space*. We suppose that $\mathcal{M} = \mathcal{A} = \{0,1\}^*$, that is, these spaces contain all the finite binary strings. We suppose that $\mathcal{C} \subseteq \{0,1\}^*$.
Given the set $\mathcal{Y}$, we write $y \leftarrow \mathcal{Y}$ to denote the uniformly random selection of y in $\mathcal{Y}$.
We reserve sans serif (Alg) notations for algorithms. If the algorithm Alg is probabilistic, we can think of its output as a distribution. We denote with $a \leftarrow \mathsf{Alg}(b, c, d)$ the fact that we sample from the distribution induced by algorithm Alg on inputs (b, c, d), and we obtain a. We may write part of the arguments of the algorithm as subscripts or superscripts, that is, $\mathsf{Alg}_b^c(d) = \mathsf{Alg}_b(c, d) = \mathsf{Alg}(b, c, d)$.
A (q, t)-adversary A is a probabilistic algorithm which can make at most q queries to the oracle(s) he is granted access to, and runs in time bounded by t.
Let algorithm Alg be an algorithm whose inputs are in $\mathcal{S}^1 \times \cdots \times \mathcal{S}^n$ and whose output is in $\mathcal{Y}$. We say that algorithm Alg *does not reveal, via the length of its output, any information about its inputs apart from their lengths* if there exists a deterministic function $f : \mathbb{N}^n \longmapsto \mathbb{N}$ s.t. $|y| = f(|s_1|, ..., |s_n|)$ for all possible inputs $(s_1, ..., s_n)$. We assume that all the Enc and AEnc algorithms we use have this property.
Given a game, where the adversary A is allowed to query many oracles, we use a single counter for all the queries made by adversary A, during the game. The oracle $\perp(\cdot,\cdot)$ always answers $\perp$. When an adversary is playing a game where he has access either to an oracle implemented with algorithm $\mathsf{Alg}(\cdot,\cdot)$ or the oracle $\$(\cdot,\cdot)$ it means that the oracle $\$(\cdot,\cdot)$ answers a random bit string of length $|\mathsf{Alg}(\cdot,\cdot)|$. Moreover, it keeps in memory the answers it gives.
We denote with $c^i \leftarrow \mathsf{O}(a^i, b^i)$ the output c^i of the i-th query on oracle O with input (a^i, b^i). Usually we use only one counter for all the queries, that is the i query can be to oracle O_j and the $i+1$th to oracle O_l where oracle O_j and O_l are among the oracles the adversary is granted access. It can be that that the two oracle are the same, that is $\mathsf{O}_j = \mathsf{O}_l$.
We write $\Pr[B; A_1, A_2, ...]$ for the probability of event B after the experiment described by steps $A_1, A_2, ...$.

In the rest of this section we provide many standard definitions. The expert reader may skip all this section, except Def. 4, which is non-standard.

2.2 Pseudorandom functions (PRF)

We now define the PRF-security notion, the base of many cryptographic primitives:

Definition 1. *A function* $\mathsf{F} : \mathcal{K} \times \mathcal{M} \longmapsto \mathcal{T}$ *is a* (q, t, ϵ)-pseudorandom function (PRF) *if for every* (q, t) *adversary* A*, the advantage :*

$$\mathsf{Adv}^{PRF}_{\mathsf{F}}(\mathsf{A}) := |\Pr[\,\mathsf{A}^{\mathsf{F}_k(\cdot)} \Rightarrow 1\,] - \Pr[\,\mathsf{A}^{\mathsf{f}(\cdot)} \Rightarrow 1\,]|$$

is upper bounded by ϵ *where* k *and* f *are chosen uniformly at random from their domains, namely* $\mathcal{K}$ *and the set of functions from* $\mathcal{M}$ *to* $\mathcal{T}$*,* $\mathcal{FUNC}(\mathcal{M}, \mathcal{T})$*.*
In a similar way, F *is a* pseudorandom permutation *(PRP) if* F_k *is a permutation and the above advantage is* ϵ *bounded when* f *is selected as a random permutation.*

We remind that a PRP is a PRF (see Proposition 3.27 [14]).

In some of our constructions, we will also use tweakable pseudorandom permutations [18]. They are PRPs with an additional input, the tweak: $\mathsf{E} : \mathcal{K} \times \mathcal{TW} \times \mathcal{M} \longmapsto \mathcal{T}$, and their security advantage is then defined as $\mathsf{Adv}^{\mathrm{TPRP}}_{\mathsf{E}}(\mathsf{A}) := \mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{F}}(\mathsf{A})$ where $\mathsf{F}(k, (tw, m)) := \mathsf{E}(k, tw, m)$ and for any choice of k and tw $\mathsf{E}_k(tw, \cdot)$ is a permutation.

2.3 Nonce-based Authenticated Encryption (nAE) and Encryption (nE and ivE) schemes

For the syntax of encryption schemes we follow the approach of Namprempre et al. [20] (taken by the work of Rogaway [26]) where the encryption algorithm is deterministic and an "*initialization vector*"(IV) iv is surfaced (and it may be seen as part of the AD [27]). Using this approach we classify encryption schemes according to the requirements of this extra input to provide CPA-security.

Definition 2 ([20]). *A scheme for* nonce-based authenticated encryption *(*nAE*) is a triple* $\boldsymbol{\Pi} := (\mathcal{K}, \mathsf{AEnc}, \mathsf{ADec})$*, where the keyspace* $\mathcal{K}$ *is a nonempty set, the encryption algorithm* AEnc *is a deterministic algorithm which takes as input the tuple* $(k, n, a, m) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$ *and outputs a string* $c \leftarrow \mathsf{AEnc}^{n,a}_k(m)$ *called ciphertext.*
The decryption algorithm ADec *is a deterministic algorithm which takes as input the tuple* $(k, n, a, c) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C}$ *and outputs* $m \leftarrow \mathsf{ADec}^{n,a}_k(c)$ *which is either a string* $m \in \mathcal{M}$ *or the symbol* $\perp$ *("invalid").*
We require that the algorithms AEnc *and* ADec *are the inverse of each other, that is:*

- (Correctness) *if* $\mathsf{AEnc}^{n,a}_k(m) = c$ *then* $\mathsf{ADec}^{n,a}_k(c) = m$
- (Tidiness) *if* $\mathsf{ADec}^{n,a}_k(c) = m \neq \perp$ *then* $\mathsf{AEnc}^{n,a}_k(m) = c$

If $\mathsf{ADec}^{n,a}_k(c) = \perp$ *we say that the algorithm* rejects c*, otherwise it* accepts c*.*
A sloppy nAE *scheme satisfies everything but the tidiness condition.*
A nonce-based Encryption scheme (nE) *is a triple* $\Pi = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$*, where*

Enc *and* Dec *do not take input the AD, that is,* $\mathsf{Enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \longmapsto \mathcal{C}$ *and* $\mathsf{Dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{C} \longmapsto \mathcal{M}$.
*An iv-*based encryption scheme ivE *is synctactically equivalent to a* nE *scheme, with the only difference that the nonce space* $\mathcal{N}$ *is replaced with an IV space* $\mathcal{IV}$.

Tidiness, as correctness, is usually seen as a syntactic requirement (for example Namprempre et al., [20]). Instead, in this paper, we show an explicit case where this property is fundamental to provide security (see Section 7.1).
Paterson et al. [23] defined the "collision-resistant decryption" (CRD), which is a security property. Tidy schemes are inherently CRD-secure, since there is one and only valid ciphertext for each input, but the converse is not valid (because CRD-security is obtained when adversaries are able to break it with negligible probability, while tidiness always works).

The difference between nE schemes and ivE schemes lies in their security requirements. A complete survey about nAE, nE and ivE schemes can be found in the extended version.

2.4 Security for nAE, nE and ivE schemes

The security definitions for nAE, nE and ivE schemes are inspired from those in [20] and [27].

Definition 3. *A nonce-based authenticated encryption scheme* (nAE) $\boldsymbol{\Pi} := (\mathcal{K}, \mathsf{AEnc}, \mathsf{ADec})$ *is* (q, t, ϵ)-nAE-secure *if the advantage*

$$\mathsf{Adv}^{\mathsf{nAE}}_{\boldsymbol{\Pi}}(\mathsf{A}) := \left|\Pr\left[\mathsf{A}^{\mathsf{AEnc}_k(\cdot,\cdot,\cdot),\mathsf{ADec}_k(\cdot,\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathsf{A}^{\$(\cdot,\cdot,\cdot),\perp(\cdot,\cdot,\cdot)} \Rightarrow 1\right]\right| \quad (1)$$

is bounded by ϵ *for every* (q, t)*-adversary* A *that respects the following two conditions: (i) If* A *queried the first (encryption) oracle on input* (n, a, m) *and was answered* c*, then he is not allowed to query the second (decryption) oracle on input* (n, a, c)*. (ii)* A *is not allowed to repeat the first component (the nonce) on different left oracle queries.*

$\boldsymbol{\Pi}$ *is* (q, t, ϵ)-nAE-E *secure, if the advantage*

$$\mathsf{Adv}^{\mathsf{nAE\text{-}E}}_{\boldsymbol{\Pi}}(\mathsf{A}) := \left|\Pr\left[\mathsf{A}^{\mathsf{AEnc}_k(\cdot,\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathsf{A}^{\$(\cdot,\cdot,\cdot)} \Rightarrow 1\right]\right| \quad (2)$$

is bounded by ϵ *for every* (q, t)*-adversary* A *that respects Condition (ii) above.*

A nonce-based encryption scheme (nE) $\Pi := (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ *is* (q, t, ϵ)-nE-secure *if the advantage,*

$$\mathsf{Adv}^{\mathsf{nE}}_{\boldsymbol{\Pi}}(\mathsf{A}) := \left|\Pr\left[\mathsf{A}^{\mathsf{Enc}_k(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathsf{A}^{\$(\cdot,\cdot)} \Rightarrow 1\right]\right| \quad (3)$$

is bounded by ϵ *for every* (q, t)*-adversary* A *that respects Condition (ii) above.*

An iv-based encryption scheme ivE $\Pi := (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ *is* (q, t, ϵ)-ivE-secure *if the advantage*

$$\mathsf{Adv}^{\mathsf{ivE}}_{\Pi}(\mathsf{A}) := \left|\Pr\left[\mathsf{A}^{\mathsf{Enc}^{\$}_k(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathsf{A}^{\$(\cdot)} \Rightarrow 1\right]\right| \quad (4)$$

is bounded by ϵ for every (q,t)-adversary. Here the oracle $\mathsf{Enc}^{\$}(m)$ picks a random $iv \leftarrow \mathcal{IV}$, then computes $c \leftarrow \mathsf{Enc}_k(iv,m)$ and returns (iv,c).

As a result of this definition, the only difference between ivE and nE security is the requirement on their auxiliary input: non-repeating nonces for nE and random *iv*s for ivE. We observe that ivE-security implies nE security when uniformly random *iv*s are expected to differ with overwhelming probability. The contrary does not hold: the CTR mode is well-known illustration (details are provided in the extended version).

In some cases, it is desirable to guarantee some security even if nonces are repeated: this is called resistance to nonce misuse, or simply misuse resistance.

Definition 4. *If we drop Condition (ii) on the non repetition of the nonces in the* nE *security definitions, then we augment the security notions with* misuse resistance. *Namely, we say that the* nE *scheme is (q,t,ϵ)-*mrE *secure.*

We point out that in the mrE definition the adversary has only access to an encryption oracle, differently from the standard *misuse resistance for authenticated encryption* mrAE [27]. An example of an nE scheme which is mrE and not mrAE is given in the extended version as well as many examples [24,8].
The mrE definition is trivially extended to ivE schemes, since the syntax of nE schemes and ivE schemes is identical.

2.5 Chosen-plaintext attack security with chosen nonce

We define mCPA security for nE and ivE schemes, following the left-or-right definition of Bellare et al. [2], but adapted to the nonce-based setting. This definition is a multi-challenge definition, contrary to the common single-challenge variant [14]. The two versions of the definition are asymptotically equivalent, but come with a difference of a linear factor when quantitative bounds are used, as we do here.

Definition 5. *A nonce-based Encryption scheme* nE *$\Pi = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ is (q,t,ϵ)-*mCPA *secure, or (q,t,ϵ)-secure against chosen plaintext attacks for multiple encryptions, if:*

$$\mathsf{Adv}^{\mathsf{mCPA}}_{\Pi}(\mathsf{A}) := \left| \frac{1}{2} - \Pr\left[b' = b; b \leftarrow \{0,1\},\ b' \leftarrow \mathsf{A}^{\mathsf{Enc}^b_k(\cdot,\cdot,\cdot)} \right] \right|$$

is bounded by ϵ for any (q,t)-adversary. Here the oracle $\mathsf{Enc}^b_k(\cdot,\cdot,\cdot)$ is an oracle, which on input $(n,m_0,m_1) \in \mathcal{N} \times \mathcal{M}^2$ outputs $c \leftarrow \mathsf{Enc}_k(n,m_b)$ for a random secret bit $b \leftarrow \{0,1\}$, which the oracle has picked at the start of the game. When the adversary A *queries $\mathsf{Enc}^b_k(\cdot,\cdot,\cdot)$, he must choose two messages m_0 and m_1 s.t. $|m_0| = |m_1|$. Moreover he cannot repeat the first input (the nonce) in different queries.*

There is a completely similar definition for ivE schemes. We only have to replace $\mathsf{Enc}^b_k(\cdot,\cdot,\cdot)$ with $\mathsf{Enc}^{b,\$}_k(\cdot,\cdot)$, and to adapt the $\$(\cdot,\cdot,\cdot)$ oracle accordingly.

Similarly there is a similar notions for nAE schemes, obtained from the previous one by replacing $\mathsf{Enc}_k(\cdot,\cdot)$ with $\mathsf{AEnc}_k(\cdot,\cdot,\cdot)$ and adapting $\$(\cdot,\cdot,\cdot)$ accordingly.

2.6 Authenticity (INT-CTXT)

Following Bellare et al. [5], we focus on the notion of ciphertext integrity with a single decryption query.

Definition 6. *A nonce-based authenticated encryption scheme* nAE $\boldsymbol{\Pi} = (\mathcal{K},$ AEnc, ADec) *is* (q,t,ϵ)-INT-CTXT1(Ciphertext integrity with only 1 decryption query)-secure *if*

$$\mathsf{Adv}^{\text{INT-CTXT1}}_{\boldsymbol{\Pi}}(\mathsf{A}) := \Pr\left[\perp \neq m^* \leftarrow \mathsf{ADec}_k(n^*,a^*,c^*);\ (n^*,a^*,c^*) \leftarrow \mathsf{A}^{\mathsf{AEnc}_k(\cdot,\cdot,\cdot)}\right]$$

is bounded by ϵ *for every* (q,t) *adversary. The adversary* A *is not allowed to repeat the first component (the nonce) on different oracle queries. Moreover he is not allowed to output* (n^*,a^*,c^*) *if he received* c^* *as* $c^* \leftarrow \mathsf{AEnc}_k(n^*,a^*,m^*)$ *for a certain input* (n^*,a^*,m^*) *that he asked to the first oracle.*

As we can expect, an nAE scheme that offers both mCPA and INT-CTXT1 security is an nAE scheme, (and we prove this the extended version).

2.7 Message Authentication Code (MAC)

Apart from an encryption scheme, all our composition modes are based on a deterministic notion of Message Authentication Code (MAC).

Definition 7. *A* Message Authentication Code MAC *is a triple* $\Pi = (\mathcal{K}, \mathsf{Mac}, \mathsf{Vrfy})$ *where the keyspace* $\mathcal{K}$ *is a non-empty set, the tag-generation algorithm* Mac *is a deterministic algorithm that takes as input the couple* $(k,m) \in \mathcal{K} \times \mathcal{M}$ *and outputs the tag* $\tau \leftarrow \mathsf{Mac}_k(m)$ *from the tag space* $\mathcal{T}$*. The verification algorithm* Vrfy *takes as input a triple* (k,m,τ) *in* $\mathcal{K} \times \mathcal{M} \times \mathcal{T}$ *and outputs* $\top$ *(accept) or* $\perp$ *(reject). We ask that* $\mathsf{Vrfy}(k,m,\mathsf{Mac}(k,m)) = \top$.
A string-input MAC strMAC *has as input space a set of strings, that is* $\mathcal{M} \subseteq \{0,1\}^*$.
A vector-input MAC vecMAC *has as input space* $\mathcal{M}$ *which has one or more component and it can accept tuples of strings as input.*

Usually the security for MACs is expressed as unforgeability, but our composition modes rely on a Mac function that is a $(q,t,\epsilon_{\text{PRF}}) - \text{PRF}$.

Definition 8. *([20]) A* MAC $\Pi = (\mathcal{K}, \mathsf{Mac}, \mathsf{Vrfy})$ *is* $(q,t,\epsilon) - \text{PRF}$*-secure if*

$$\mathsf{Adv}^{PRF}_{\Pi}(\mathsf{A}) := \mathsf{Adv}^{PRF}_{\mathsf{Mac}}(\mathsf{B})$$

is bounded by ϵ *for any* (q,t) *adversary* B *and if* $\mathsf{Vrfy}(k,m,\mathsf{Mac}(k,m)) = \top$ *iff* $\tau = \mathsf{Mac}(k,m)$.

For completeness, the standard definitions are put in the extended version.

3 Problem

As discussed earlier, Namprempre et al. [20] left open the problem of the nAE security of 4 modes based on the Tag-then-Encrypt paradigm, which have been shown in Fig. 1.

Formally, the first three modes compose an ivE scheme $\Pi = (\mathcal{K}_E, \mathsf{Enc}, \mathsf{Dec})$ and two vecMAC schemes using the same key, $\mathsf{MAC}^{\mathsf{IV}} = (\mathcal{K}_M, \mathsf{Mac}^{\mathsf{IV}}, \mathsf{Vrfy}^{\mathsf{IV}})$ and $\mathsf{MAC}^{\mathsf{Tag}} = (\mathcal{K}_M, \mathsf{Mac}^{\mathsf{Tag}}, \mathsf{Vrfy}^{\mathsf{Tag}})$ in this way:

- A10: $\mathsf{AEnc}^{n,a}_{k_E,k_M}(m) := c$ with $iv = \mathsf{Mac}^{\mathsf{IV}}_{k_M}(n,a)$, $\tau = \mathsf{Mac}^{\mathsf{Tag}}_{k_M}(a,m)$ and $c = \mathsf{Enc}_{k_E}(iv, m||\tau)$
- A11: $\mathsf{AEnc}^{n,a}_{k_E,k_M}(m) := c$ with $iv = \mathsf{Mac}^{\mathsf{IV}}_{k_M}(n,a)$, $\tau = \mathsf{Mac}^{\mathsf{Tag}}_{k_M}(m)$ and $c = \mathsf{Enc}_{k_E}(iv, m||\tau)$
- A12: $\mathsf{AEnc}^{n,a}_{k_E,k_M}(m) := c$ with $iv = \mathsf{Mac}^{\mathsf{IV}}_{k_M}(n)$, $\tau = \mathsf{Mac}^{\mathsf{Tag}}_{k_M}(a,m)$ and $c = \mathsf{Enc}_{k_E}(iv, m||\tau)$

The fourth mode composes a nE Encryption scheme $\Pi = (\mathcal{K}_E, \mathsf{Enc}, \mathsf{Dec})$ and a vecMAC $= \mathsf{MAC} = (\mathcal{K}_M, \mathsf{Mac}, \mathsf{Vrfy})$:

- N4: $\mathsf{AEnc}^{n,a}_{k_E,k_M}(m) := c$ with $\tau = \mathsf{Mac}^{\mathsf{Tag}}_{k_M}(a,m)$ and $c = \mathsf{Enc}_{k_E}(n, m||\tau)$

For clarity we reserve bold notations $\mathbf{m}$ for the messages inputs of the nAE scheme $\boldsymbol{\Pi}$ and normal notations m for the messages inputs to the underlying nE (or ivE)-scheme Π (so, we typically have that $m = \mathbf{m}\|\tau$).

If Π is tidy and the MAC is PRF-secure, then the AE scheme $\boldsymbol{\Pi}$, obtained composing these components, is tidy. These modes also offer CPA security [20], which directly results from the underlying encryption schemes (a quantitative proof of this statement is available in the extended version [9]).

As a result, the open question lies in the INT-CTXT security of these modes.

4 Attack against mode N4

We provide here an attack against the mode N4, explicitly presenting an nAE-scheme $\boldsymbol{\Pi}$, based on an nE Encryption scheme $\Pi = (\mathcal{K}_E, \mathsf{Enc}, \mathsf{Dec})$ and a vecMAC $\mathsf{MAC} = (\mathcal{K}_M, \mathsf{Mac}, \mathsf{Vrfy})$ which is PRF-secure. For simplicity, we consider only the case when the message $\mathbf{m}$ of $\boldsymbol{\Pi}$ is λ-bit long and the tag is λ-bit long, leaving the general case to the extended version. The nE Encryption scheme, which encrypts 2λ-bit long message, is nE-secure and tidy, but the nAE-scheme $\boldsymbol{\Pi}$ obtained composing them according to mode N4, is not secure and, in particular, it is not INT-CTXT1-secure as we show a forgery.
The idea of the forgery is to force the tag τ of a couple $(a, \mathbf{m})$ to be encrypted identically for two different nonces, while keeping the nE-security.

4.1 Construction

Following the definition of mode $N4$, an authenticated ciphertext is computed as $c = \mathsf{Enc}_{k_E}(n, \mathbf{m}||\mathsf{Mac}_{k_M}(a, \mathbf{m}))$, for which Mac is a PRF. We now define the nE scheme Π.

The keys produced in Π are made of two components (k, v^*): the key $k \in \mathcal{K}$ of a TPRP E and a random value v^*, which has the size of block of E, that is, λ bits. This value v^* will be leaked to Adv when asking for the encryption of a message with the nonce $n = 1$, and will then be used to trigger a kind of Trojan in the encryption scheme. That Trojan will have the following behavior: for nonces $n = 1, 2$, and if the first message block is v^*, then the last ciphertext block will be computed in a way that ignores the value of n.

This behavior is benign when considering the nE security of Π: the only way to observe it would be to make two encryption queries with nonces 1 and 2, and first message block v^*. But doing this would require guessing v^* before querying with nonce 1 (the nE adversary is nonce respecting), and this cannot be done but with probability $2^{-\lambda}$: it would require guessing v^*.

As we will see, it is not benign anymore when considering the ciphertext integrity property: there, Adv is free to use the nonces 1 and 2 in its decryption query, even if these nonces were used in encryption queries.

To make things concrete, we define the encryption process Enc of Π using a TPRP $\mathsf{E} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^\lambda \longmapsto \mathcal{T} = \{0,1\}^\lambda$ with tweak space $\mathcal{TW} = \{0,1,2\} \times \{0,1\}$. For a message $m = (m_1, m_2) \in \{0,1\}^{2\lambda}$, the ciphertext $\mathsf{Enc}_k^n(m)$ is made of three blocks (c_0, c_1, c_2) computed as follows:

- $c_0 = \mathsf{E}_k^{(0,0)}(n)$ unless $n = 1$, in which case $c_0 := v^*$.
- $c_1 = m_1 \oplus \mathsf{E}_k^{(1,0)}(n)$.
- $c_2 = m_2 \oplus \mathsf{E}_k^{(2,0)}(n)$, unless the condition $[(n = 1 \vee n = 2) \wedge m_1 = v^*]$ is met, in which case $c_2 = m_2 \oplus \mathsf{E}_k^{(2,1)}(0)$.

With such a definition, the block c_0 looks random for any input, and its only purpose is to leak v^* when $n = 1$. The block c_1 is a traditional encryption of the message block m_1 using the TPRP E. The block c_2 is computed in the same way (just incrementing the tweak), except under a very specific condition: the nonce is either 1 or 2, and the first message block $m_1 = v^*$. Under that condition, the ciphertext block becomes independent of the nonce. As explained above, this condition is designed in such a way that it cannot lead to any observable event when Adv can only access an encryption oracle in a nonce respecting way: that would require querying Enc on a message starting with v^* on both $n = 1$ and $n = 2$, but v^* is only learned after a query with $n = 1$, and it is then not permitted to make a second query with $n = 1$ and v^* as message block.

The decryption of Π works in the natural way. In particular, in order to guarantee the tidiness of the nE encryption scheme, Dec must verify the correctness of the first ciphertext block c_0.

The proofs that Π is nE-secure can be found in the extended version.

4.2 Forgery

The composition of the previous nE scheme Π with a PRF-secure MAC according to mode N4 is not $\mathsf{INT\text{-}CTXT}1$-secure. In fact, we provide a forgery where the adversary A asks the encryption of only two messages:

1. It first asks for an encryption of $(1, a, \mathbf{m})$, for arbitrary choices of a and $\mathbf{m}$. This returns a ciphertext whose first block is v^*, second block is $c_1 = \mathbf{m} \oplus \mathsf{E}_k^{(1,0)}(1)$, and third block is ignored.
2. It then asks for an encryption of $(2, a, v^*)$. This returns a ciphertext whose last block is $c_2 = \mathsf{Mac}_{k_M}(a, v^*) \oplus \mathsf{E}_k^{(2,1)}(0)$.

Eventually, Adv makes a decryption query on $(1, a, (v^*, c_1 \oplus \mathbf{m} \oplus v^*, c_2))$, which is different of the two previously obtained ciphertexts, and has a valid decryption to v^*, hence violating the ciphertext integrity property.

This shows that N4 is not a secure composition mode, in general.

Ciphertext extension We observe that the encryption scheme we use to break N4 uses the ciphertext expansion, that is the ciphertext is bigger than the plaintext (i.e. $|c| < |m|$ with $c = \mathsf{Enc}_k(m)$). It is left the question whether if the encryption scheme does not have the ciphertext expansion, scheme N4 is secure.

5 Security relations among A10, A11 and A12

While we are able to prove the generic insecurity of N4, we are not able to prove that modes A10,A11 and A12 are either secure or insecure in general. Still, in this section, we prove that these three modes are either all secure or all insecure.

To prove it we need to replace the two $\mathsf{vecMACs}$ $\mathsf{vecMAC}^{\mathsf{IV}}$ and $\mathsf{vecMAC}^{\mathsf{Tag}}$ with two $\overline{\mathsf{vecMACs}}$ based on the random functions f^{IV} and $\mathsf{f}^{\mathsf{Tag}}$. Now the key of the new nAE scheme is $k := (k_E, \mathsf{f}^{\mathsf{IV}}, \mathsf{f}^{\mathsf{Tag}})$. To highlight these changes, we call the new modes $\overline{\mathrm{A10}}, \overline{\mathrm{A11}}$ and $\overline{\mathrm{A12}}$ and the new nAE-schemes $\overline{\boldsymbol{\Pi}}$. The security relations among modes $\overline{\mathrm{A10}}, \overline{\mathrm{A11}}$ and $\overline{\mathrm{A12}}$ immediately lift to modes A10,A11 and A12. The standard details (replacing $\mathsf{Mac}^{\mathsf{IV}}$ and $\mathsf{Mac}^{\mathsf{Tag}}$ with two random functions) are discussed in the extended version, where we prove that if $\overline{\mathrm{A}i}$ is $(q, t, \epsilon_{\mathsf{INT\text{-}CTXT}})$-$\mathsf{INT\text{-}CTXT}$ secure then Ai is $(q, t, \epsilon_{\mathsf{INT\text{-}CTXT}} + \epsilon_{\mathrm{PRF}})$-$\mathsf{INT\text{-}CTXT}$ secure provided that $\mathsf{Mac}^{\mathsf{IV}}$ and $\mathsf{Mac}^{\mathsf{Tag}}$ are $(q, t, \epsilon_{\mathrm{PRF}})$-PRF-secure.

We show the security equivalence of A10, A11 and A12 based on two events, B and C, that we define below. Consider a $\mathsf{INT\text{-}CTXT1}$ adversary A against an nAE scheme $\overline{\boldsymbol{\Pi}}$ (which is made according to any of $\overline{\mathrm{A10}}$, $\overline{\mathrm{A11}}$ or $\overline{\mathrm{A12}}$). If the q-th decryption query (n^q, a^q, c^q) is valid, then $c^q = \overline{\mathsf{AEnc}_k}(n^q, a^q, \mathbf{m}^q)$ for a certain message $\mathbf{m}^q$, as a result of tidiness. Depending on the value of (n^q, a^q) (or only n^q for $\overline{A12}$), we distinguish between two possibilities, which define event B:

- (n^q, a^q) is fresh, that is, $(n^q, a^q) \neq (n^j, a^j)\ \forall j = 1, ..., q-1$ (we call this event B) [for mode $\overline{\mathrm{A12}}$, we only demand that n^q is fresh, that is $n^q \neq n^j\ \forall j = 1, ..., q-1$].
- $(n^q, a^q) = (n^j, a^j)$ for a $j \in \{1, ..., q-1\}$ (This j is unique since the nonce n cannot be repeated) [for mode $\overline{\mathrm{A12}}$, we only demand that $n^q = n^j$ for a $j \in \{1, ..., q-1\}$].

With regard to $(a^q, \mathbf{m}^q)$ (or only $\mathbf{m}^q$ for mode $\overline{A11}$), we again consider two possibilities, which define event C:

– $(a^q, \mathbf{m}^q)$ is fresh, that is $(a^q, \mathbf{m}^q) \neq (a^j, \mathbf{m}^j)\ \forall j = 1, ..., q-1$ (we call this event C) [for mode $\overline{\text{A11}}$, we only demand that m^q is fresh, that is $\mathbf{m}^q \neq \mathbf{m}^j\ \forall j = 1, ..., q-1$].
– $(a^q, \mathbf{m}^q) = (a^j, \mathbf{m}^j)$ for some $j \in \{1, ..., q-1\}$ (there can be several such j's) [for mode $\overline{\text{A11}}$, we only demand $\mathbf{m}^q = \mathbf{m}^j$ for some $j \in \{1, ..., q-1\}$].

Clearly by total law of probability

$$\Pr[\mathsf{A}\text{ wins}] = \Pr[\mathsf{A}\text{ wins } \cap C] + \Pr[\mathsf{A}\text{ wins } \cap B \cap \overline{C}] + \Pr[\mathsf{A}\text{ wins } \cap \overline{B} \cap \overline{C}]$$

With the following lemma we treat the first two addends of the previous equation:

Lemma 1. *Let* $\mathsf{f}^{\mathsf{IV}} : \mathcal{N} \times \mathcal{A} \longmapsto \mathcal{IV}$ *[for mode* $\overline{A12}$*,* $\mathsf{f}^{\mathsf{IV}} : \mathcal{N} \longmapsto \mathcal{IV}$ *] and* $\mathsf{f}^{\mathsf{Tag}} : \mathcal{A} \times \mathcal{M} \longmapsto \mathcal{T}$ *[for mode* $\overline{A11}$*,* $\mathsf{f}^{\mathsf{Tag}} : \mathcal{M} \longmapsto \mathcal{T}$ *] be two random functions and let* $\Pi = (\mathcal{K}_E, \mathsf{Enc}, \mathsf{Dec})$ *be a* $(q, t, \epsilon_{\mathsf{ivE}})$-$\mathsf{ivE}$*-secure encryption scheme. Let* $\overline{\boldsymbol{\Pi}}$ *be the* nAE *scheme obtained composing* f^{IV}*,* $\mathsf{f}^{\mathsf{Tag}}$ *and* Π *according to mode* $\overline{A10}$ *or* $\overline{A11}$ *or* $\overline{A12}$*. Then we can bound*

$$\Pr[\mathsf{A}\ \textit{wins}\ \cap C] + \Pr[\mathsf{A}\ \textit{wins}\ \cap B \cap \overline{C}] \leq q|\mathcal{T}|^{-1} + (q-1)\epsilon_{\mathsf{ivE}}$$

The proof is completely standard and can be found in the extended version (the ideas of this proof are already present in Namprempre et al. [20]. The proofs of the security implications between the 3 "A" modes then results from implications in the case A wins $\cap \overline{B} \cap \overline{C}$, which we examine in the rest of this section.

In order to make our notations more precise, if either f^{IV} or $\mathsf{f}^{\mathsf{Tag}}$ have different signatures for two modes that we compare, we use a subscript to denote the mode that is used (e.g. $\mathsf{f}^{\mathsf{IV}_{10}}$ for mode A10).
In some proves we use hash function and their collision resistance, for more details about this see Katz and Lindell [14].

5.1 The INT-CTXT1-security of $\overline{\text{A12}}$ implies the INT-CTXT1-security of $\overline{\text{A10}}$

Proposition 1. *Let* $\mathsf{f}^{\mathsf{IV}_{10}} : \mathcal{N} \times \mathcal{A} \longmapsto \mathcal{IV}$ *and* $\mathsf{f}^{\mathsf{Tag}} : \mathcal{A} \times \mathcal{M} \longmapsto \mathcal{T}$ *be two random functions and let* $\Pi = (\mathcal{K}_E, \mathsf{Enc}, \mathsf{Dec})$ *be a* $(q, t, \epsilon_{\mathsf{ivE}})$-$\mathsf{ivE}$*-secure encryption scheme. Then, if mode* $\overline{A12}$ *implemented with the random function* $\mathsf{f}^{\mathsf{IV}_{12}} : \mathcal{N} \longmapsto \mathcal{IV}$ *is* $(q-1, t, \epsilon_{\mathsf{INT\text{-}CTXT1}})$-$\mathsf{INT\text{-}CTXT1}$*-secure then mode* $\overline{A10}$ *is* $(q-1, t, q|\mathcal{T}|^{-1} + (q-1)\epsilon_{\mathsf{ivE}} + \epsilon_{\mathsf{INT\text{-}CTXT1}})$-$\mathsf{INT\text{-}CTXT1}$*-secure*

Let $\mathsf{f}'^{\mathsf{IV}_{12}} : \mathcal{N} \times \mathcal{A} \longmapsto \mathcal{IV}$ be defined $\mathsf{f}'^{\mathsf{IV}_{12}}(n, a) := \mathsf{f}^{\mathsf{IV}_{12}}(n)\ \forall n \in \mathcal{N}, a \in \mathcal{A}$ (it is an extension of $\mathsf{f}^{\mathsf{IV}_{12}}$). The proof is based on the fact that it is impossible using only encryption queries to mode $\overline{\text{A10}}$ to distinguish if it is used $\mathsf{f}^{\mathsf{IV}_{10}}$ or $\mathsf{f}'^{\mathsf{IV}_{12}}$ (as in mode $\overline{\text{A12}}$), since it is not possible for the adversary A to force the nAE algorithm to call $\mathsf{f}^{\mathsf{IV}_{10}}$ on inputs (n, a_1) and (n, a_2) (with $a_1 \neq a_2$) during encryption queries. Moreover, the couple (n^q, a^q) of the decryption query must not be fresh (due to event $\overline{B}$), thus, using $\mathsf{f}'^{\mathsf{IV}_{12}}$ is indistinguishable from using $\mathsf{f}^{\mathsf{IV}_{10}}$.

5.2 The INT-CTXT1-security of $\overline{\text{A11}}$ implies the INT-CTXT1-security of $\overline{\text{A10}}$

Proposition 2. *Let* $\mathsf{f}^{\mathsf{IV}} : \mathcal{N} \times \mathcal{A} \longmapsto \mathcal{IV}$ *and* $\mathsf{f}^{\mathsf{Tag}_{10}} : \mathcal{A} \times \mathcal{M} \longmapsto \mathcal{T}$ *be two random functions and let* $\Pi = (\mathcal{K}_E, \mathsf{Enc}, \mathsf{Dec})$ *be a* $(q, t, \epsilon_{\mathsf{ivE}})$-ivE*-secure encryption scheme. Let* $\mathsf{H} : \mathcal{A} \longmapsto \{0,1\}^N$ *be a* $(0, t, \epsilon_{cr})$ *collision resistant hash function. Then, if mode* $\overline{A11}$*, implemented with the random function* $\mathsf{f}^{\mathsf{Tag}_{11}} : \mathcal{M} \longmapsto \mathcal{T}$ *and with any* $(q, t, \epsilon_{\mathsf{ivE}} + \frac{q^2}{2|\mathcal{IV}|}) -$ ivE*-secure Encryption scheme, is* $(q-1, t, \epsilon_{\text{INT-CTXT1}})$-INT-CTXT1*-secure then mode* $\overline{A10}$ *is* $(q-1, t, \epsilon)$-INT-CTXT1*-secure, where*

$$\epsilon = q|\mathcal{T}|^{-1} + (q-1)\epsilon_{\mathsf{ivE}} + \epsilon_{cr} + \epsilon_{\text{INT-CTXT1}}.$$

The idea is to reduce the INT-CTXT1 adversary A against scheme $\overline{\boldsymbol{\Pi}}$ (mode $\overline{\text{A10}}$), which uses the ivE scheme Π, to a INT-CTXT1 adversary C against scheme $\overline{\boldsymbol{\Pi}'}$ (mode $\overline{\text{A11}}$), which uses the ivE scheme Π'. When the adversary A makes an encryption query $(n^i, a^i, \mathbf{m}^i)$ the adversary C makes an encryption query $(n^i, a^i, \mathbf{m'}^i)$ with $\mathbf{m'}^i = \mathsf{H}(a^i)\|\mathbf{m}^i$. The ivE scheme Π' encrypts $m'^i = (\mathsf{H}(a^i)\|m^i[= \mathbf{m}^i\|\tau^i])$ in this way: $\mathsf{Enc}'(m'^i) := \mathsf{H}(a^i) \oplus \mathsf{f}^{\mathsf{Enc}}(iv^i)\|\mathsf{Enc}(iv^i, m^i)$, where $\mathsf{f}^{\mathsf{Enc}}$ is a random function (and it is part of the key the scheme Π'). When the adversary A makes his decryption query (n^q, a^q, c^q) the adversary C simply asks the decryption of $(n^q, a^q, \left[\mathsf{f}^{\mathsf{Enc}}(iv^q) \oplus \mathsf{H}(a^q)\right] \|c^q)$ (the iv^q must be not fresh due to event $\overline{B}$).

5.3 The INT-CTXT-security of $\overline{\text{A10}}$ implies the INT-CTXT-security of $\overline{\text{A12}}$

Proposition 3. *Let* $\mathsf{f}^{\mathsf{IV}_{12}} : \mathcal{N} \longmapsto \mathcal{IV}$ *and* $\mathsf{f}^{\mathsf{Tag}} : \mathcal{A} \times \mathcal{M} \longmapsto \mathcal{T}$ *be two random functions and let* $\Pi = (\mathcal{K}_E, \mathsf{Enc}, \mathsf{Dec})$ *be a* $(q, t, \epsilon_{\mathsf{ivE}})$-ivE*-secure encryption scheme. Let* $\overline{\boldsymbol{\Pi}}$ *be the* nAE*-scheme obtained composing these components according to mode* $\overline{A12}$*. Let* $\mathsf{H} : \mathcal{A} \longmapsto \{0,1\}^N$ *be* $(0, t, \epsilon_{cr})$*.*
Then, if mode $\overline{A10}$*, implemented with the random function* $\mathsf{f}^{\mathsf{IV}_{10}} : \mathcal{N} \times \mathcal{A} \longmapsto \mathcal{IV}$ *and with any* $(q, t, \epsilon_{\mathsf{ivE}} + \frac{q^2}{2|\mathcal{IV}|}) -$ ivE*-secure Encryption scheme, is* $(q, t, \epsilon_{\text{INT-CTXT1}})$-INT-CTXT1*-secure then mode* $\overline{A12}$ *is* $(q-1, t, \epsilon)$-INT-CTXT1*-secure with*

$$\epsilon = q|\mathcal{T}|^{-1} + (q-1)\epsilon_{\mathsf{ivE}} + \epsilon_{cr} + \epsilon_{\text{INT-CTXT1}}.$$

The idea of the proof is similar to the previous one (Propo. 2), where we replace $\mathbf{m}^i$ with $\mathbf{m'}^i = (\mathsf{H}(a^i\|\mathbf{m}^i)$.

5.4 The INT-CTXT-security of $\overline{\text{A10}}$ implies the INT-CTXT-security of $\overline{\text{A11}}$

Proposition 4. *Let* $\mathsf{f}^{\mathsf{IV}} : \mathcal{N} \times \mathcal{A} \longmapsto \mathcal{IV}$ *and* $\mathsf{f}^{\mathsf{Tag}_{11}} : \mathcal{M} \longmapsto \mathcal{T}$ *be two random functions and let* $\Pi = (\mathcal{K}_E, \mathsf{Enc}, \mathsf{Dec})$ *be a* $(q, t, \epsilon_{\mathsf{ivE}})$-ivE*-secure encryption*

scheme. Let $\overline{\boldsymbol{\Pi}}$ *be the* nAE *scheme obtained composing these components according to mode* $\overline{A11}$*. Let* $\mathsf{H} : \mathcal{A} \longmapsto \{0,1\}^N$ *be a* $(0, t, \epsilon_{cr})$*-collision resistant hash function.*
Then, if mode $\overline{A10}$*, implemented with the random function* $\mathsf{f}^{\mathsf{Tag}_{10}} : \mathcal{A} \times \mathcal{M} \longmapsto \mathcal{T}$*, is* $(q, t, \epsilon_{\mathsf{INT\text{-}CTXT1}})$-INT-CTXT1*-secure then mode* $\overline{A11}$ *is* $(q-1, t, \epsilon')$-INT-CTXT1*-secure with*

$$\epsilon = q|\mathcal{T}|^{-1} + (q-1)\epsilon_{\mathsf{ivE}} + \epsilon_{cr} + \epsilon_{\mathsf{INT\text{-}CTXT1}}.$$

The idea is to reduce the INT-CTXT1 adversary A against scheme $\overline{\boldsymbol{\Pi}}$ (mode $\overline{\text{A11}}$) to a INT-CTXT1 adversary C against scheme $\overline{\boldsymbol{\Pi}_{10}}$ (mode $\overline{\text{A10}}$). When the adversary A makes an encryption query $(n^i, a^i, \mathbf{m}^i)$, the adversary C makes an encryption query $(n^i \| \mathsf{H}(a^i), a, \mathbf{m}^i)$. When the adversary A makes his decryption query (n^q, a^q, c^q) the adversary A' simply asks the decryption of $(n^q \| \mathsf{H}(a^q), a, c^q)$.

6 Secure variants of modes N4, A10, A11 and A12

As a step towards the proof of the generic (in-)security of A10, A11 and A12, we consider two natural conditions on the ivE scheme that are sufficient to guarantee a secure composition. More precisely, we show that, if the ivE scheme is misuse resistant or if it is "message-malleable" (a condition that is satisfied by many standard modes, and that we formalize precisely below), then these modes are secure. Interestingly, these two properties are the two extreme of the range (clearly, it is impossible for a scheme to have both properties).

We prove everything only for mode A10, since the proofs can be straightforwardly extended to the other two modes. In this section we use the same replacement as in the previous one (we replace mode A10 with mode $\overline{\text{A10}}$). Surprisingly, we prove the same results for mode N4.

Then, we conclude this section, comparing our partial results about the (in)-security of modes A10, A11 and A12 with those of Namprempre et al. [20].

6.1 Misuse-resistant ivE scheme

The question is interesting since the misuse notion we consider (mrE, Def. 4) does not consider decryption queries.

Proposition 5. *Let the* ivE *scheme* Π *be a* $(q, t, \epsilon_{\mathsf{mrE}})$*-misuse resistant* mrE *and* $(q, t, \epsilon_{\mathsf{ivE}}) -$ ivE *secure, let* $\mathsf{f}^{\mathsf{IV}} : \mathcal{N} \times \mathcal{A} \longmapsto \mathcal{IV}$ *and* $\mathsf{f}^{\mathsf{Tag}} : \mathcal{A} \times \mathcal{M} \longmapsto \mathcal{T}$ *be two random functions. Then, the scheme* $\overline{\boldsymbol{\Pi}}$ *obtained composing these components according to mode* $\overline{A10}$*, is* $(q-1, t, (q-1)|\mathcal{T}|^{-1} + (q-1)\epsilon_{\mathsf{ivE}} + (q-1)\epsilon_{\mathsf{mrE}}) -$ INT-CTXT1*-secure.*

As seen above, we only need to consider the case not studied in Lemma 1. The idea of the proof is to reduce the INT-CTXT1 adversary to a mrE-adversary. Since we are not in the cases studied in Lemma 1, the couples (n^q, a^q) and $(a^q, \mathbf{m}^q)$

are not fresh, and it is enough for the mrE adversary to ask one more encryption query guessing that the message encrypted $\mathbf{m}^q \| \tau^q$ is one of the message the INT-CTXT1 adversary has already asked to encrypt with the same AD a^q (that is, $\mathbf{m}^q \in \mathcal{M}_{a^q}$ where $\mathcal{M}_{a^q} := \{\mathbf{m}_i \ i = 1, ..., q-1 \ s.t. a^i = a^q\}$). If the ciphertext obtained is the ciphertext c^q that he is asked to decrypt, then he outputs 1 and, otherwise, 0. The mrE adversary wins only if he guesses correctly and he can guess correctly at most with probability $(q-1)^{-1}$.
Allowing the mrE adversary to ask $(2q-2)$ encryption queries the scheme $\overline{A10}$ would be $(q, t, \frac{2q-1}{|\mathcal{T}|} + 2\epsilon_{\mathsf{mrE}}) - \mathsf{INT\text{-}CTXT1}$-secure, because the mrE adversary may try every possible message in $\mathcal{M}_{a^q}$.
We remember that for the misuse-resistance of Enc (Def. 3) the adversary has only access to encryption queries.

6.2 "Message-malleable" nE scheme

Definition 9. *A nonce-based encryption scheme* nE $\Pi = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ *is* message-malleable *if, given an encryption c of a message m with nonce n, an adversary can efficiently decrypt all couples (n, c'), i.e., he is able to compute m' s.t. $m' \leftarrow \mathsf{Dec}_k(n, c')$ without having access to a decryption oracle.*

The same definition may be done for ivE schemes. Many schemes (as CTR and OFB [14]) have this "malleability" property when they are used for fixed length messages. We detail some examples the extended version [8,10]. Message-malleability is easy to prove in many cases, e.g., when the ciphertext $c = \mathsf{Enc}_k^{iv}(m)$ is computed as a pseudorandom bitstream r computed from the iv and it is XORed with the message m (that is, $c = r \oplus m$), then $\mathsf{Dec}_k^{iv}(c') = c \oplus c' \oplus m$.

Message-malleability allows us to prove the following proposition for $\overline{A10}$, which can be easily extended to modes $\overline{A11}$ and $\overline{A12}$).

Proposition 6. *Let the* ivE *scheme Π be $(q, t, \epsilon_{\mathsf{ivE}})$-*ivE*-secure, $(q-1, t, \epsilon_{\mathsf{mCPA}})$-*mCPA*-secure and "message-malleable", let $\mathsf{f}^{\mathsf{IV}} : \mathcal{N} \times \mathcal{A} \longmapsto \mathcal{IV}$ and $\mathsf{f}^{\mathsf{Tag}} : \mathcal{A} \times \mathcal{M} \longmapsto \mathcal{T}$ be two random functions. Then, the scheme $\overline{\boldsymbol{\Pi}}$ obtained composing these components according to mode $\overline{A10}$, is $(q, t, (q-1)\epsilon_{\mathsf{ivE}} + q|\mathcal{T}|^{-1} + 8\epsilon_{\mathsf{mCPA}}) -$ INT-CTXT1-secure.*

Again, we only need to consider the case that is not covered by Lemma 1. The idea of the proof is to reduce the INT-CTXT1 adversary to an mCPA-adversary. Since we are not in the cases studied in Lemma 1, the couples (n^q, a^q) (thus iv^q) and $(a^q, \mathbf{m}^q)$ are not fresh. The mCPA adversary, when he is asked to simulate the AEnc oracle on input (n^i, a^i, m^i) simply computes iv^i and τ^i using the appropriate functions and asks his Enc oracle on input $(iv^i, m^i \| \tau^i, m^i \| r^i)$ where r^i is a random value picked in $\mathcal{T}$, receiving c^i which he forwards to the INT-CTXT adversary. When this latter adversary outputs (n^q, a^q, c^q), the mCPA adversary computes iv^q, which, due to the fact that we are in the case not covered by Lemma 1, is iv^j for a $j \in \{1, ..., q-1\}$. Now using the fact that Π is "nonce-message-malleable", he can decrypt c^q as if $c^i = \mathsf{Enc}_{k_E}^{iv^i}(m^i \| \tau^i)$. He outputs 0 if

the decryption query is valid, 1 otherwise. We observe that if $c^i = \mathsf{Enc}_{k_E}^{iv^i}(m^i \| r^i)$ the decryption query may be valid with probability $|\mathcal{T}|^{-1}$ since the tags have never been used before the decryption query.

6.3 Extension to N4

Surprisingly, although mode N4 is not secure in general (see Sec. 4), if the nE scheme is either misuse-resistant or message-malleable, mode N4 is INT-CTXT1-secure and, thus, nAE secure. It is easy to prove easily adapting the proofs of Propo. 5 and Propo. 6 to the nE case.
This implies that for N4 it is capital that the adversary can efficiently decrypt *everything*. In fact, the nE scheme used in Sec. 4 is message-malleable except in the case if $n = 1$ or 2 when trying to decrypt or encrypt $(v^*, \cdot)$.

6.4 Comparison to Namprempre et al. [20]

Namprempre et al. [20] gave partial results using the Knowledge-of-Tag property (KoT) (introduced in the extended version). That is, adversaries must forge without any (extractable) knowledge of the tag used in the decryption query [20]. With respect to their work, although the main ideas of the proofs are very similar, it is much easier to prove that a scheme is mrE or message-malleable, than to prove that a scheme is KoT-secure (while it may be easy to prove that it is not KoT-secure). In fact, to prove that a scheme is message-malleable it is enough to provide an algorithm which efficiently computes the result. On the other hand to prove that a scheme is not message-malleable (a part from proving that it is mrE), it must be proved that *all* efficient adversaries are not able always to decrypt. Similarly to prove the KoT security it must be proved that for *all* possible efficient extractors the scheme has this property, while to prove that a scheme is not KoT secure, it is enough to provide a counterexample.

7 Insecure variants of modes A10, A11 and A12

While, in the previous section, we proved the security of A10, A11 and A12 by making some extra requirements on the ivE scheme, this section considers the relaxation of some of the requirements on ivE that makes these 3 modes to become insecure. More precisely, we show how to compute forgeries against the INT-CTXT property of mode A10 when the ivE scheme is non tidy or stateful. These attacks imply that the three modes are not nAE-secure, when implemented with such schemes.

7.1 Tidiness as a security property

Given an IV-based encryption $\Pi = (\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$, our idea is to turn Π into a sloppy IV-based scheme. This modification augments the ciphertext $c = \mathsf{Enc}_k(iv, m)$ with $c' = \mathsf{Enc}_{k'}(iv, m)$, leading to a double and independent encryption m with

the same iv. It is easy to see that, for random iv, the new scheme has pseudorandom ciphertext $C = (c, c')$ as long as Π has pseudorandom ciphertext, (that is, if Π is ivE-secure). However, given iv, if we define the decryption of $C = (c, c')$ simply as $\mathsf{Dec}_k(iv, c)$ without any validity consideration on c', the new scheme is not tidy whether Π is tidy or not. Therefore, since the c' part of C is "out of control", any ciphertext $C' = (c, c'')$ decrypts to m and is deemed valid. Moreover, the A10 composition mode with two PRF-secure vecMACs does not rule out this malleability so that we can build a forgery with a single encryption query. Dropping the tidiness requirement of ivE, and then of nAE, is thus sufficient to leave a security breach in the resulting nAE.

More formally, we build $\Pi' = (\mathcal{K}, \mathsf{Enc}', \mathsf{Dec}')$ with keyspace $\mathcal{K}^2$, message space $\mathcal{M}$ and ciphertext space $\mathcal{C}^2$ as follows: $\mathsf{Enc}'_{(k,k')}(iv, m)$ outputs $C = (c, c')$ where $c = \mathsf{Enc}_k(iv, m)$ and $c' = \mathsf{Enc}_{k'}(iv, m)$; $\mathsf{Dec}'_{(k,k')}(iv, C)$ parses the ciphertext as $C = (c, c')$ and outputs $m = \mathsf{Dec}_k(iv, c)$. For any $c'' \neq c'$, we have $\mathsf{Enc}'(iv, \mathsf{Dec}'(iv, (c, c''))) = (c, c') \neq (c, c'')$ so that Π' is not tidy.

Let nAE be the authenticated encryption obtained from the A10 mode whose ciphertext has the form $C = (c, c')$ where $c = \mathsf{Enc}_k(iv, \mathbf{m}\|\tau)$ and $c' = \mathsf{Enc}_{k'}(iv, \mathbf{m}\|\tau)$ with $iv = \mathsf{Mac}^{\mathsf{IV}}_{k_M}(n, a)$ and $\tau = \mathsf{Mac}^{\mathsf{Tag}}_{k_M}(a, \mathbf{m})$. Now, we consider the forger A which makes a single encryption query on any triple $(n, a, \mathbf{m})$ and receives back $C = (c, c')$ as above. Then, A picks any (samplable) $c'' \in \mathcal{C}$ distinct of c' and outputs $C^\star = (c, c'')$. Following the description of the A10 mode we find that the decryption starts by running $iv = \mathsf{Mac}^{\mathsf{IV}}_{k_M}(n, a)$ and then $\mathsf{Dec}'_{(k,k')}(iv, C^\star) = \mathsf{Dec}_k(iv, c) = \mathbf{m}\|\tau$. Finally, since the check $\tau = \mathsf{Mac}^{\mathsf{Tag}}_{k_M}(a, \mathbf{m})$ passes $\mathbf{m} \neq \perp$ is returned although $C^\star \neq C$.

Message-malleability. In order to further emphasize the crucial role of the tidiness in the insecurity of the authenticated encryption based on Π', we stress that if the underlying IV-based scheme Π is tidy and message-malleable (Def. 9), the A10 composition implemented with Π leads to an nAE-secure scheme (as shown in Sec. 6.2). However, even if Π' is not tidy, it is easy to see that Π' remains message-malleable while we proved that it never leads to a nAE-secure scheme. As a summary, (non) tidiness alone has an intrinsic propensity to degrade the nAE-security of the AEnc based on the Tag-then-Encrypt paradigm.

7.2 Forgery against stateful A10, A11 & A12

In stateful AE schemes the AEnc and ADec algorithms receive at the start of the game an additional input, the *state*, which is updated during every call and kept in memory to be reused in the following call. The scheme we use has a stateless ADec algorithm, that is, it does not use the state and every reordering and omission is tolerable (L_0 of Rogaway et al. [28]). With respect to their work we allow the adversary to choose the state at the start of the game.

The idea of this forgery is to use the state, which in our case is simply a counter of the encryption queries, as the nonce was used in the attack against

mode N4 (Sec. 4). At the end of the section we discuss the meaning of tidiness for stateful schemes.

The ivE we present is an adaptation of the nE scheme used in Sec. 4. As there, we present it only for N-bit long message, leaving the general case to the extended version. The main changes are:

- We use a TPRP $\mathsf{E} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^{\lambda} \longmapsto \{0,1\}^{\lambda}$.
- A new block c_{-1} is added to the ciphertext, in order to give the decryption algorithm the actual value of the counter *ctr* which is *an internal state only of the encryption device*, and $c_{-1} = \mathsf{E}_k^{(0,1)}(ctr)$. The Dec algorithm inverts this to retrieve the correct *ctr*. The block c_{-1} is random since it is always obtained with different inputs (as long as the number of encryption queries is $<< 2^n$).
- To compute this block, the TPRP E is called with a tweak $(0,1)$ that is never used else
- The boxed **if** is triggered by the value of the counter *ctr* (not of the nonce) and m
- The iv replaces the nonce n in the input of the TPRP E.

Note again that, due to mode A10, the messages which the nAE scheme $\boldsymbol{\Pi}$ can encrypt, are λ bits long, while those which Π can encrypt are 2λ bits long.

The forgery is an easy adaptation of that presented in Sec. 4 .

The scheme Π is clearly ivE-secure, the only important change with Sec. 4 is the fact that we have to consider also the block c_{-1}). Now we have to discuss what means for a stateful nAE (or nE or ivE) scheme to be *tidy*.
For stateless nAE schemes the definition was given in Def. 2 (similarly for nE and ivE): if $\mathsf{ADec}_k^{n,a}(c) = m \neq \perp$ then $\mathsf{AEnc}_k^{n,a}(m) = c$.

Now if the nAE scheme is stateful it means that $\mathsf{AEnc}_k^{n,a}(m)$ is no more defined, because the state s may influence the output of $\mathsf{AEnc}_k^{(\cdot,\cdot)}(\cdot)$. Thus, denoting with $\mathcal{S}$ the set of possible states, we redefine tidiness as:

Definition 10. *We say that an* nAE *scheme is* tidy *if* $\mathsf{ADec}_k^{n,a}(c) = m$ *then* $c \in \{\mathsf{AEnc}_{k,s}^{n,a}(m)\}_{s\in\mathcal{S}}$.
Similarly an nE *(resp. an* ivE*) scheme is* tidy *if* $\mathsf{Dec}_k^{n,a}(c) = m$ *(resp.* $\mathsf{Dec}_k^{iv,a}(c) = m$ *then* $c \in \{\mathsf{Enc}_{k,s}^{n,a}(m)\}_{s\in\mathcal{S}}$ *(resp.* $c \in \{\mathsf{Enc}_{k,s}^{iv,a}(m)\}_{s\in\mathcal{S}}$*).*

According with this new definition, the ivE scheme Π which we have just used, presented, is tidy, as it follows from a close inspection of the pseudocode provided, thus the nAE scheme $\boldsymbol{\Pi}$ is tidy.

We have also to redefine for stateful schemes all the notions presented in Sec. 2. We do it allowing the adversary *at the start of the game* to set the state of the scheme as he wishes.

8 Conclusion

In this paper we have studied four generic composition modes, N4, A10, A11 and A12, for building authenticated encryption for an encryption scheme and a PRF MAC. The security of these four modes was left open in previous works, and three of them are the most efficient among the 180 possible modes based on these building blocks.

We have proved that mode N4 is not secure in general, and that modes A10, A11 and A12 have equivalent security. Moreover we have proved that if these four modes are instantiated with many common schemes (like CTR, OFB) they are all secure. Finally, we have showed that tidiness (again) and being stateless can have a decisive impact on security, as the application of A10, A11 and A12 on untidy or stateful modes can lead to insecure solutions.

Having used an encryption scheme using ciphertext expansion to break N4, we leave as a question for future work the proof of the security of N4 if the encryption scheme does not expand the ciphertext.

Our analysis still leaves as an open problem to decide if modes A10, A11, and A12 are secure in general.

Acknowledgments Thomas Peters is a postdoctoral researcher of the Belgian Fund for Scientific Research (F.R.S.-FNRS). This work has been funded in parts by the European Union (EU) and the Walloon Region through the FEDER project USERMedia (convention number 501907-379156) and the ERC project SWORD (convention number 724725).

References

1. Vijayalakshmi Atluri, editor. *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*. ACM, 2002.
2. Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 394–403. IEEE Computer Society, 1997.
3. Mihir Bellare, Tadayoshi Kohno, and Chanathip Namprempre. Authenticated encryption in SSH: provably fixing the SSH binary packet protocol. In Atluri [1], pages 1–11.
4. Mihir Bellare, Tadayoshi Kohno, and Chanathip Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the encode-then-encrypt-and-mac paradigm. *ACM Trans. Inf. Syst. Secur.*, 7(2):206–241, 2004.
5. Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Okamoto [22], pages 531–545.
6. Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In Okamoto [22], pages 317–330.

7. Dan J Bernstein. Caesar call for submissions, final, january 27 2014.
8. Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Ciphertext integrity with misuse and leakage: Definition and efficient constructions with symmetric primitives. In Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim, editors, *Proceedings of the 2018 on Asia Conference on Computer and Communications Security, AsiaCCS 2018, Incheon, Republic of Korea, June 04-08, 2018*, pages 37–50. ACM, 2018.
9. Francesco Berti, Olivier Pereira, and Thomas Peters. Reconsidering generic composition: the tag-then-encrypt case. Cryptology ePrint Archive, Report 2018/991, 2018. https://eprint.iacr.org/2018/991.
10. Francesco Berti, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. On leakage-resilient authenticated encryption with decryption leakages. *IACR Trans. Symmetric Cryptol.*, 2017(3):271–293, 2017.
11. Colin Boyd, Britta Hale, Stig Frode Mjølsnes, and Douglas Stebila. From stateless to stateful: Generic authentication and authenticated encryption constructions with application to TLS. In Kazue Sako, editor, *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, volume 9610 of *Lecture Notes in Computer Science*, pages 55–71. Springer, 2016.
12. Morris J Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. Technical report, 2007.
13. IETF. The transport layer security (tls) protocol version 1.3 draft-ietf-tls-tls13-28. Technical report, 2018. https://tools.ietf.org/html/draft-ietf-tls-tls13-28.
14. Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. CRC Press, 2014.
15. Jonathan Katz and Moti Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In Bruce Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 284–299. Springer, 2000.
16. Tadayoshi Kohno, Adriana Palacio, and John Black. Building secure cryptographic transforms, or how to encrypt and MAC. *IACR Cryptology ePrint Archive*, 2003:177, 2003.
17. Hugo Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is ssl?). In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 310–331. Springer, 2001.
18. Moses Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable block ciphers. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.
19. David A. McGrew. An interface and algorithms for authenticated encryption. *RFC*, 5116:1–22, 2008.
20. Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering generic composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on*

the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings, volume 8441 of *Lecture Notes in Computer Science*, pages 257–274. Springer, 2014.
21. Yoav Nir and Adam Langley. Chacha20 and poly1305 for IETF protocols. *RFC*, 7539:1–45, 2015.
22. Tatsuaki Okamoto, editor. *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*. Springer, 2000.
23. Kenneth G. Paterson, Thomas Ristenpart, and Thomas Shrimpton. Tag size does matter: Attacks and proofs for the TLS record protocol. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 372–389. Springer, 2011.
24. Thomas Peyrin and Yannick Seurin. Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 33–63. Springer, 2016.
25. Phillip Rogaway. Authenticated-encryption with associated-data. In Atluri [1], pages 98–107.
26. Phillip Rogaway. Nonce-based symmetric encryption. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2004.
27. Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.
28. Phillip Rogaway and Yusi Zhang. Simplifying game-based definitions - indistinguishability up to correctness and its application to stateful AE. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2018.