Reconsidering Generic Composition: the modes A10, A11 and A12 are insecure

Francesco Berti*
francesco.berti@biu.ac.il

Bar-Ilan University, Ramat-Gan 529002, Israel

Abstract. Authenticated Encryption (AE) achieves privacy and authenticity with a single scheme. It is possible to obtain an AE scheme gluing together an encryption scheme (privacy secure) and a Message Authentication Code (authenticity secure). This approach is called *generic composition* and its security has been studied by Namprempre et al. [NRS14]. They looked into all the possible gluings of an encryption scheme with a secure MAC to obtain a nonce-based AE-scheme. The encryption scheme is either IV-based (that is, with an additional random input, the initialization vector [IV]) or nonce-based (with an input to be used once, the *nonce*). Nampremepre et al. assessed the security/insecurity of all possible composition combinations except for 4 (N4, A10, A11 and A12). Berti et al. [BPP18a] showed that N4 is insecure and that the remaining modes (A10, A11, and A12) are either all secure or insecure.
Here, we prove that these modes are all insecure with a counterexample.

Keywords: AE · generic composition · integrity

1 Introduction

Privacy and authenticity are two of the most important goals of cryptography. Encryption schemes provide privacy, that is, no information about a plaintext (except its length) can be obtained from a ciphertext encrypting it; while Message Authentication Codes (MAC) provide authenticity, that is, it is not possible to send a message impersonating another person. *Authenticated Encryption* (AE) is the cryptographic primitive that provides both. In addition, AE allows the presence of Associated Data (AD), which are data sent in clear but authenticated. This primitive is the object of flourishing research from the seminal papers [BN00,Rog02,RS06,BN08], with many constructions proposed, see for example [RBBK01,BDH$^+$17,HKR15,PS16,BDPA11,BMPS21] and the CAESAR competition [Ber14,AFL16]. Moreover, there is an ongoing NIST competition for a lightweight AE scheme [NIS18], whose finalists have been announced [NIS21] and whose winner is ASCON [DEMS21]. See [JZK$^+$22] for a survey of the AE-literature.

* Work done when this author was at TU Darmstadt, Germany, CAC - Applied Cryptography

It is possible to design an AE-scheme from scratch (as the case of OCB [RBBK01], for example) or to combine an encryption scheme with a MAC. This second approach is called *generic composition* [BN00].
About generic composition, the first result is the well-known "Encrypt-then-MAC is secure" [BR00,Kra01]. Namprempre et al. [NRS14] studied thoroughly the generic composition problem. They realised that while the first result [BR00,Kra01] assumed that the encryption scheme is probabilistic, the literature moved to IV-based or nonce-based encryption schemes [RS06,KL14]. Since probabilistic encryption schemes are hard to design, we usually use a deterministic encryption scheme and provide the random coins needed externally with an initialization vector, the IV [BDJR97]. These are the IV-based encryption schemes.
Unfortunately, in practice, the IV is not always sampled as it should, that is, uniformly at random [RS06]. Thus, we can replace the IV with a *nonce* ("number used once"). Nonce-based encryption schemes are assumed to be secure as long as the nonce is not repeated [RS06].
Namprempre et al. [NRS14] studied all possible combinations of IV-based and nonce-based encryption schemes with prf-MACs (that is, MACs which provide a pseudo-random tag) to obtain a nonce-based AE scheme. They proved that 164 modes are insecure, 12 secure (9 with IV-based encryption schemes and 3 with nonce-based encryption schemes). Only 4 modes remained elusive: N4 (using a nonce-based encryption scheme) and A10, A11, and A12 (using an IV-based), see Fig. 1. For these modes, the security remained undecided.
Note that all these modes follow the MAC-then-Encrypt paradigm. Moreover, N4, A11, and A12 are among the "most efficient" AE-composition modes, in the sense that they use the nonce, the AD, and the message the least possible number of times (and there is the hope that they are secure).
Finally, their security has been proved using an additional hypothesis: the "Knowledge-of-Tags" (KOT) [NRS14]. However, the problem of knowing if KOT is implied by the privacy requirement of the encryption scheme remains.

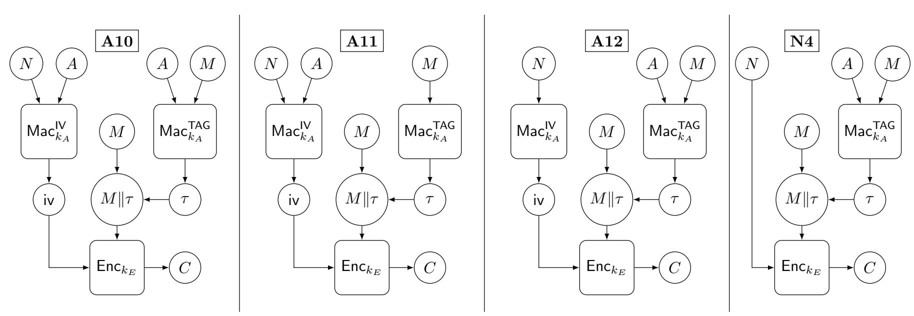

Fig. 1. The modes A10, A11, A12 and N4 [NRS14]. Note that the IV used may be sent in clear along with the ciphertext to speed decryption.

Berti et al. [BPP18a] investigated the security of these 4 modes, giving some partial results. First, they proved that N4 is not secure, offering a counterexample with a nonce-based encryption scheme Π with "a kind of Trojan injected". Unfortunately, Π outputs ciphertexts longer than the plaintext. Second, they proved that modes A10, A11, and A12 have the same security: from a counterexample against any of them, we can build counterexamples against the other 2 modes. Third, they proved that the modes A10, A11, and A12 are secure if the secure encryption scheme has any of these two hypotheses: either "misuse-resistance" (that is, using the same IV and different messages, the encryption schemes still outputs pseudorandom ciphertexts) or "message-malleability" (that is, having the encryption of any message with a given IV iv, the adversary can correctly encrypt all other messages with that iv). Since these two hypotheses are, in a certain way, one the opposite of the other, it seems that these modes are secure, but we still do not have the proof.

Our contribution. Surprisingly, this paper proves that modes A10, A11, and A12 are not secure in general. In particular, they do not provide authenticity. That is, being able to encrypt messages, it is possible to produce a triple (nonce, AD, ciphertext), (N^*, A^*, C^*) which is fresh and valid (that is, $\mathsf{ADec}(N^*, A^*, C^*)$ does not answer "invalid"). We exhibit a counterexample: a secure ivE scheme Π_1, whose composition according to mode A12 with a secure prf-MAC, we can forge. Using [BPP18a], we immediately extend this result to modes A10 and A11.
Π_1 uses a tweakable block-cipher [1] (TBC) F. If the message m is s.t. the first two blocks are different, (that is, $m_1 \neq m_2$), substantially Π_1 is a TBC-based version of CTR, that is $c_i = \mathsf{F}^{1,i}(\mathsf{iv}) \oplus m_i$ with the difference that the last block (the one carrying the tag in A12) is encrypted with a slightly different tweak, that is, $c_l = \mathsf{F}^{2,l}(\mathsf{iv}) \oplus m_i$. If the first two message blocks are equal, instead, we modify the encryption of the two last ciphertext blocks: the second-to-last ciphertext block is obtained as $c_{l-1} = \mathsf{F}^{2,m_3}(\mathsf{iv}) \oplus \mathsf{F}^{2,m_3}(m_1) \oplus m_{l-1}$, while the last block is obtained as $c_l = \mathsf{F}^{1,l}(\mathsf{iv}) \oplus m_l$. Further, we assume that Π_1 outputs the IV iv it uses with the ciphertext c. Π_1 is IV-secure, as we prove in Thm. 1.
When mode A12 is implemented with Π_1, a forgery can be created, proceeding as follow: the attacker asks the encryption of a message $M = M_1, ..., M_l$ with nonce N and AD A, obtaining $\mathsf{iv}_1, C_1, ..., C_{l+1}$ (Remind that $C = c = \mathsf{Enc}(\mathsf{iv}, m)$ with $m = M \| \tau$). Then, she asks the encryption C' of N', A', M', where M' is one block longer than M and $M' = \mathsf{iv}, \mathsf{iv}, l+1, ...,$. Our goal is to produce C^* s.t. $\mathsf{ADec}(N', A, C^*) = M$. From C' it is easy to compute the correct $C_1^*, ..., C_l^*$, while to compute C_{l+1}^* (the block encrypting the tag τ), we need both C_{l+1} and C'_{l+1}. The details are in Sec. 4.1.
Since to obtain the correct C_l^*, the adversary needs to know the IV iv used by Π_1 to produce C, a natural solution seems to use the new syntax introduced by Bellare et al. [BNT19]. They assumed that the decryption algorithm needs only to know the ciphertext (and the key) to decrypt correctly (and not the IV, or

[1] Tweakable block ciphers (TBCs) were introduced by Liskov et al. [LRW02]. They are block-ciphers (BCs) with an additional input, the *tweak*, to add flexibility.

the nonce). Unfortunately, this simple solution does not work. In fact, we offer as a counterexample Π_2, a variant of Π_1, where the IV is sent as $C_0 = \mathsf{F}^{0,0}(\mathsf{iv})$. Third, we show that, to prove that N4 is not secure, we do not need an encryption scheme outputting ciphertexts longer than the plaintexts. We offer two counterexamples: a variant of Π_1 and a variant of the scheme Π presented in [BPP18a]. Since TBCs can be build from BCs [LRW02], our construction can be built only from BCs.

This work concludes the classification of all generic composition modes (when the encryption scheme is either nonce-based or IV-based). Moreover, we have proved that IV-security does not imply KOT.

2 Background

Notations. We denote with $\{0,1\}^n$ the set of all n-bit long strings and with $\{0,1\}^*$ the set of all finite strings. We denote the length of the string x with $|x|$. To denote that x is picked uniformly at random from the set $\mathcal{X}$, we use $x \xleftarrow{\$} \mathcal{X}$. In our security games, we use *adversaries*, which are probabilistic algorithms. An adversary A who has access to oracles $\mathsf{O}_1, \dots, \mathsf{O}_T$ is denoted with $\mathsf{A}^{\mathsf{O}_1(\cdot),\dots,\mathsf{O}_T(\cdot)}$. A $(q_1, \dots, q_T, t)$-*adversary* A can do at most q_i queries to oracle O_i and runs in time bounded by t. We denote with $\mathsf{A}^{\mathsf{O}_1(\cdot),\dots,\mathsf{O}_T(\cdot)} \Rightarrow x$ the fact that the adversary A outputs x.

2.1 Tweakable blockciphers (TBCs)

Encryption schemes and MACs usually use (tweakable)-block ciphers to produce the randomness they need. Formally,

Definition 1. *A* tweakable blockcipher *(*TBC*) is a function* $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^n \to \{0,1\}^n$ *s.t.* $\forall (k, tw) \in \mathcal{K} \times \mathcal{TW}$, $\mathsf{F}(k, tw, \cdot) : \{0,1\}^n \to \{0,1\}^n$ *is a permutation.*

We use often $\mathsf{F}_k^{tw}(x)$ and $\mathsf{F}_k(tw, x)$ to denote $\mathsf{F}(k, tw, x)$. To denote the inverse of $\mathsf{F}_k^{tw}(\cdot)$, we use $\mathsf{F}_k^{-1,tw}(\cdot)$. We call n the *block-length* of F.

We want that a TBC outputs values indistinguishable from random ones. Formally:

Definition 2. *A* TBC $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^n \to \{0,1\}^n$ *is a* (q, t, ϵ)-tweakable pseudorandom permutation (tprp) *if* $\forall$ (q, t)*-adversary* A*, the following advantage*

$$\left|\Pr[\mathsf{A}^{\mathsf{F}_k(\cdot,\cdot)} \Rightarrow 1] - \Pr[\mathsf{A}^{\mathsf{f}(\cdot,\cdot)} \Rightarrow 1]\right| \leq \epsilon$$

where $k \xleftarrow{\$} \mathcal{K}$ *and* $\mathsf{f} \xleftarrow{\$} \mathcal{TWP}$*.* $\mathcal{TWP}$ *is the set of all tweakable permutations* f*, that is, the functions* $\mathsf{f} : \mathcal{TW} \times \{0,1\}^n \to \{0,1\}^n$ *s.t.* $\forall tw \in \mathcal{TW}$, $\mathsf{f}(tw, \cdot) : \{0,1\}^n \to \{0,1\}^n$ *is a permutation.*

When the adversary, even having access also to the inverse of F, cannot distinguish F from f, we say that F is a strong tprp. Formally:

Definition 3. *A* TBC $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^n \rightarrow \{0,1\}^n$ *is a* (q,t,ϵ)-strong tweakable pseudorandom permutation (stprp) *if* $\forall$ (q_1, q_2, t)*-adversary* A*, the following advantage*

$$\left|\Pr[\mathsf{A}^{\mathsf{F}_k(\cdot,\cdot),\mathsf{F}_k^{-1}(\cdot,\cdot)} \Rightarrow 1] - \Pr[\mathsf{A}^{\mathsf{f}(\cdot,\cdot),\mathsf{f}^{-1}(\cdot,\cdot)} \Rightarrow 1]\right| \leq \epsilon$$

where $k \xleftarrow{\$} \mathcal{K}$, $\mathsf{f} \xleftarrow{\$} \mathcal{TWP}$, $\mathsf{f}^{-1}(\cdot,\cdot)$ *is the inverse of* f*, and* $q_1 + q_2 \leq q$.

When we do not need that F is a permutation, we use the following security definition

Definition 4. *A* (q,t,ϵ)-pseudorandom function (prf) *is a function* $\mathsf{F} : \mathcal{K} \times \{0,1\}^{n'} \rightarrow \{0,1\}^n$ *s.t.* $\forall(q,t)$*-adversary* A*, the following advantage*

$$\left|\Pr[\mathsf{A}^{\mathsf{F}_k(\cdot)} \Rightarrow 1] - \Pr[\mathsf{A}^{\mathsf{f}(\cdot)} \Rightarrow 1]\right| \leq \epsilon$$

where $k \xleftarrow{\$} \mathcal{K}$ *and* $\mathsf{f} \xleftarrow{\$} \mathcal{RF}$ *with* $\mathcal{RF}$ *is the set of all functions* f *which are the functions* $\mathsf{f} : \{0,1\}^{n'} \rightarrow \{0,1\}^n$.

Note that tprp-secure implies prf-secure [KL14].

2.2 Encryption schemes

Encryption schemes are the cryptographic primitive used to provide privacy. To have security, we need that the encryption is probabilistic [KL14]. Often, to have probabilistic encryption, we use a random input, called the *initialization vector* (IV), or an input used only once, called a *nonce*. Thus, we have IV-based and nonce-based encryption scheme. Formally:

Definition 5. *An* IV-based encryption (ivE) scheme *is a triple* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *where*

- *the* key-generation algorithm Gen *generates a* key k_E *from the keyspace* $\mathcal{K}_E$ *(usually* $k_E \xleftarrow{\$} \mathcal{K}$*);*
- *the* encryption algorithm Enc *takes as input a key* $k_E \in \mathcal{K}_E$*, an initialization vector (*IV*)* iv *in the* IV*-space (*$\mathcal{IV}$*), and a message* m *in the message space* $m \in \mathcal{M}$*, and outputs a string* $c \leftarrow \mathsf{Enc}_{k_E}^{\mathsf{iv}}(m)$ *called* ciphertext*;*
- *the* decryption algorithm Dec *takes as input a key* $k_E \in \mathcal{K}_E$*, an* IV $\mathsf{iv} \in \mathcal{IV}$*, and a ciphertext* $c \in \{0,1\}^*$*, and outputs either a string* $m \in \mathcal{M}$ *or the symbol* $\perp$ *("invalid"); we denote this with* $m/\perp \leftarrow \mathsf{Dec}_{k_E}^{\mathsf{iv}}(c)$.

We require that Enc *and* Dec *are the "inverse" of the other. That is,*

- correctness: *if* $\mathsf{Enc}_{k_E}^{\mathsf{iv}}(m) = c$ *(when defined), then,* $\mathsf{Dec}_{k_E}^{\mathsf{iv}}(c) = m$*;*
- tidyness: *if* $\mathsf{Dec}_{k_E}^{\mathsf{iv}}(c) = m \neq \perp$*, then,* $\mathsf{Enc}_{k_E}^{\mathsf{iv}}(m) = c$.

We assume that the length of the ciphertexts does not depend on the key and on the IV*, that is,* $\forall m \in \mathcal{M}\ |\mathsf{Enc}^{\mathsf{iv}}_{k_E}(m)| = |\mathsf{Enc}^{\mathsf{iv}'}_{k'_E}(m)|\ \forall k_E, k'_E \in \mathcal{K}_E,\ \mathsf{iv}, \mathsf{iv}' \in \mathcal{IV}$.
A nonce-based encryption scheme (nE) *is defined as an* IV*-based encryption scheme where the* IV iv *is replaced with a nonce* n*.*

To distinguish nonce from block-size, we use always capital letters for nonces, e.g. N.

Note that syntactically, ivE and nE schemes are the same. But, their security definitions are different: we want that the ciphertexts are indistinguishable from random when the IVs are randomly picked (for ivE) or used only once (for nE). Formally:

Definition 6. *An* ivE *encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* (q, t, ϵ)*-secure (*ivE*) if* $\forall (q, t)$*-adversary* A*, the following advantage*

$$\left|\Pr[\mathsf{A}^{\mathsf{Enc}^{\$}_{k_E}(\cdot)} \Rightarrow 1] - \Pr[\mathsf{A}^{\$(\cdot)} \Rightarrow 1]\right| \leq \epsilon$$

where $k_E \leftarrow \mathsf{Gen}$*,* $\mathsf{Enc}^{\$}_{k_E}(m)$*, first, randomly picks the* IV*,* $\mathsf{iv} \xleftarrow{\$} \mathcal{IV}$ *and then outputs* $c \leftarrow \mathsf{Enc}^{\mathsf{iv}}_{k_E}(m)$*, and* \$ *picks* $(\mathsf{iv}, c) \xleftarrow{\$} \mathcal{IV} \times \{0,1\}^{|\mathsf{Enc}^{\$}_{k_E}(m)|}$ *uniformly at random.*

Note that iv is picked in the same way in both cases.

Definition 7. *An* nE *encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* (q, t, ϵ)*-secure (*nE*) if* $\forall (q, t)$*-adversary* A*, the following advantage*

$$\left|\Pr[\mathsf{A}^{\mathsf{Enc}_{k_E}(\cdot,\cdot)} \Rightarrow 1] - \Pr[\mathsf{A}^{\$(\cdot,\cdot)} \Rightarrow 1]\right| \leq \epsilon$$

where $k_E \leftarrow \mathsf{Gen}$*, and* \$ *picks* $c \xleftarrow{\$} \{0,1\}^{|\mathsf{Enc}_{k_E}(N,m)|}$ *uniformly at random. The adversary is not allowed to do a query on input* (N, m) *if she has already done a query on input* (N, m') *for* $m \neq m'$*. That is, each nonce* N *is used at most once.*

For both ivE and nE-security, the adversary cannot query the decryption oracle (or an ideal counterpart).

2.3 Message Authentication Codes (MAC)

Message authentication codes (MACs) are the cryptographic primitive used for authenticity.

Definition 8. *A* MAC *is a triple* $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ *where*
- *the* key-generation algorithm Gen *generates a* key k_A *from the keyspace* $\mathcal{K}_{\mathsf{A}}$ *(usually* $k_A \xleftarrow{\$} \mathcal{K}_{\mathsf{A}}$*);*

- *the* tag-generation algorithm Mac *takes as input a key* $k_A \in \mathcal{K}_{\mathsf{A}}$*, and a value* x *in the domain space* $x \in \mathcal{X}$*, and outputs a string called* tag $\tau \leftarrow \mathsf{Mac}_{k_A}(x)$*;*
- *the* verification algorithm Vrfy *takes as input a key* $k_A \in \mathcal{K}_{\mathsf{A}}$*, a value* $x \in \mathcal{X}$ *and a tag* τ*, and outputs either a string* $\top$ *("valid") or the symbol* $\perp$ *("invalid") and we denote this with* $\top / \perp \leftarrow \mathsf{Vrfy}_{k_A}(x, \tau)$*.*

We require that Mac *and* Vrfy *are one the "inverse" of the other. That is,*
- correctness: *if* $\mathsf{Mac}_{k_A}(x) = \tau$ *(when defined), then,* $\mathsf{Vrfy}_{k_A}(x, \tau) = \top$*;*
- tidyness: *if* $\mathsf{Vrfy}_{k_A}(x, \tau) = \top$*, then,* $\mathsf{Mac}_{k_A}(x) = \tau$*.*

The tidiness is implied, when the verification algorithm is the most obvious: on input (x, τ), Vrfy_{k_A} computes $\tau' = \mathsf{Mac}_{k_A}(x)$ and checks if $\tau = \tau'$.

The security definition that we use for MAC, as in [NRS14], is not standard: we ask that Mac is a prf. Formally,

Definition 9. *A* MAC $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ *is* (q, t, ϵ)-prf secure *if* Mac *is a* (q, t, ϵ)-prf *where the key is picked according to* Gen*.*

The standard definition (unforgeability, see [KL14]) is implied by this definition, but it is not " a suitable starting point when the goal is to create a nAE scheme " [NRS14].

2.4 Authenticated Encryption (AE)

Authenticated Encryption is the cryptographic primitive used to provide both privacy and authenticity. We assume, following [Rog02], that there is a nonce, and there are data to be authenticated but not encrypted. They are called *Associated Data* (AD).

Definition 10. *A* nonce-based authenticated encryption (nAE) *is a triple* $\Pi = (\mathsf{Gen}, \mathsf{AEnc}, \mathsf{ADec})$ *where*
- *the* key-generation algorithm Gen *generates a* key K *from the keyspace* $\mathcal{K}_{\mathsf{AE}}$ *(usually* $K \xleftarrow{\$} \mathcal{K}_{\mathsf{AE}}$*);*
- *the* encryption algorithm AEnc *takes as input a key* $K \in \mathcal{K}_{\mathsf{AE}}$*, a nonce* N *in the nonce-space (*$\mathcal{N}$*), an associated data* A *in the associated data space (*$\mathcal{A}$*), and a message* M *in the message space* $M \in \mathcal{M}_{\mathsf{AE}}$*, and outputs a string* $C \leftarrow \mathsf{AEnc}_K(N, A, M)$ *called* ciphertext*;*
- *the* decryption algorithm ADec *takes as input a key* $K \in \mathcal{K}_{\mathsf{AE}}$*, a nonce* $N \in \mathcal{N}$ *, and a ciphertext* $C \in \{0,1\}^*$*, and outputs either a string* $M \in \mathcal{M}_{\mathsf{AE}}$ *or the symbol* $\perp$ *("invalid"); we denote this with* $M / \perp \leftarrow \mathsf{ADec}_K(N, A, C)$*.*

We require that AEnc *and* ADec *are one the "inverse" of the other. That is,*
- correctness: *if* $\mathsf{AEnc}_K(N, A, M) = C$ *(when defined), then,* $\mathsf{ADec}_K(N, A, C) = M$*;*
- tidyness: *if* $\mathsf{ADec}_K(N, A, C) = M \neq \perp$*, then,* $\mathsf{AEnc}_K(N, A, M) = C$*.*

We assume that the length of the ciphertext does not depend on the key K*, that is,* $\forall (N, A, M) \in \mathcal{N} \times \mathcal{A} \times \mathcal{M}_{\mathsf{AE}}$ $|\mathsf{AEnc}_K(N, A, M)| = |\mathsf{AEnc}_{K'}(N, A, M)|$ $\forall K, K' \in \mathcal{K}_{\mathsf{AE}}$*.*

Note that, syntactically, nAE schemes are very similar to nE schemes (Def. 5) with the addition of associated data.
To make the reading clearer, we use capital letters (e.g., M) for the inputs of AEnc and ADec, while small letters (e.g., m) for the inputs of Enc, Dec, Mac, and Vrfy. This will make the next section more accessible.

nAE schemes want to provide privacy and authenticity with the same scheme. The following definition captures this:

Definition 11. *An* nAE *encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{AEnc}, \mathsf{ADec})$ *is* (q_1, q_2, t, ϵ)*-secure (*nAE*) if* $\forall (q_1, q_2, t)$*-adversary* A*, the following advantage*

$$\left| \Pr[\mathsf{A}^{\mathsf{AEnc}_K(\cdot,\cdot,\cdot),\mathsf{ADec}_K(\cdot,\cdot,\cdot)} \Rightarrow 1] - \Pr[\mathsf{A}^{\$(\cdot,\cdot,\cdot),\perp(\cdot,\cdot,\cdot)} \Rightarrow 1] \right| \leq \epsilon$$

where $K \leftarrow \mathsf{Gen}$*,* $\$(N, A, M)$ *outputs a random string with the same length as* $\mathsf{AEnc}_K(N, A, M)$*, and* $\perp(\cdot,\cdot,\cdot)$ *always outputs* $\perp$*. The adversary is not allowed to ask her second oracle on input* (N, A, C) *if she has received* C *as an answer to a query to the first oracle on input* (N, A, M) *for any* $M \in \mathcal{M}_{\mathsf{AE}}$*. Moreover, the adversary cannot query her first oracle on input* (N, A, M) *if she has already queried her first oracle on input* (N, A', M')*. That is, each nonce* N *is used at most once during "encryption" (first oracle) queries.*

This notion implies that the adversary cannot find a *forgery*, that is a triple (N, A, C) which is *fresh* and *valid*, that is, (N, A, C) does not come as answer to a previous query on input (N, A, M) $[C = \mathsf{AEnc}_K(N, A, M)]$ and $\mathsf{ADec}_K(N, A, C) \neq \perp$.

3 Generic composition and the elusive generic composition modes

3.1 Generic composition

A natural way to obtain an AE scheme is to compose an encryption scheme with a MAC [BN00]. This approach is the so-called *generic composition*. In the original paper considering the security of the generic composition [BN00], the authors studied the composition of a *probabilistic* encryption schemes[2] with a MAC. There are three possible composition methods: *Encrypt-and-*MAC*,* MAC*-then-Encrypt*, and *Encrypt-then-*MAC. They proved that Encrypt-then-MAC is always secure.

Namprempre et al. [NRS14] studied the generic composition when the encryption scheme is either ivE or nE-based and the MAC scheme is prf-secure. For ivE-based, the prf-MAC provides both the IV to the ivE scheme and the tag. To prevent trivial attacks, there is the domain separation between these two calls,

[2] A probabilistic encryption scheme is a triple $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ s.t. the output of Enc is probabilistic. For all its other requirements, see [KL14].

that is, the IV iv is obtained from $\mathsf{Mac}^{\mathsf{IV}}_{k_A}$, while the tag τ from $\mathsf{Mac}^{\mathsf{TAG}}_{k_A}$. There are three possible type compositions modes, with $C = \mathsf{AEnc}_K(N, A, M)$ with $K = (k_E, k_A)$:

E&M Encrypt-&-MAC where $C = (c\|\tau)$, $c = \mathsf{Enc}^{\mathsf{iv}}_{k_E}(M)$, $\mathsf{iv} = \mathsf{Mac}^{\mathsf{IV}}_{k_A}(N|U, A|U, M|U)$ and $\tau = \mathsf{Mac}^{\mathsf{TAG}}_{k_A}(N|U, A|U, M|U)$. (With $X|U$, we denote that the input either contains the string X or is absent).

EtM Encrypt-then-MAC, where $C = (c\|\tau)$, $c = \mathsf{Enc}^{\mathsf{iv}}_{k_E}(M)$, $\mathsf{iv} = \mathsf{Mac}^{\mathsf{IV}}_{k_A}(N|U, A|U, M|U)$ and $\tau = \mathsf{Mac}^{\mathsf{TAG}}_{k_A}(N|U, A|U, C)$.

MtE MAC-then-Encrypt , where $C = c$, $c = \mathsf{Enc}^{\mathsf{iv}}_{k_E}(m)$, with $m = M\|\tau$, $\mathsf{iv} = \mathsf{Mac}^{\mathsf{IV}}_{k_A}(N|U, A|U, M|U)$ and $\tau = \mathsf{Mac}^{\mathsf{TAG}}_{k_A}(N|U, A|U, C)$.

These are the so called A-modes.

In general, we can suppose that the IV is public and it is sent with C. This can speed decryption (anyway, we can check if the IV is correct). The fact that the IV is public follows from [NRS14]'s description.

When we compose a MAC with an nE scheme, then, we have the following types of composition modes, $C = \mathsf{AEnc}_K(N, A, M)$ with $K = (k_E, k_A)$:

E&M Encrypt-&-MAC where $C = (c\|\tau)$, $c = \mathsf{Enc}^{N}_{k_E}(M)$, $\tau = \mathsf{Mac}^{\mathsf{TAG}}_{k_A}(N|U, A|U, M|U)$.

EtM Encrypt-then-MAC, where $C = (c\|\tau)$, $c = \mathsf{Enc}^{N}_{k_E}(M)$, and $\tau = \mathsf{Mac}^{\mathsf{TAG}}_{k_A}(N|U, A|U, C)$.

MtE MAC-then-Encrypt , where $C = c$, $c = \mathsf{Enc}^{N}_{k_E}(m)$, with $m = M\|\tau$, and $\tau = \mathsf{Mac}^{\mathsf{TAG}}_{k_A}(N|U, A|U, C)$.

These are the so-called N-modes.

Note that both AEnc and Enc use the same nonce.

Thus, there are 160 possible modes when we use an ivE scheme and 20 possible modes when we use a nE scheme.

3.2 The four elusive modes: A10,A11,A12,N4.

Namprempre et al. [NRS14] were able to prove the security of 9 modes for ivE-composition and 3 for nE-composition, and the insecurity of all others except for 4 modes, all MAC-then-Encrypt type:

A10 MtE with $\mathsf{MAC}^{\mathsf{IV}}(N, A, U)$ and $\mathsf{MAC}^{\mathsf{TAG}}(U, A, M)$.

A11 MtE with $\mathsf{MAC}^{\mathsf{IV}}(N, A, U)$ and $\mathsf{MAC}^{\mathsf{TAG}}(U, U, M)$.

A12 MtE with $\mathsf{MAC}^{\mathsf{IV}}(N, U, U)$ and $\mathsf{MAC}^{\mathsf{TAG}}(U, A, M)$.

N4 MtE with $\mathsf{MAC}^{\mathsf{TAG}}(U, A, M)$.

We have depicted them in Fig. 1.

For decryption either the IV is sent in clear and it is checked and used for decryption, or it is recomputed from $(N, A|U)$.

Knowledge-of-Tag based security. Namprempre et al. [NRS14] proved that modes A10, A11, and A12 are secure if the ivE-scheme is Knowledge-of-Tag-secure

(KOT). In the KOT-experiment, "*knowing* a tag is captured by introducing a plaintext extractor Ext, a deterministic algorithm that takes as input all the inputs explicitly available to the forging adversary and outputs a string x or $\perp$" [NRS14]. Roughly speaking, a scheme is KOT-secure, if the adversary cannot " produce a forgery that uses an old $\mathsf{iv}^* = \mathsf{iv}_j$ and an old $m^* \| \tau^* = m_i \| \tau_i$, for which it [the adversary] does not (explicitly) know τ_i, and yet the extractor fails to determine this $m_i \| \tau_i$. Loosely speaking if the forger wins the KOT game, it has done so without (extractable) knowledge of the tag τ_i" [NRS14]. We depict the experiment in Tab. 1 in App. A.
It was left open the problem of whether ivE-security implies KOT.

Partial results on these modes [BPP18a]. At Indocrypt18, Berti et al. [BPP18a] proved some results about these modes: 1) mode N4 is insecure (using an nE-scheme which expands the ciphertext), 2) modes A10, A11, and A12 are either all secure or insecure, 3) modes A10, A11, A12 are secure if the IV scheme used is either misuse resistant or "message-malleable". On the other hand, if the ivE scheme used is either stateful or untidy, the modes are not secure. Here, we give some insights into these results.

Mode N4 is insecure. Berti et al. [BPP18a] provides a counterexample using the nE scheme Π (detailed in App. B in Alg. 3). Π has a key composed of two components $k_E = (k, v^*)$ where k is a key for a TBC with n-bit block, and v^* is a n-bit random string.
For the encryption Π proceeds as follow: the first ciphertext block c_0 is a pseudorandom value, except if the nonce is 1. In this case $c_0 = v^*$, where v^* is a secret random value; all others ciphertext block (except the last) are computed as $c_i = \mathsf{F}_k^{i,0}(N) \oplus m_i$, the last ciphertext block is computed as $c_l = \mathsf{F}_k^{l,0}(N) \oplus m_l$, except if the nonce is either 1 or 2 and the second to last message block m_{l-1}, is v^*: in this case, $c_l = \mathsf{F}_k^{l,1}(0) \oplus m_l$. That is, m_l is encrypted in the same way with both $N = 1$ and $N = 2$ in the case $m_{l-1} = v^*$.
We leave the proof that this scheme is nE-secure to the original paper [BPP18a], [3] as well with the description when the length of the message is not a multiple of n.
Observe that the ciphertext is n-bit longer than the message since there is the block c_0. We can see v^* as the trigger of a trojan which forces the same block to be encrypted in the same way in two different encryption queries.

The forgery against N4, when Enc is implemented with Π is straightforward:
- Authenticated encrypt $(1, A, M)$ with $M = M_1, ..., M_l$, obtaining C. Note that $C_0 = v^*$.
- Authenticated encrypt $(2, A, M^1)$ with $M^1_{l-1} = v^*$ and $|M| = |M^1|$, obtaining C^1.

[3] The only problem is if the adversary can do an encryption query (N, m) with $N = 1$ and $m_{l-1} = v^*$, but this cannot happen since v^* is random and leaked only during a query with $N = 1$.

– The forgery is $(1, A, C^*)$ with $C_0^* = v^*$, $C_i^* = C_i \oplus M_i \oplus M_i^1$ for $i = 1, ..., l$, and $C_{l+1}^* = C_{l+1}^1$ [we remind that C_{l+1}^* encrypts the tag in N4 when Enc is Π]. Note that $\mathsf{ADec}(1, A, C^*) = M^1$.

The forgery is correct (we leave the easy proof to the original paper).

Equivalent security among modes A10, A11 and A12. In the same paper, Berti et al. [BPP18a] proved that modes A10, A11, and A12 are either all secure or all insecure. First, they proved that all forgeries (except with negligible probability) must use an IV iv and a tag τ already computed. Then, they prove that in this case (reusing an iv and a τ) if an adversary can create a forgery against one of these modes, she can easily create a forgery against the other two modes. The main ingredients of this last step are these:

– *A12 secure ⇒ A10 secure:* Since the nonce N cannot be repeated during encryption queries, the adversary cannot distinguish if $\mathsf{iv} = \mathsf{MAC}^{\mathsf{IV}}_{k_A}(N)$ or $\mathsf{iv} = \mathsf{MAC}^{\mathsf{IV}}_{k_A}(N, A)$.
– *A11 secure ⇒ A10 secure:* Encrypt with A11 $M' = \mathsf{H}(A) \| M$, with H a hash function. Use an ivE scheme for A11 s.t. the encryption of $\mathsf{H}(A)$ is independent from the one of M, e.g., $\mathsf{Enc}'_{k_E}(\mathsf{iv}, M') = \mathsf{f}(k_E, \mathsf{iv}) \oplus \mathsf{H}(A) \| \mathsf{Enc}_{k_E}(\mathsf{iv}, M)$, where f is a random function $\mathsf{f} : \{0,1\}^{2n} \rightarrow \{0,1\}^n$.
– *A10 secure ⇒ A12 secure:* We use the same idea as before, encrypting with A12 $M' = \mathsf{H}(A) \| M$.
– *A10 secure ⇒ A11 secure:* We use a similar idea, but here we modify the nonce. The nonce used for A10 is N, while for A11 is $N' = N \| \mathsf{H}(A)$.

We leave the full details to the original paper [BPP18a] and its extended version [BPP18b].

Partial security/unsecurity results. Finally, in the same paper [BPP18a], the authors proved that modes A10, A11 and A12 are secure if the ivE scheme is either "*misuse resistant*" (that is, an adversary has no advantage if she can reuse the same IV during encryption queries [4]) or *message-malleable* (that is, if an adversary receives the decryption, different from $\perp$, of (iv, c), she can correctly decrypt (iv, c') $\forall c'$, as for example CTR, Counter mode [KL14].)

On the other hand, if the ivE scheme is not tidy or stateful, then the adversary can create a forgery against modes A10, A11, and A12 when implemented with certain ivE schemes (for the stateful case, we can use a variant of the scheme used against N4). We leave the details to the original paper [BPP18a] and its extended version [BPP18b].

4 The modes A10, A11, A12 are insecure

Now, we show that mode A12 is insecure, giving a counterexample. Thanks to [BPP18a], this means that also modes A11 and A10 are not secure.

[4] Note that this misuse-resistant definition is weaker then the standard one (see [RS06] for the original definition), where the adversary can do also decryption queries.

The first natural idea is to start from the counterexample against N4 and try to adapt it to the A12 case. But this is impossible because the iv is random, and the adversary does not choose it. Thus, if too many IVs reveal v^* or for which the last block is encrypted differently, the scheme is no more ivE-secure. On the other hand, with too few such IVs, the forgery may be done only with negligible probability.
Thus, we need a different idea.

4.1 Warming up - suppose that ivE outputs the IV

We start considering the case when the ivE scheme reveals the IV it used during the encryption queries. Note that in mode A12, the AE scheme does not need to reveal the IV since it can be correctly computed even by the decryption oracle ($\mathsf{iv} = \mathsf{MAC}^{\mathsf{IV}}_{k_A}(N)$). But, following the original paper, we assume that the IV is revealed. This follows also from the KOT definition [NRS14].

Construction. We propose an ivE-scheme Π_1 which is based on a TBC F and whose key k_E is the key k of the TBC.
If the message is s.t. the first two blocks are different, (that is, $m_1 \neq m_2$), substantially it is a TBC-based version of CTR, that is $c_i = \mathsf{F}^{1,i}_k(\mathsf{iv}) \oplus m_i$ with the difference that the last block (the one carrying the tag in A12) is encrypted with a slightly different tweak, that is, $c_l = \mathsf{F}^{2,l}_k(\mathsf{iv}) \oplus m_l$. Instead, if the first two message blocks are equal, the encryption is the same except for the two last ciphertext blocks: the second-to-last ciphertext block is obtained as $c_{l-1} = \mathsf{F}^{2,m_3}_k(\mathsf{iv}) \oplus \mathsf{F}^{2,m_3}_k(m_1) \oplus m_{l-1}$, while the last block is obtained as $c_l = \mathsf{F}^{1,l}_k(\mathsf{iv}) \oplus m_l$. The details are in Alg. 1.

The idea is that if $m_1 = m_2$, we are encrypting the second to last block (not the last block because it carries the tag that it is not known by an adversary, differently from the message that she has chosen to encrypt) in a secure way. Still, it reveals the information necessary to forge using previous encryptions. Note that if the adversary asks for an encryption of a message M with block-length $l-1$, she receives the iv used to encrypt and a ciphertext C. Now, if she asks to encrypt a second message M' s.t. it has block-length $l' = l+1$, $M_1 = M_2 = \mathsf{iv}$, and $M_3 = l+1$, she receives C', where a random iv' is used. C_{l+1} and C'_{l+1} reveal the crucial information for the forgery:

$$C_{l+1} \oplus C'_{l+1} \oplus M'_{l+1} = \mathsf{F}^{2,l+1}_k(\mathsf{iv}) \oplus m_{l+1} \oplus \mathsf{F}^{2,l+1}_k(\mathsf{iv}') \oplus \mathsf{F}^{2,l+1}_k(\mathsf{iv}) \oplus M'_{l+1} \oplus M'_{l+1} = m_{l+1} \oplus \mathsf{F}^{2,l+1}_k(\mathsf{iv}')$$

where $m = M \| \tau$, thus $m_i = M_i$ for $i = 1, ..., l$ and m_{l+1} is the tag τ of A12.

For simplicity, we have considered the case where all message has a length of a multiple of n with a minimum of $3n$. We can easily extend Π_1 to overcome these limitations.

Algorithm 1 The ivE encryption algorithm Π_1.

It uses a TBC $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^n \to \{0,1\}^n$ with $\mathcal{TW} = \{1,2\} \times \{0,1\}^n$

Gen:
- Return $k \xleftarrow{\$} \mathcal{K}$

$\mathsf{Enc}_k(\mathsf{iv}, m)$:
- Parse $m = m_1, ..., m_l$ with $|m_i| = n$
- For $i = 1, ..., l-2$
  - $c_i = \mathsf{F}_k^{1,i}(\mathsf{iv}) \oplus m_i$
- If $m_1 \neq m_2$
  - $c_{l-1} = \mathsf{F}_k^{1,l-1}(\mathsf{iv}) \oplus m_{l-1}$
  - $c_l = \mathsf{F}_k^{2,l}(\mathsf{iv}) \oplus m_l$
- Else
  - $c_{l-1} = \mathsf{F}_k^{2,m_3}(\mathsf{iv}) \oplus \mathsf{F}_k^{2,m_3}(m_1) \oplus m_{l-1}$
  - $c_l = \mathsf{F}_k^{1,l}(\mathsf{iv}) \oplus m_l$
- Return (iv, c) with $c = (c_1, ..., c_l)$

$\mathsf{Dec}_k(\mathsf{iv}, c)$:
- Parse $c = c_1, ..., c_l$ with $|c_i| = n$
- For $i = 1, ..., l-2$
  - $m_i = \mathsf{F}_k^{1,i}(\mathsf{iv}) \oplus c_i$
- If $m_1 \neq m_2$
  - $m_{l-1} = \mathsf{F}_k^{1,l-1}(\mathsf{iv}) \oplus c_{l-1}$
  - $m_l = \mathsf{F}_k^{2,l}(\mathsf{iv}) \oplus c_l$
- Else
  - $m_{l-1} = \mathsf{F}_k^{2,m_3}(\mathsf{iv}) \oplus \mathsf{F}_k^{2,m_3}(m_1) \oplus c_{l-1}$
  - $m_l = \mathsf{F}_k^{1,l}(\mathsf{iv}) \oplus c_l$
- Return $m = (m_1, ..., m_l)$

ivE-*security of* Π_1. The ivE-security of Π_1 is straightforward. It is easy to see that each ciphertext block is obtained XORing at least a call to F that has never been asked before, with the following exceptions:
- if two IVs are repeated, that is $\mathsf{iv}^i = \mathsf{iv}^j$;
- if iv^j is equal to m_1^i with $i \leq j$;

But both conditions happen with negligible probability since the IVs are randomly picked. Note that this the reason why there is a first component of the tweak that it is different for c_l (when $m_1 \neq m_2$), and c_{l-1} (when $m_1 = m_2$). Formally,

Theorem 1. *Let* F *be a* $(q_1, t, \epsilon_{\mathsf{tprp}})$-$\mathsf{tprp}$*, where the block-length is* n *bits, then* Π_1 *is* (q, t, ϵ)-ivE-*secure with*

$$\epsilon \leq \epsilon_{\mathsf{tprp}} + \frac{(\tilde{L}+2)(q+1)^2}{2^{n+1}},$$

where $q_1 = L + q$*, with* L *the total number of message blocks to be encrypted, and* $\tilde{L}$ *the maximal number of blocks in any message query.*

We leave the easy proof to App. C.1.

Forgery for A12 when the ivE-*scheme is* Π_1. The idea of the forgery is to ask the encryption of a message M s.t. $M_1 \neq M_2$ and then ask the encryption of a message M' s.t. $M'_1 = M'_2 = \mathsf{iv}^1$ which is at least a block longer than M. For the forgery, we proceed as follow:
- Ask the encryption of (N, A, M) with the message M s.t. $M_1 \neq M_2$ and it has l blocks. Obtain the ciphertext $C = (\mathsf{iv}, C_1, ..., C_l, C_{l+1})$. Π_1 encrypts $m = M \| \tau$ with $\tau = \mathsf{Mac}_{k_A}^{\mathsf{TAG}}(A, M)$ using as IV $\mathsf{iv} = \mathsf{Mac}_{k_A}^{\mathsf{IV}}(N)$.

– Ask the encryption of (N', A', M') with the message M' s.t. $M'_1 = M'_2 = \mathsf{iv}$, $M'_3 = l + 1$ and it has $l + 1$ blocks, and $N \neq N'$. Obtain the ciphertext $C' = (\mathsf{iv}', C'_1, ..., C'_l, C'_{l+1}, C'_{l+2})$.
– The forgery is (N^*, A^*, C^*) with $N^* = N'$, $A^* = A$ and C^* defined as follow:
  - $\mathsf{iv}^* = \mathsf{iv}'$;
  - $C^*_i = C'_i \oplus M'_i \oplus M_i$ for $i = 1, ..., l$;
  - $C^*_{l+1} = C_{l+1} \oplus C'_{l+1} \oplus M'_{l+1}$.

This is a valid forgery (encrypting M), as we formally prove in the next proposition:

Proposition 1. *Let Π_1 be the* ivE *scheme defined in Alg. 1. Let* MAC *be a* prf*-secure* MAC *with n-bit long output. Then the A12 composition is not* nAE*-secure.*

Proof. Observe that to break the nAE security (Def. 11) is enough to provide a valid forgery because, in the left world (AEnc, ADec), the answer will be different from the right world $(\$, \perp)$ which is always invalid.
Now, we have to prove that the forgery just described is *fresh* and *valid*.
We use the same notation as in the previous paragraph.
The fact that (N^*, A^*, C^*) is fresh is trivial since with nonce N^*, we have obtained only a ciphertext C', which is one block longer.
For validity, we start observing that we have never repeated a nonce. Now, we want to prove that $\mathsf{ADec}(N^*, A^*, C^*) = M$. To do this we compute $\tilde{C} = \mathsf{AEnc}(N', A, M)$:
– $\tilde{\mathsf{iv}} := \mathsf{MAC}^{\mathsf{IV}}(N')$. Thus, $\tilde{\mathsf{iv}} = \mathsf{iv}' = \mathsf{iv}^*$;
– For $i = 1, ..., l-2$, $\tilde{C}_i = \mathsf{F}^{1,i}_k(\mathsf{iv}^*) \oplus M_i = \mathsf{F}^{1,i}_k(\mathsf{iv}') \oplus M'_i \oplus M'_i \oplus M_i = C'_i \oplus M'_i \oplus M_i$ (and both M_i and M'_i are known by the adversary since she has chosen them).
– Since $M_1 \neq M_2$, then $\tilde{C}_l = \mathsf{F}^{1,l}_k(\tilde{\mathsf{iv}}) \oplus M_l = \mathsf{F}^{1,l}_k(\mathsf{iv}') \oplus M'_l \oplus M'_l \oplus M_l = C'_l \oplus M'_l \oplus M_l$ (and both M_i and M'_i are known by the adversary since she has chosen them). Note that C'_l is the third to last ciphertext block of C'. In fact, during the second encryption query the message encrypted by Π_1 is $m' = M' \| \tau' = M'_1 \| ... \| M'_l \| M'_{l+1} \| \tau'$.
– $\tilde{\tau} = \mathsf{MAC}^{\mathsf{TAG}}_{k_A}(A, M) = \tau$.
– $\tilde{C}_{l+1} = \mathsf{F}^{2,l+1}_k(\tilde{\mathsf{iv}}) \oplus \tilde{\tau} = \mathsf{F}^{2,l+1}_k(\tilde{\mathsf{iv}}) \oplus \mathsf{F}^{2,l+1}_k(\mathsf{iv}) \oplus M'_{l+1} \oplus \mathsf{F}^{2,l+1}_k(\mathsf{iv}) \oplus \tilde{\tau} \oplus M'_{l+1} = \mathsf{F}^{2,l+1}_k(\mathsf{iv}') \oplus \mathsf{F}^{2,M'_3}_k(M'_1) \oplus M'_{l+1} \oplus \mathsf{F}^{2,l+1}_k(\mathsf{iv}) \oplus \tau \oplus M'_{l+1} = C'_{l+1} \oplus C_{l+1} \oplus M'_{l+1}$, since $\tilde{\mathsf{iv}} = \mathsf{iv}'$, $M'_3 = l + 1$, $M'_1 = \mathsf{iv}$ and M'_{l+1} is known by the adversary (since chosen).

Thus, $\tilde{C} = C^*$ consequently $\mathsf{ADec}(N^*, A^*, C^*) = \mathsf{ADec}(N^*, A^*, \tilde{C}) = M$.

This and [BPP18a] proves that modes A10, A11 and A12 are not nAE-secure. Formally,

Theorem 2. *Let* MAC *be a* prf*-secure* MAC*. Then, there exist 3* ivE*-secure* ivE*-encryption schemes Π_{10}, Π_{11}, Π_{12} outputting the* IV *s.t. modes A10, A11 and A12 are not* nAE*-secure when implemented with* MAC *and the corresponding Π.*

Proof. For mode A12, the proof follows easily from the previous proposition, setting $\Pi_{12} := \Pi_1$ where the TBC has a block-length equal to the size of the MAC output. The proof that Π_{12} is ivE-secure is in Prop. 1.
For the other two cases, A10 and A11, in [BPP18a] it has been proved that a forgery against a mode A12 composition can be extended to a forgery to a mode A10 or A11 composition(see Sec. 3.2). This proves our statement.

As a side remark, it is easy to see that if in our forgery attack we had set $A' = A$, Π_1 is a good candidate as Π_{10} and Π_{11}. The details are provided in App. E.

This result also proves a domain separation between ivE and KOT. Formally,

Theorem 3. ivE-*secure* $\not\Rightarrow$ KOT-*secure.*

Proof. Π_1 is ivE-secure and not KOT-secure. The previous attack breaks the KOT-definition (App. A).

4.2 Broadcasting the IV in the ciphertext - Attack when the IV is hidden

In an interesting paper, Bellare et al. [BNT19] realized that sending the nonce along with the ciphertext can create security problems. Thus, they proposed a new syntax (NBE2) for AE scheme where the decryption oracle needs only as input the ciphertext and the header (and the key) to decrypt correctly.
Note that nonce-based encryption scheme and IV-based encryption scheme are syntactically equivalent (see Sec. 2.2), thus we can use their syntax also for ivE-scheme.
Since in the forgery attack we have presented in the previous section, we need that the adversary knows the IV used by Π_1 during the first authenticated encryption query, it is natural to wonder if it is enough to hide the IV used to prevent the previous attack and prove that modes A10, A11 and A12 are secure. Moreover, the IV is not needed to decrypt since it can be recomputed from N.
Unfortunately, this is not the case, as we prove in this section by providing a variant of Π_1, called Π_2 which can be forged even if the adversary has no clue about the IV used.

The construction Π_2. We add a block before all the ciphertext, called c_0. This block contains an encryption of the iv used ($c_0 = \mathsf{F}_k^{0,0}(\mathsf{iv})$). Now, even if the adversary cannot recover the iv from c_0, this pseudo-random block can be used in the forgery. Then, Π_2 is equal to Π_1 with the exception of c_{l-1} when $m_1 = m_2$. Instead of computing $c_{l-1} = \mathsf{F}_k^{2,m_3}(\mathsf{iv}) \oplus \mathsf{F}_k^{2,m_3}(m_1) \oplus m_{l-1}$, we compute $c_{l-1} = \mathsf{F}_k^{2,m_3}(\mathsf{iv}) \oplus \mathsf{F}_k^{2,m_3}(w) \oplus m_{l-1}$, with $w = \mathsf{F}_k^{-1,(0,0)}(m_1)$. Thus, with m_1, we can tell the encryption algorithm for which iv, that we do not know, we want some information.
Note that we can create a variant for the decryption that does not need IV as an input: Dec'. Dec' simply computes the iv as $\mathsf{iv} = \mathsf{F}_k^{-1,(0,0)}(c_0)$ and then proceeds as for Dec.

Algorithm 2 The ivE encryption algorithm Π_2.

It uses a TBC $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^n \to \{0,1\}^n$ with $\mathcal{TW} = \{0,1,2\} \times \{0,1\}^n$

Gen:
- Return $k \xleftarrow{\$} \mathcal{K}$

$\mathsf{Enc}_k(\mathsf{iv}, m)$:
- Parse $m = m_1, ..., m_l$ with $|m_i| = n$
- $c_0 = \mathsf{F}_k^{0,0}(\mathsf{iv})$
- For $i = 1, ..., l-2$
  - $c_i = \mathsf{F}_k^{1,i}(\mathsf{iv}) \oplus m_i$
- If $m_1 \neq m_2$
  - $c_{l-1} = \mathsf{F}_k^{1,l-1}(\mathsf{iv}) \oplus m_{l-1}$
  - $c_l = \mathsf{F}_k^{2,l}(\mathsf{iv}) \oplus m_l$
- Else
  - $w = \mathsf{F}_k^{-1,(0,0)}(m_1)$
  - $c_{l-1} = \mathsf{F}_k^{2,m_3}(\mathsf{iv}) \oplus \mathsf{F}_k^{2,m_3}(w) \oplus m_{l-1}$
  - $c_l = \mathsf{F}_k^{1,l}(\mathsf{iv}) \oplus m_l$
- Return $c = (c_0, c_1, ..., c_l)$

$\mathsf{Dec}_k(\mathsf{iv}, c)$:
- Parse $c = c_0, c_1, ..., c_l$ with $|c_i| = n$
- If $c_0 \neq \mathsf{F}_k^{0,0}(\mathsf{iv})$
  - Return $\perp$
- For $i = 1, ..., l-2$
  - $m_i = \mathsf{F}_k^{1,i}(\mathsf{iv}) \oplus c_i$
- If $m_1 \neq m_2$
  - $m_{l-1} = \mathsf{F}_k^{1,l-1}(\mathsf{iv}) \oplus c_{l-1}$
  - $m_l = \mathsf{F}_k^{2,l}(\mathsf{iv}) \oplus c_l$
- Else
  - $w = \mathsf{F}_k^{-1,(0,0)}(m_1)$
  - $m_{l-1} = \mathsf{F}_k^{2,m_3}(\mathsf{iv}) \oplus \mathsf{F}_k^{2,m_3}(w) \oplus c_{l-1}$
  - $m_l = \mathsf{F}_k^{1,l}(\mathsf{iv}) \oplus c_l$
- Return $m = (m_1, ..., m_l)$

ivE-*security of Π_2.* We have only to show that the modification that we have done does not affect security. In particular, c_0 is a pseudo-random block, and we need to use a stprp-secure F because we use F^{-1} during encryption, and we have a problem if w is equal to a previous iv.
Thus, we have that

Theorem 4. *Let* F *be a* $(q_1, t, \epsilon_{\mathsf{stprp}})$-stprp*, where the block-length is n bits, then Π_2 is* (q, t, ϵ)-ivE-*secure with*

$$\epsilon \leq \epsilon_{\mathsf{stprp}} + \frac{(\tilde{L} + 4)(q+1)^2}{2^{n+1}},$$

where $q_1 = L + 3q$, with L the total number of message blocks to be encrypted, and $\tilde{L}$ the maximal number of blocks in any message query.

We leave the easy proof to App. C.2.

Forgery for A12 when the ivE *scheme is Π_2.* It is easy to extend to forgery for mode A12 when implemented with Π_1 to mode A12 implemented with Π_2 as follow:
- Ask the encryption of (N, A, M) with the message M s.t. $M_1 \neq M_2$ and it has l blocks. Obtain the ciphertext $C = (C_0, C_1, ..., C_l, C_{l+1})$.
- Ask the encryption of (N', A', M') with the message M' s.t. $M'_1 = M'_2 = C_0$, $M'_3 = l + 1$ and it has $l + 1$ blocks, and $N \neq N'$. Obtain the ciphertext $C' = (C'_0, C'_1, ..., C'_l, C'_{l+1}, C'_{l+2})$.
- The forgery is (N^*, A^*, C^*) with $N^* = N'$, $A^* = A$ and C^* defined as follow:

- $C^*_0 = C'_0$;
- $C^*_i = C'_i \oplus M'_i \oplus M_i$ for $i = 1, ..., l$;
- $C^*_{l+1} = C_{l+1} \oplus C'_{l+1} \oplus M'_{l+1}$.

This is a valid forgery (encrypting M). Formally,

Proposition 2. *Let Π_2 be the* ivE *scheme defined in Alg. 1. Let* MAC *be a* prf-*secure* MAC *with n-bit long output. Then the A12 composition is not* nAE-*secure.*

The proof is the same as for Prop. 1 with the difference that we have to replace in the computation of $\tilde{C}_{l+1}$, $\mathsf{F}_k^{2,M'_3}(M'_1)$ with $\mathsf{F}_k^{2,M'_3}(w')$ where

$$w' = \mathsf{F}_k^{-1,(0,0)}(C^*_0) = \mathsf{F}_k^{-1,(0,0)}(M'_0) = \mathsf{F}_k^{-1,(0,0)}\left(\mathsf{F}_k^{(0,0)}(\mathsf{iv})\right) = \mathsf{iv}.$$

This and [BPP18a] proves that modes A10, A11 and A12 are not nAE-secure even if the IV is not broadcast. Formally,

Theorem 5. *Let* MAC *be a* prf-*secure* MAC. *Then, there exist 3* ivE-*secure* ivE-*encryption schemes Π_{10}, Π_{11}, Π_{12} s.t 1) they do not output the* IV, *2) the composition of Π_i with a* prf-*secure* MAC *according to mode A i is not* nAE-*secure for $i = 10,\ 11,\ 12$.*

The proof is the same as for Thm. 2.

Note that this attack proves that ivE-security does not imply Knowledge-of-Tag secure.

4.3 Fixed length nE scheme for N4

Finally, we prove that it is unnecessary to use an nE encryption scheme whose ciphertext is longer than plaintext to prove that N4 is not secure. We propose two constructions: one which is a modified version of Π_1 (Alg. 1) and another is a version of the scheme of [BPP18a].

Π_3, a variant of Π_1. The first idea is to use Π_1 directly since ivE-schemes and nE-schemes are syntactically equivalent.
Unfortunately, Π_1 is not nE-secure. It is trivial to see that the condition iv^i equal to m^j for $j \leq i$ does not happen with negligible probability since the IV is replaced with a nonce which the adversary chooses.
Thus, we modify Π_1, obtaining Π_3 as follows:

- the condition if $m_1 \neq m_2$ becomes $m_1 \neq m_2 \wedge N \neq 2$
- in the else we replace $c_{l-1} = \mathsf{F}_k^{2,m_3}(\mathsf{iv}) \oplus \mathsf{F}_k^{2,m_3}(m_1) \oplus m_{l-1}$ with $c_{l-1} = \mathsf{F}_k^{2,m_3}(N) \oplus \mathsf{F}_k^{2,m_3}(1) \oplus m_{l-1}$

The idea is that we always enter in the if except when the nonce $N = 2$. When we do not enter in the if, we obtain information to obtain a forgery combined with the information given by an encryption with $N = 1$.
It is easy to see that Π_3 is nE secure: If we do not enter in the else, Π_3 is secure. If we enter in the else we observe that c_{l-1} when encrypted with $N = 2$, and c_l when $N = 1$ are independently. We describe formally Π_3 in Alg. 4 in App. D.

Π_4 a variant of [BPP18a] Π_4 is obtained from the nE scheme described in Alg. 3 with these modifications:
- we remove v^* and c_0.
- the if condition becomes if $(N = 1 \vee N = 2) \wedge m_2 = \mathsf{F}_k^{1,0}(1) \oplus m_1$

To enter the if condition during encryption twice, it is necessary to guess $\mathsf{F}_k^{1,0}(1)$ before it is computed. We describe formally Π_4 in Alg. 5 in App. D and the forgery is detailed in App. F.

5 Conclusions

We have proved that modes A10, A11, and A12 are not secure in general. This concludes the classification of [NRS14].
Note that our results *do not imply* that all schemes obtained using mode N4, A10, A11, and A12 composition are insecure. Instead, these modes seem insecure only when implemented with artificial schemes, while they are secure when implemented with "natural" schemes. But, these compositions need ad-hoc proofs and cannot rely on general proof.
Finally, this work gives some insights into the limitation of indistinguishability from randomness. That is, having a random ciphertext encrypting the tag may not be enough to make it not usable for forgeries.
Acknowledgements: This work was partly supported by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. F. Berti was partly funded by the Israel Science Foundation (ISF) grant 2569/21.

References

[AFL16] Farzaneh Abed, Christian Forler, and Stefan Lucks. General classification of the authenticated encryption schemes for the CAESAR competition. *Comput. Sci. Rev.*, 22:13–26, 2016.

[BDH+17] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: parallel permutation-based cryptography. *IACR Trans. Symmetric Cryptol.*, 2017(4):1–38, 2017.

[BDJR97] 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.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.

[Ber14] Dan J Bernstein. Caesar call for submissions, final. Technical report, 2014.

[BMPS21] Olivier Bronchain, Charles Momin, Thomas Peters, and François-Xavier Standaert. Improved leakage-resistant authenticated encryption based on hardware AES coprocessors. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):641–676, 2021.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In 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*, pages 531–545. Springer, 2000.

[BN08] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *J. Cryptol.*, 21(4):469–491, 2008.

[BNT19] Mihir Bellare, Ruth Ng, and Björn Tackmann. Nonces are noticed: AEAD revisited. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 235–265. Springer, 2019.

[BPP18a] Francesco Berti, Olivier Pereira, and Thomas Peters. Reconsidering generic composition: The tag-then-encrypt case. In Debrup Chakraborty and Tetsu Iwata, editors, *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings*, volume 11356 of *Lecture Notes in Computer Science*, pages 70–90. Springer, 2018.

[BPP18b] Francesco Berti, Olivier Pereira, and Thomas Peters. Reconsidering generic composition: the tag-then-encrypt case. *IACR Cryptol. ePrint Arch.*, page 991, 2018.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In 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*, pages 317–330. Springer, 2000.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight authenticated encryption and hashing. *J. Cryptol.*, 34(3):33, 2021.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44. Springer, 2015.

[JZK+22] Mohamud Ahmed Jimale, Muhammad Reza Z'aba, Miss Laiha Mat Kiah, Mohd Yamani Idna Bin Idris, Norziana Jamil, Moesfa Soeheila Mohamad, and Mohd Saufy Rohmad. Authenticated encryption schemes: A systematic review. *IEEE Access*, 10:14739–14766, 2022.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. CRC Press, 2014.

[Kra01] 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.

[LRW02] Moses D. 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.

[NIS18] NIST. Submission requirements and evaluation criteria for the lightweight cryptography standardization process. Technical report, 2018.

[NIS21] NIST. Lightweight cryptography - finalists. Technical report, 2021.

[NRS14] 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.

[PS16] 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.

[RBBK01] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. In Michael K. Reiter and Pierangela Samarati, editors, *CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001*, pages 196–205. ACM, 2001.

[Rog02] Phillip Rogaway. Authenticated-encryption with associated-data. In Vijayalakshmi Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, pages 98–107. ACM, 2002.

[RS06] 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.

A Knowledge-of-Tag (KOT)

We describe the KOT experiment in Tab. 1.
For an ivE encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, a plaintext extractor Ext, a tag length T and a tag-input selection function $\mathcal{T}_{sel}$, we write

$$\mathsf{Adv}^{\mathsf{KOT}}_{\Pi,\mathsf{Ext},\mathcal{T}_{sel},T}(\mathsf{A}) := \Pr[\mathsf{KOT}_{\Pi,\mathsf{Ext},\mathcal{T}_{sel},T}(\mathsf{A}) = 1],$$

for the KOT-advantage of A.
Note that a plaintext extractor is a deterministic algorithm that takes as input all the inputs explicitly available to the forging adversary and outputs a string or $\perp$, "invalid" [NRS14].

$\mathsf{KOT}_{\Pi,\mathsf{Ext},\mathcal{T}_{sel},T}(\mathsf{A})$: $i \leftarrow 0$; $\mathsf{win} \leftarrow 0$ $K \xleftarrow{\$} \mathcal{K}$ Run $\mathsf{A}^{\mathsf{Enc},\mathsf{Reveal},\mathsf{Test}}$ Return win Oracle $\mathsf{Reveal}(j)$: $\mathcal{T} \leftarrow \mathcal{T} \cup \{(j, \tau_j)\}$ Return τ_j	Oracle $\mathsf{AEnc}(N, A, M)$: $i \leftarrow i + 1$ $(N_i, A_i, M_i) \leftarrow (N, A, M)$ $\mathsf{iv}_i \xleftarrow{\$} \{0,1\}^n$ $\tau_i \leftarrow \mathcal{T}_{sel}(N_i, A_i, M_i)$ if $T[S_i] = \perp$, then $T[S_i] \xleftarrow{\$} \{0,1\}^T$ $\tau_i \leftarrow T[S_i]$ $X_i \leftarrow M_i \Vert \tau_i$ $C_i \leftarrow \mathsf{Enc}_k(\mathsf{iv}_i, X_i)$ $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(i, \mathsf{iv}_i, M_i, C_i)\}$ Return (iv_i, C_i)	Oracle $\mathsf{Test}(j^*, C^*)$: $X \leftarrow \mathsf{Ext}(j^*, C^*, \mathcal{Q}, \mathcal{T})$ $\mathsf{valid} \leftarrow \mathsf{xgood} \leftarrow 0$ if $\exists X_i$ s.t. a) $C^* = \mathsf{Enc}_K(\mathsf{iv}_{j^*}, X_i)$ and b) $(\cdot, \tau_i) \notin \mathcal{T}$ and c) $X_i = X_{j^*}$ then $\mathsf{valid} \leftarrow 1$ if $X = X_i$ then $\mathsf{xgood} \leftarrow 1$ if $\mathsf{valid} \wedge \mathsf{xgood}$ then $\mathsf{win} \leftarrow 1$ Return 1 Return 0

Table 1. The Knowledge-of-Tag (KOT) experiment [NRS14].

The attack detailed in Sec. 4.1, breaks the KOT-security since we never use the oracle Reveal, and when we ask the forgery to the oracle Test. This query makes the Ext output M and $\mathsf{win} \leftarrow 1$.

B Attack against N4

We describe the nE scheme proposed by Berti et al. [BPP18a] to prove that N4 is not secure in Alg. 3.

C ivE-security of Π_1 and Π_2

C.1 Π_1 is ivE-secure

Here we prove Thm. 1.

Algorithm 3 The nE encryption algorithm Π [BPP18a].

It uses a TBC $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^n \to \{0,1\}^n$ with $\mathcal{TW} = \{1,2\} \times \{0,1\}^n$

Gen:
- $k \xleftarrow{\$} \mathcal{K}$
- $v^* \xleftarrow{\$} \{0,1\}^n$
- Return $k_E = (k, v^*)$

$\mathsf{Enc}_{k_E}(N, m)$:
- Parse $m = (m_1, ..., m_l)$ with $|m_i| = n$
- If $N = 1$
  - $c_0 = v^*$
- Else
  - $c_0 = \mathsf{F}_k^{0,0}(N)$
- For $i = 1, ..., l-1$
  - $c_i = \mathsf{F}_k^{i,0}(N) \oplus m_i$
- If $(N = 1 \vee N = 2) \wedge m_{l-1} = v^*$
  - $c_l = \mathsf{F}_k^{l,1}(1) \oplus m_l$
- Else
  - $c_l = \mathsf{F}_k^{l,0}(N) \oplus m_l$
- Return $c = (c_0, ..., c_l)$

$\mathsf{Dec}_{k_E}(n, c)$:
- Parse $c = (c_1, ..., c_l)$ with $|c_i| = n$
- If $N = 1$
  - If $c_0 \neq v^*$
    - Return $\perp$
- Else
  - If $c_0 \neq \mathsf{F}_k^{0,0}(n)$
    - Return $\perp$
- For $i = 1, ..., l-1$
  - $m_i = \mathsf{F}_k^{i,0}(N) \oplus c_i$
- If $(N = 1 \vee N = 2) \wedge m_{l-1} = v^*$
  - $m_l = \mathsf{F}_k^{l,1}(1) \oplus c_l$
- Else
  - $m_l = \mathsf{F}_k^{l,0}(N) \oplus c_l$
- Return $m = (m_1, ..., m_l)$

Proof. Let Game 0 be the ivE-game where the adversary A has to distinguish Π_1 from a random scheme Π (outputting ciphertexts with the same length as Π_1). Let E_0 be the event that A wins this game.
Let Game 1 be Game 0 where we abort if two different IVs are equal, that is, there exists $i, j \in \{1, ..., q\}$ s.t. $\mathsf{iv}^i = \mathsf{iv}^j$. Let E_1 be the event that A wins this game.
Clearly $|\Pr[E_0] - \Pr[E_1]| \leq \Pr[B]$ where B is the event that 2 IVs collides. Since the IVs are uniformly randomly picked, due to the well known birthday bound (see, for example [KL14]), $\Pr[B] \leq \frac{q^2}{2^{n+1}}$.
Let Game 2 be Game 1 where we have replaced the tprp F with a random function. Let E_2 be the event that A wins this game.
Clearly $|\Pr[E_1] - \Pr[E_2]| \leq \epsilon_{\mathsf{tprp}} + \frac{\tilde{L}q^2}{2^{n+1}}$. This is proved in two steps:
a) we replace F with a tweakable random permutation $\tilde{\mathsf{f}}$. We observe that we need at most $(l+1)$ call to F per encryption query (only for c_{l-1} two calls may be needed), thus in total we need to do at most $L + q$ queries to F and the running time is the same (there are no other primitives involved).
b) we replace $\tilde{\mathsf{f}}$ with a random function f. To use tightly the well-known result that a random permutation is a random function (they can be distinguished with probability $\leq \frac{Q^2}{2^{n+1}}$ when queried at most Q times, see [KL14], for example), we observe that for each possible tweak there are at most $2q$ different inputs. This is obvious for the tweaks $(1, i)$, there is at most one call for each query. Instead for the tweak of type $2, j$, we can have at most 2 calls for each query. Thus, there

at most $\tilde{L}$ possible tweaks for which this happens.[5]
Game 3 is Game 2 where we abort if iv^j is equal to m_1^i with $i \leq j$. Let call C this event. Let E_3 be the event that A wins this game.
Clearly $|\Pr[E_3] - \Pr[E_2]| \leq \Pr[C]$. To bound $\Pr[C]$, we call C_j the event that $\mathsf{iv}^j = m_1^i$ for $i \leq j$. Clearly $\Pr[C] \leq \sum_{j=1}^{q} \Pr[C_j]$ and $\Pr[C_j] = \frac{j}{2^n}$. Thus, $\sum_{j=1}^{q} \Pr[X_j] = \frac{1}{2^n} \sum_{j=1}^{q} (j) = \frac{q(q+1)}{2^{n+1}} \leq \frac{(q+1)^2}{2^{n+1}}$.
Finally, we observe that the probability that A wins Game 3 is 0 since for all message blocks except c_{l-1}^j and c_l^j ($\forall j = 1, ..., q$), $c_i^j = \mathsf{f}^{1,i}(\mathsf{iv}^j) \oplus m_i^j$ is indistinguishable from random ciphertext blocks since f is a random function and $\mathsf{f}^{1,i}(\mathsf{iv}^j)$ has never been computed before (the IVs are all different). For c_{l-1}^j if $m_1^j \neq m_2^j$, the previous argument holds. Instead, if $m_1^j = m_2^j$ since iv^j is different from all previous IVs and $m_1^{j'}$ for all $j' \leq j$, $\mathsf{f}^{2,m_3}(\mathsf{iv}^j)$ has never been computed before, thus, we can reuse the previous argument. Similarly, for the last ciphertex block c_l^j, we have that if $m_1^j \neq m_2^j$, $\mathsf{f}^{2,l}(\mathsf{iv}^j)$ has never been computed before due to the non collision of IVs and the event C_j, while if $m_1^j = m_2^j$, we can reuse easily a previous argument.
Summing up everything we obtain the thesis. Thus,

$$\Pr[E_0] \leq \frac{q^2}{2^{n+1}} + \epsilon_{\mathsf{tprp}} + \frac{\tilde{L}q^2}{2^{n+1}} + \frac{(q+1)^2}{2^{n+1}} \leq \epsilon_{\mathsf{tprp}} + \frac{(\tilde{L}+2)(q+1)^2}{2^{n+1}}.$$

C.2 Π_2 is ivE-secure

Here we prove Thm. 4.

Proof. The proof follows the proof of Thm. 1 with the following difference:
When we do the transition between Game 2 and Game 1, when we replace the stprp F with a random tweakable permutation $\tilde{\mathsf{f}}$, we need at most $l+3$ queries to F and its inverse (or to $\tilde{\mathsf{f}}$). Thus, in total we need $L + 2q$ queries to $(\mathsf{F}, \mathsf{F}^{-1})$ or to $(\tilde{\mathsf{f}}, \tilde{\mathsf{f}}^{-1})$. Then, we want to replace $\tilde{\mathsf{f}}^{-1}$ with a random permutation $\tilde{\mathsf{g}}$ except when $\tilde{\mathsf{f}}^{-1}$ is previously defined. That is, $c_0^i = \tilde{\mathsf{f}}^{0,0}$, then if $m_1^j = c_0^i$, $\tilde{\mathsf{f}}^{-1,(0,0)}(m_1^j)$ is already defined. If it is not the case, instead of using $\tilde{\mathsf{f}}^{-1}$ we use $\tilde{\mathsf{g}}$. We observe that the adversary can distinguish the two situations if an iv picked is equal to a previous output of $\tilde{\mathsf{g}}$. We call this event D, and we call D_i the event that iv^i is equal to w^j with $j < i$. Clearly the replacement of $\tilde{\mathsf{f}}^{-1}$ with $\tilde{\mathsf{g}}$ is undetectable

[5] Observe that for the second case, since the adversary can do at most 2 queries with tweak $2, m_3$ per encryption query, if she uses different $2, m_3$ tweaks in different queries, then, the total number of queries which can result in collision for f and $\tilde{\mathsf{f}}$ remains the same, but the bound is different since it is $\sum_{j \in \{0,1\}^n} \frac{Q(j)^2}{2^{n+1}}$ with $Q(j)$ the number of queries asked with tweak $2, j$. Note that $\sum_{j \in \{0,1\}^n} Q(j) = 2q$ and $Q(j) \geq 0$. It is easy to see that the max for the bound for all possible distribution of $Q(j)$ is $\frac{4q^2}{2^{n+1}}$.

if event D does not happen, which means $\forall i = 1, ..., n$ that event D_i does not happen[6]. But $\Pr[D] \leq \sum_{i=1}^{q} \Pr[D_i] \leq \sum_{i=1}^{q} \frac{i-1}{2^n} = \frac{q(q-1)}{2^{n+1}}$.
Finally, when we replace $\tilde{\mathsf{f}}$ and $\tilde{\mathsf{g}}$ with two random functions f and g, we observe that we may have collision also when the tweak is $(0, 0)$, thus there are at most $\tilde{L} + 1$ possible tweaks for which a collision may happen.
For Game 3, instead of having the problem that $\mathsf{iv}^i = m_1^j$ for $j \leq i$, we have the problem if $\mathsf{iv}^i = w^j$ for $j \leq i$. The transition between Game 2 and 3 is the same. Finally, when we prove that $\Pr[E_3] = 0$, we have only to consider in the analysis of the case c_{l-1}^j if $m_1^j = m_2^j$ that we need to replace m_1 in the computation with w with $w = \mathsf{g}(m_1)$. Thus, putting everything together we obtain that

$$\Pr[E_0] \leq \frac{q^2}{2^{n+1}} + \epsilon_{\mathsf{stprp}} + \frac{q(q-1)}{2^{n+1}} + \frac{(\tilde{L}+1)q^2}{2^{n+1}} + \frac{(q+1)^2}{2^{n+1}} \leq \epsilon_{\mathsf{stprp}} + \frac{(\tilde{L}+4)(q+1)^2}{2^{n+1}}.$$

D New schemes against N4

We describe the algorithm Π_3 in Alg. 4 and Π_4 in Alg. 5.

Algorithm 4 The nE encryption algorithm Π_3.

It uses a TBC $\mathsf{F} : \mathcal{K} \times \mathcal{TW} \times \{0,1\}^n \to \{0,1\}^n$ with $\mathcal{TW} = \{1,2\} \times \{0,1\}^n$

Gen:
- $k \xleftarrow{\$} \mathcal{K}$

$\mathsf{Enc}_k(\mathsf{iv}, m)$:
- Parse $m = m_1, ..., m_l$ with $|m_i| = n$
- For $i = 1, ..., l-2$
  - $c_i = \mathsf{F}_k^{1,i}(n) \oplus m_i$
- If $m_1 \neq m_2 \wedge n \neq 2$
  - $c_{l-1} = \mathsf{F}_k^{1,l-1}(n) \oplus m_{l-1}$
  - $c_l = \mathsf{F}_k^{2,l}(n) \oplus m_l$
- Else
  - $c_{l-1} = \mathsf{F}_k^{2,m_3}(n) \oplus \mathsf{F}_k^{2,m_3}(1) \oplus m_{l-1}$
  - $c_l = \mathsf{F}_k^{1,l}(n) \oplus m_l$
- Return $c = (c_1, ..., c_l)$

$\mathsf{Dec}_k(n, c)$:
- Parse $c = c_1, ..., c_l$ with $|c_i| = n$
- For $i = 1, ..., l-2$
  - $m_i = \mathsf{F}_k^{1,i}(n) \oplus c_i$
- If $m_1 \neq m_2 \wedge n \neq 2$
  - $m_{l-1} = \mathsf{F}_k^{1,l-1}(n) \oplus c_{l-1}$
  - $m_l = \mathsf{F}_k^{2,l}(n) \oplus c_l$
- Else
  - $m_{l-1} = \mathsf{F}_k^{2,m_3}(n) \oplus \mathsf{F}_k^{2,m_3}(1) \oplus c_{l-1}$
  - $m_l = \mathsf{F}_k^{1,l}(n) \oplus c_l$
- Return $m = (m_1, ..., m_l)$

E Attacks against A10 and A11 using the scheme Π_1

We give the details of how we can use Π_1 as a counterexample for modes A10 and A11.

[6] To improve this result, we should consider that the computation of $\tilde{\mathsf{g}}$ is not always triggered and we can do a more detailed analysis, but this is not necessary.

Algorithm 5 The nE encryption algorithm Π_4.

It uses a TBC $\mathsf{F}: \mathcal{K} \times \mathcal{TW} \times \{0,1\}^n \to \{0,1\}^n$ with $\mathcal{TW} = \{1,2\} \times \{0,1\}^n$

Gen:
- Return $k \xleftarrow{\$} \mathcal{K}$

$\mathsf{Enc}_k(N, m)$:
- Parse $m = (m_1, ..., m_l)$ with $|m_i| = n$
- For $i = 1, ..., l-1$
  - $c_i = \mathsf{F}_k^{i,0}(N) \oplus m_i$
- If $(N = 1 \vee N = 2) \wedge m_2 = \mathsf{F}_k^{1,0}(1) \oplus m_1$
  - $c_l = \mathsf{F}_k^{l,1}(1) \oplus m_l$
- Else
  - $c_l = \mathsf{F}_k^{l,0}(N) \oplus m_l$
- Return $c = (c_1, ..., c_l)$

$\mathsf{Dec}_k(n, c)$:
- Parse $c = (c_1, ..., c_l)$ with $|c_i| = n$
- For $i = 1, ..., l-1$
  - $m_i = \mathsf{F}_k^{i,0}(N) \oplus c_i$
- If $(N = 1 \vee N = 2) \wedge m_2 = \mathsf{F}_k^{1,0}(1) \oplus m_1$
  - $m_l = \mathsf{F}_k^{l,1}(1) \oplus c_l$
- Else
  - $m_l = \mathsf{F}_k^{l,0}(N) \oplus c_l$
- Return $m = (m_1, ..., m_l)$

Forgery for mode A10 when the ivE*-scheme is* Π_1. The idea of the forgery is to ask the encryption of a message M s.t. $M_1 \neq M_2$ and then ask the encryption of a message M' s.t. $M'_1 = M'_2 = \mathsf{iv}^1$ which is at least a block longer than M. For the forgery, we proceed as follow:
- Ask the encryption of (N, A, M) with the message M s.t. $M_1 \neq M_2$ and it has l blocks. Obtain the ciphertext $C = (\mathsf{iv}, C_1, ..., C_l, C_{l+1})$. Π_1 encrypts $m = M \| \tau$ with $\tau = \mathsf{Mac}_{k_A}^{\mathsf{TAG}}(A, M)$ using as IV $\mathsf{iv} = \mathsf{Mac}_{k_A}^{\mathsf{IV}}(N, A)$.
- Ask the encryption of (N', A', M') with the message M' s.t. $M'_1 = M'_2 = \mathsf{iv}$, $M'_3 = l+1$ and it has $l+1$ blocks, and $N \neq N'$. Obtain the ciphertext $C' = (\mathsf{iv}', C'_1, ..., C'_l, C'_{l+1}, C'_{l+2})$.
- The forgery is (N^*, A^*, C^*) with $N^* = N'$, $A^* = A$ and C^* defined as follow:
  - $\mathsf{iv}^* = \mathsf{iv}'$;
  - $C^*_i = C'_i \oplus M'_i \oplus M_i$ for $i = 1, ..., l$;
  - $C^*_{l+1} = C_{l+1} \oplus C'_{l+1} \oplus M'_{l+1}$.

This is a valid forgery (encrypting M). The proof is the same as Prop. 1.

Forgery for mode A11 when the ivE*-scheme is* Π_1. The idea of the forgery is to ask the encryption of a message M s.t. $M_1 \neq M_2$ and then ask the encryption of a message M' s.t. $M'_1 = M'_2 = \mathsf{iv}^1$ which is at least a block longer than M. For the forgery, we proceed as follow:
- Ask the encryption of (N, A, M) with the message M s.t. $M_1 \neq M_2$ and it has l blocks. Obtain the ciphertext $C = (\mathsf{iv}, C_1, ..., C_l, C_{l+1})$. Π_1 encrypts $m = M \| \tau$ with $\tau = \mathsf{Mac}_{k_A}^{\mathsf{TAG}}(M)$ using as IV $\mathsf{iv} = \mathsf{Mac}_{k_A}^{\mathsf{IV}}(N, A)$.
- Ask the encryption of (N', A', M') with the message M' s.t. $M'_1 = M'_2 = \mathsf{iv}$, $M'_3 = l+1$ and it has $l+1$ blocks, and $N \neq N'$. Obtain the ciphertext $C' = (\mathsf{iv}', C'_1, ..., C'_l, C'_{l+1}, C'_{l+2})$.
- The forgery is (N^*, A^*, C^*) with $N^* = N'$, $A^* = A$ and C^* defined as follow:
  - $\mathsf{iv}^* = \mathsf{iv}'$;
  - $C^*_i = C'_i \oplus M'_i \oplus M_i$ for $i = 1, ..., l$;

- $C^*_{l+1} = C_{l+1} \oplus C'_{l+1} \oplus M'_{l+1}$.

This is a valid forgery (encrypting M). The proof is the same as Prop. 1.

F Forgery of mode N4 using Π_4

The forgery proceeds as follow:

- Ask the encryption of (N, A, M) with $N = 1$, $M = (M_1, M_2)$, with M_1, M_2 picked uniformly at random in $\{0,1\}^n$. Obtain the ciphertext $C = (C_1, C_2, C_3)$ with $C_i = \mathsf{F}_k^{i,0}(N) \oplus M_i$, for $i = 1, 2$. With probability equal to $1 - 2^n$, $M_2 \neq \mathsf{F}_k^{1,0}(1) \neq M_1$, thus, $C_3 = \mathsf{F}_k^{3,0}(N) \oplus \tau$ with $\tau = \mathsf{Mac}_{k_A}^{\mathsf{TAG}}(A, M)$ (since the last ciphertext block encrypts the tag).
- Ask the encryption of (N', A, M') with $N' = 2$, $M = (M'_1, M'_2)$, with $M'_1 = M_1$, and $M'_2 = C_1$, thus, $M'_2 = \mathsf{F}_k^{1,0}(N) \oplus M_1$. Obtain the ciphertext $C' = (C'_1, C'_2, C'_3)$ with $C'_i = \mathsf{F}_k^{i,0}(N') \oplus M'_i$, for $i = 1, 2$. Since $M'_2 = \mathsf{F}_k^{1,0}(1) \oplus M_1$, then, $C'_3 = \mathsf{F}_k^{3,1}(1) \oplus \tau'$, with $\tau' = \mathsf{Mac}_{k_A}^{\mathsf{TAG}}(A, M')$.
- The forgery is (N^*, A^*, C^*) with $N^* = 1$ $A^* = A$, and $C^* = C^*_1, C^*_2, C^*_3$, where $C^*_1 = C_1$, $C^*_2 = C_2 \oplus M_2 \oplus M'_2$, and $C^*_3 = C'_3$.

This is a valid forgery. In fact, if we consider an encryption of $(1, A, M')$, we obtain:

- $C^*_1 = \mathsf{F}_k^{1,0}(1) \oplus M_1 = C_1$
- $C^*_2 = \mathsf{F}_k^{2,0}(1) \oplus M'_2 = \mathsf{F}_k^{2,0}(1) \oplus M_2 \oplus M_2 \oplus M'_2 = C_2 \oplus M_2 \oplus M'_2$.
- $C^*_3 = \mathsf{F}^{3,1}(1) \oplus \tau' = C'_3$ since $N^* = 1$ and $M'_2 = C_1 = M'_2 = \mathsf{F}_k^{1,0}(N) \oplus M_1$

Thus, with probability $1 - 2^n$ (N^*, A^*, C^*) is a forgery, since it correctly encrypts $(1, A, M')$.