AET-LR: Rate-1 Leakage-Resilient AEAD based on the Romulus Family

Extended Abstract

Chun Guo[1], Mustafa Khairallah[2,3] and Thomas Peyrin[2,3]

[1] Shandong University, Shandong, China
201999900076@sdu.edu.cn
[2] Nanyang Technological University, Singapore, Singapore
[3] Temasek Laboratories@NTU, Singapore, Singapore
mustafam001@e.ntu.edu.sg,thomas.peyrin@ntu.edu.sg

Abstract. While lightweight cryptography has been a prevailing area of research in symmetric key cryptography over the past decade or more, leakage resilience of symmetric-key primitive have also been a steadily growing field of study in the recent years. However, these two design goals are usually opposed to each other, providing the designer with a tough trade-off. Most leakage-resilient modes require either multiple primitive (*e.g.* Spook), expensive re-keying functions (*e.g.* LRICB), slower designs (*e.g.* TEDT) or expensive countermeasures (*e.g.* PHOTON-BEETLE). In this paper, we propose a new AEAD design; AET-LR, which is a slight variant of the Romulus family of AEAD. It is purely based on Tweakable Block Ciphers (TBC), and requires only 1 TBC call per plaintext block, in addition to two heavily protected TBC calls per message. Hence, it has almost the same cost as Romulus. However, a crucial difference between AET-LR and Romulus is that the plaintext also affects the tweakey, making the TBC calls irreversible. This ensures that the mode offers Ciphertext Integrity with nonce Misuse and Leakage resistance with unbounded leakage (CIML2), up to $2^{n/2}$ attack complexity, and indistinguishability against Chosen Ciphertext Attacks with nonce misuse and leakage resilience with unbounded encryption leakage (CCAml1). These results also imply weaker security guarantees, including Integrity with Realease of Unverified Plaintext (INT-RUP) and Ciphertext Integrity with Misuse Resistance (MR-CINT).

Keywords: Romulus · leakage resilience · AEAD · AET-LR

1 Introduction

In this article, we introduce a *lightweight* Authenticated Encryption with Associated Data (AEAD) mode targeted towards secure practical implementations called AET-LR. The design is based on Tweakable Block Ciphers (TBC). Specifically, it can be seen as a slight adaptation of the Romulus-N [IKMP20a, IKMP20b] AEAD mode, following the same design philosophy, but introducing new ideas and design strategies to achieve leakage resilience. In short, the main difference with the Romulus-N mode is simply a feed-forward of the message block into the tweak input of the TBC calls. The philosophy of the design is to maintain the minimum lightweight performance for TBC:

(i) Optimal computational efficiency, *i.e.* rate-1 operation.

(ii) Minimum state size of a TBC mode, *i.e.* $(n + t + k)$-bit for n-bit block, t-bit tweak and k-bit key TBC.

Simultaneously, the design adopts the leveled implementation philosophy, where only the first and last TBC calls need to be heavily protected against physical attacks. Such implementations of AET-LR satisfy several misuse and leakage-based security notions, including *Ciphertext INTegrity nonce-Misuse Resistance* (MR-CINT), *INTegrity with the Release of Unverified Plaintext* (INT-RUP), *Ciphertext Integrity with Misuse and Leakage in the chosen-ciphertext model* (CIML2) and *indistinguishability against Chosen Ciphertext Attacks with nonce misuse and leakage resilience with unbounded encryption leakage* (CCAml1).

The design relies on two central ideas. The first is leveled implementations with a protected Key Derivation Function (KDF) and a protected Tag Generation Function (TGF). The master key is only used in these two functions. This idea is a standard practice where a variation of it is used in most modern leakage-resilient AEAD scheme [BGP+19, BBB+20]. It limits the exposure of the master key to only two function calls per message. These two function can be implemented using expensive countermeasures. The rest of the mode can be implemented using cheaper countermeasures or even unprotected implementations. The second idea is to design the main body of the mode as both a single-pass online serial TBC-based AEAD mode, and also a TBC-based collision-resistant and preimage-resistant hash function, where each ciphertext block can be viewed as a hash tag for not only the previous message and associated data blocks, but also the message key (the output of the KDF function). Hence, even if the adversary figures out the message key, it is not trivial to use such key to forge new messages.

First, we discuss the security of the hash function constructed using a Tweakable Block Cipher (TBC) with large tweakey space. The hash function is an extension of one of the hash functions proposed by Black *et al.* [BRS02], under the assumption that the TBC is secure in the *chosen-tweakey model*. Given this construction, we propose an AEAD design (Figure 1) which statisfies two interesting properties:

(i) Each ciphertext block acts as the tag of a collision/pre-image resistant hash function of all the previous plaintext and associated data blocks, even in the known-key model.

(ii) The temporary key corresponding to a given nonce is unique, allowing misuse and leakage resilience.

The rest of the article is organized as follows. In Section 2 we provide some preliminaries and definitions. In Section 3, we propose a TBC-based hash function that serves as the basis for our design. In Section 4, we study the security of our new design in the presence of leakage. We conclude in Section 5.

2 Preliminaries

In this section we provide useful notations, definitions and security notions.

2.1 General Notation

We denote by $\mathsf{F}(n)$ the set of all functions of domain $\{0,1\}^n$ and range $\{0,1\}^n$, by $\mathsf{P}(n)$ the set of all permutations on $\{0,1\}^n$, and by $\mathsf{BC}(k,2n)$ the set of all blockciphers with $2n$-bit block size and k-bit keys. Let $\{0,1\}^*$ be the set of all finite bit strings, including the empty string ε. For $X \in \{0,1\}^*$, let $|X|$ denote its bit length. Here $|\varepsilon| = 0$. For an integer $n \geqslant 0$, let $\{0,1\}^n$ be the set of n-bit strings, and let $\{0,1\}^{\mathsf{L}_{\mathsf{enc}}n} = \bigcup_{i=0,\ldots,n}\{0,1\}^i$, where $\{0,1\}^0 = \{\varepsilon\}$. Let $[\![n]\!] = \{1,\ldots,n\}$ and $[\![n]\!]_0 = \{0,1,\ldots,n-1\}$. Let $|X|_n = \max\{1, \lceil |X|/n \rceil\}$.

For two bit strings X and Y, $X \,\|\, Y$ is their concatenation. We also write this as XY if it is clear from the context. Let 0^i (1^i) be the string of i zero bits (i one bits), and for instance we write 10^i for $1 \,\|\, 0^i$. Bitwise XOR of two variables X and Y is denoted by

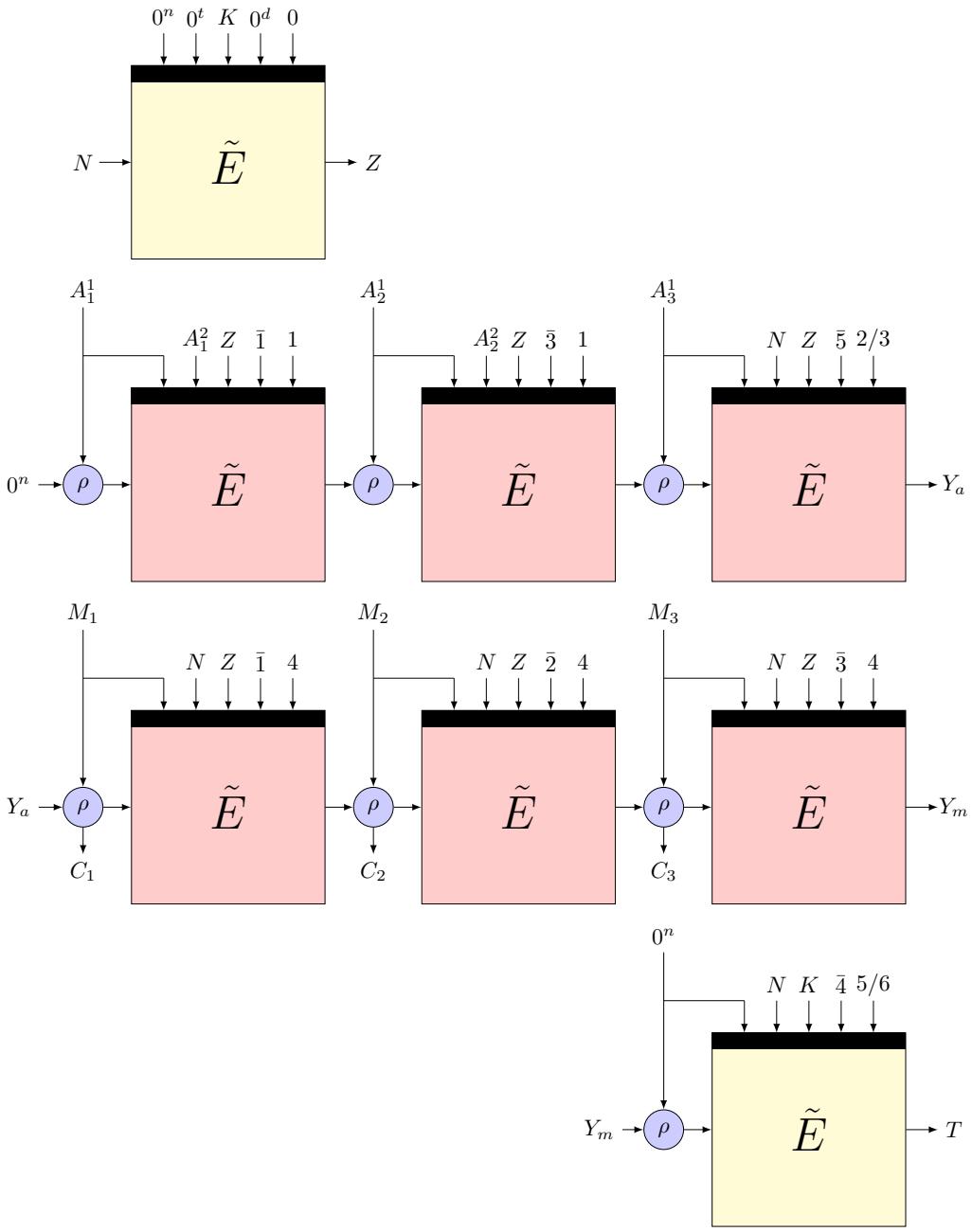

Figure 1: The AET-LR mode

$X \oplus Y$, where $|X| = |Y| = c$ for some positive integer c. For a binary string X of $|X| \geqslant x$, we write $\mathtt{lmt}_x(X)$ (resp. $\mathtt{rmt}_x(X)$) to denote the leftmost (resp. rightmost) x bits of X. Note that we do not use "MSB" and "LSB" for them, which is customary but depends on endianness.

2.2 Hash Functions in the Ideal Cipher Model

For any adversary A^E, the collision advantage of A against the hash function H^E is

$$\mathbf{Adv}^{\mathsf{cr}}_H(A) := \Pr\left[A^E \Rightarrow (m, m') : H^E(m) = H^E(m')\right].$$

We further define

$$\mathbf{Adv}^{\mathsf{cr}}_H(p) := \max_A \mathbf{Adv}^{\mathsf{cr}}_H(A),$$

where the maximum is defined over all adversaries A^E making at most p queries to E.

The definition of range oriented preimage resistance uniformly samples the target space q times instead: $(z_1, ..., z_q) \leftarrow \{0,1\}^n$, and requires finding a preimage for any of $(z_1, ..., z_q)$. Formally, for any adversary A^E, the range-oriented preimage advantage of A against the hash function H^E is

$$\mathbf{Adv}^{\mathsf{rpre}}_H(A) := \Pr\left[z_1, ..., z_q \xleftarrow{\$} \{0,1\}^n : A^E(z_1, ..., z_p) \Rightarrow m : H^E(m) \in \{z_1, ..., z_p\}\right].$$

We also define

$$\mathbf{Adv}^{\mathsf{rpre}}_H(q, p) := \max_A \mathbf{Adv}^{\mathsf{cr}}_H(A),$$

where the maximum is defined over all adversaries A^E receiving q random targets $z_1, ..., z_q$ and making at most p queries to E.

2.3 AE security with leakage

In general, leakage security definitions are stated w.r.t. *implementations* of a scheme (e.g., an AEAD), and both an encryption leakage function L_e and a decryption leakage function L_d are associated to the implementation(s). This models the real-world leaky implementations of the mathematical objects. Note that in theory, our leakage model is *non-adaptive*, as the leakages are a parameter of the to-be-studied implementations determined before the experiment starts rather than chosen by the adversary during the experiment. This restriction was motivated from the side-channel practice & the necessity for practical modes: see [YSPY10, FPS12] for some discussion. As more recently discussed in [BDF+17], adaptive leakage models also have limited relevance in the context of power and EM side-channels where the adversary has access to the (noisy) leakage of all the intermediate computations. They may be more relevant in the abstract probing model where one probe has to be excluded from the adversary's view, hence making the adaptive selection of a probe an important feature of the analysis.

Pioneered by Rogaway and Shrimpton [RS06], nowadays black-box AE analyses typically follow all-in-one definitions that integrate both confidentiality and integrity. However, in front of an adversary with access to leakage, the adoption of separate definitions for integrity and confidentiality potentially offers more insight on which implementation-level properties are necessary/sufficient for which goal. This difference follows from the important general feature of physically observable cryptography that *unpredictability is much easier to ensure than indistinguishability* [MR04], which has a strong impact on the assumptions that may be needed to prove both notions. Also, different levels of robustness against nonce reuse may be achieved w.r.t. these notions: it was shown that when a nonce is arbitrarily reused (in the same flavor as the *misuse-resistance* notion [RS06]), integrity in the presence of

leakage is achievable [BKP+18, BPPS17], yet confidentiality in the presence of leakage may not [BKP+18, GPPS19]. This difference is also reflected in the separate definitions.

In detail, regarding integrity, we rely on the *Ciphertext Integrity with Misuse*-**resistance** *and Leakage* (CIML2) defined in [BGP+19], which was built upon the single-user version CIML2 introduced in [BKP+18, BPPS17]. The suffix 2 means two leakage sources, i.e., both encryption and decryption. In some sense, the definition is obtained by enhancing the traditional INT-CTXT security with leakages.

Definition 1 (CIML2 advantage)**.** Given the implementation of a nonce-based authenticated encryption $\mathsf{AEAD} = (\mathsf{Enc}, \mathsf{Dec})$ with leakage function pair $\mathsf{L} = (\mathsf{L}_e, \mathsf{L}_d)$, the ciphertext integrity advantage with misuse-*resistance* and leakage of an adversary A against AEAD with is

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{AEAD},\mathsf{L}}(A) := \left| \Pr\left[A^{\mathsf{LEnc}_K,\mathsf{LDec}_K,E,E^{-1}} \Rightarrow 1\right] - \Pr\left[A^{\mathsf{LEnc}_K,\mathsf{LDec}^{\perp}_K,E,E^{-1}}\right] \right| ,$$

where the probability is taken over the key $K \xleftarrow{\$} \mathcal{K}$, over A's random tape and the ideal oracle E and where:

- $\mathsf{LEnc}_K(N, A, M)$: outputs the ciphertext $\mathsf{Enc}_K(N, A, M)$ and the leakages $\mathsf{L}_e(K, N, A, M)$;
- $\mathsf{LDec}_K(N, A, C, T)$: outputs $\big(\mathsf{Dec}_K(N, A, C, T), \mathsf{L}_d(K, N, A, C, T)\big)$;
- $\mathsf{LDec}^{\perp}_K(\ldots)$: computes $\mathsf{L_d} \leftarrow \mathsf{L}_d(K, N, A, C)$ and if C is an output of some leaking encryption query (i, N, A, M) for some M outputs (M, L_d), else outputs $(\perp, \mathsf{L}_d)$.

2.4 Notions under Release of Unverified Plaintext Material

In the RUP model, the adversary can always see the resulting plaintext from a decryption query. To formulate the forgery goal, the oracles are adapted. A verification oracle outputs 1 if and only if the input is valid, and 0 otherwise. A nonce-based RUP AE scheme $\widetilde{\Pi} = (\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K, \widetilde{\mathcal{V}}_K)$ is a three tuple of encryption algorithm $\widetilde{\mathcal{E}} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$, decryption algorithm $\widetilde{\mathcal{D}} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M}$, and verification algorithm $\widetilde{\mathcal{V}}_K : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \{0, 1\}$. The signature of the encryption and decryption algorithms are unchanged.

Definition 2 (INT-RUP Security)**.** Given a nonce-based RUP authenticated encryption $\widetilde{\Pi} = (\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K, \widetilde{\mathcal{V}}_K)$, the INT-RUP advantage with misuse-*resistance* of an adversary A against $\widetilde{\Pi}$ is

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{AEAD},\mathsf{L}}(A) := \left| \Pr\left[A^{\widetilde{\mathcal{E}}_K,\widetilde{\mathcal{D}}_K,\widetilde{\mathcal{V}}_K,E,E^{-1}} \Rightarrow 1\right] - \Pr\left[A^{\widetilde{\mathcal{E}}_K,\widetilde{\mathcal{D}}_K,\perp,E,E^{-1}}\right] \right| ,$$

where the probability is taken over the key $K \xleftarrow{\$} \mathcal{K}$, over A's random tape and the ideal oracle E.

2.5 Privacy Notions in the Presence of Leakage

In [BBC+20], Bellizia *et al.* discussed several security notions in the presence of nonce-misuse and leakage. They show that achieving privacy notions in the presence of leakage can be harder than integrity notions. Privacy notions can be constructed using combinations of the nonce-misuse and leakage assumptions given in Table 1.

The difference between resistance and resilience is related to the challenge queries during the attack. For leakage, leakage-resistance targets adversaries that can observe leakage even during challenge queries, while leakage-resilience targets adversaries that can observe leakage for during the game but have to distinguish a challenge that is leak-free.

Table 1: Different nonce-misuse and leakage assumptions

Nonce	Respecting (.)	Misuse-Resistance (M)	Misuse-Resilience (m)
Leakage	Leak-Free/Black Box (.)	Leakage-Resistance (L)	Leakage-Resilience (l)

Similarly, misuse-resistance targets adversaries that can control the nonce and force it to repeat even during challenge queries, while misuse-resilience refers to the security of messages encrypted using fresh nonces, even if the adversary can repeat other nonces before the challenge queries. These assumptions can be combined with classical CPA and CCA assumptions. For example, CCAML implies security against Chosen Ciphertext Attacks with both nonce Misuse and Leakage resistance. It is shown in [BBC+20] that CCAML can be very hard to achieve (potentially impossible). Hence, some relaxed targets are defined:

(i) CCAmL2: security against Chosen Ciphertext Attacks with both nonce Misuse resilience and Leakage resistance. '2' in this definition, and in the rest of the paper, refers to the type of leakage allowed to the adversary. '2' implies both the decryption and encryption oracle are leaky. However, it is shown in [BBC+20] that even this notion is hard to achieve. It is possible, but at a high performance cost [BGP+19].

(ii) CCAmL1: security against Chosen Ciphertext Attacks with both nonce Misuse resilience and Leakage resistance. '1' implies that only the encryption oracle is leaky.

(iii) CCAml1: security against Chosen Ciphertext Attacks with both nonce Misuse and Leakage resilience. '1' implies that only the encryption oracle is leaky.

The CCAmL1 notion targets adversaries that can access the encryption device for a period of time, being able to observe leakage and misuse the nonces during this period. The security of fresh nonces should be intact. On the other hand, in the case of the CCAml1 the security of fresh nonces used in leak-free queries should be intact.

The l-time assumption We define the l-time assumption as an assumption on the number of traces required in order to recover the key of a TBC using higher order side-channel attacks, *e.g.* Differential Power Analysis (DPA). Consider an implementation of the TBC that is protected against Simple Power Analysis (SPA) and uses a secret key K^*. l refers to the number of traces needed by a given attack in order to practically recover K^*. Hence, l refers to the number of TBC queries that can be allowed using a given key and it is a parameter to the targeted attack. Practical experiments are needed to specify l for a given attack.

3 Towards CIML: TBC-based Hash Function

Black *et al.*. [BRS02] studied the problem of designing hash functions from ideal ciphers that are secure in the chosen key model. They provided a framework for studying the security of such hash functions. However, the hash functions discussed used a block cipher with n-bit key size and n-bit block size. In this section, we propose a similar hash function based on TBCs with n-bit block size and k-bit key size where $k > n$. Specifically, we consider the case when $k = 3n$. The hash function is defined in Definition 3.

Definition 3. Consider an TBC $E_K(X)$ where $|X| = n$ in the block size and $|K| = 3n$ is the tweakey size. Consider a message M of arbitrary length, where $M = M_1 \| M_2 \| \dots M_l$, such that $|M_i| = 3n$, for $1 \geqslant i < l$ and $|M_l| \leqslant 3n$. $H(M)$ is given by

$$\textbf{for } i \leftarrow 1 \textbf{ to } l \textbf{ do } h_i \leftarrow h_i = f(h_{i-1}, M_i) = E_{M_i}(h_{i-1} \oplus \texttt{lmt}_n(M_i))$$

where $h_0 \in \{0,1\}^n$ is a constant.

Theorem 1. *Consider the hash function $H(M)$ is Definition 3.* $\mathbf{Adv}_H^{\mathtt{coll}}(p) \leqslant 3p(p+1)/2^n$ *for all $p \geqslant 1$.*

In order to prove this we follow the proof from Black *et al.*. [BRS02] with small modification. The proof is given in Appendix A.

Theorem 2. *Consider the hash function $H(M)$ is Definition 3.* $\mathbf{Adv}_H^{\mathsf{rpre}}(q,p) \leqslant (2q + p)(p+1)/2^n$ *for all $p \geqslant 1$.*

In order to prove this we follow the proof from Black *et al.* [BRSS10] with small modification. The proof in details is given in Appendix B.

4 Security of AET-LR in the presence of leakage

The AET-LR mode in the presence of leakage is built on top the hash function proposed in Definition 3. The security is divided into the study of integrity and confidentiality. For integrity, the security is studied under the following assumptions:

(i) The Key Derivation Function (KDF) and Tag Generation Function (TGF) are leak-free. In practice, they are heavily protected against complex side-channel attacks, such as Differential Power Analysis (DPA).

(ii) The rest of the operations of the mode suffer unbounded leakage.

(iii) Any nonce can be repeated as many times as the adversary wants.

(iv) Decryption circuits use the inverse TBC operation to convert a tag T to $E^{-1}(T)$ and the verification process compares Y_m to $E^{-1}(T)$. Hence, the decryption leakage cannot be used to find valid tags.

Given these assumptions, the security roughly reduces to the collision and preimage security of the hash function proposed in Section 3. In order to study the security of AET-LR, we first provide a simpler conceptual two-key variant called AET-LRZ, depicted in Figure 2, where both Z and K are long term keys. However, only the last TBC call using K, *i.e.* the TGF, is leak-free.

4.1 AET-LRZ and its CIML2/INT-RUP security

We present the security results on AET-LRZ as follows. First is the CIML2 security, which relies on the cryptographic strength of the underlying hash function H and tweakable blockcipher E^*.

Theorem 3 (CIML2 Security of AET-LRZ)**.** *Assume that E is an ideal cipher with n-bit blocks and $3n$-bit tweakey, then*

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{AET-LRZ}^E}(\sigma_{\mathsf{priv}}, q_d, p) \leqslant \frac{6(\sigma_{\mathsf{priv}}+p)(\sigma_{\mathsf{priv}}+p+1)}{2^n}.$$

Proof. We note that the mode $\mathsf{AET-LRZ}^E$ can be viewed as a hash-then-TBC construction $E^2_K(H^{E^1}(A,N,M))$: the ideal ciphers used by the hashing H^{E^1} and the E^2_K are distinct due to the separation in the tweak. The proof thus resembles [BPPS17]. In detail, denote by G_0 the real CIML2 security game. We will use an intermediate game G_1, in which the adversary $\mathcal{A}$ would face a slightly modified CIML2 security game. We name E_i the event that $\mathcal{A}$ outputs a valid forgery in the game G_i.

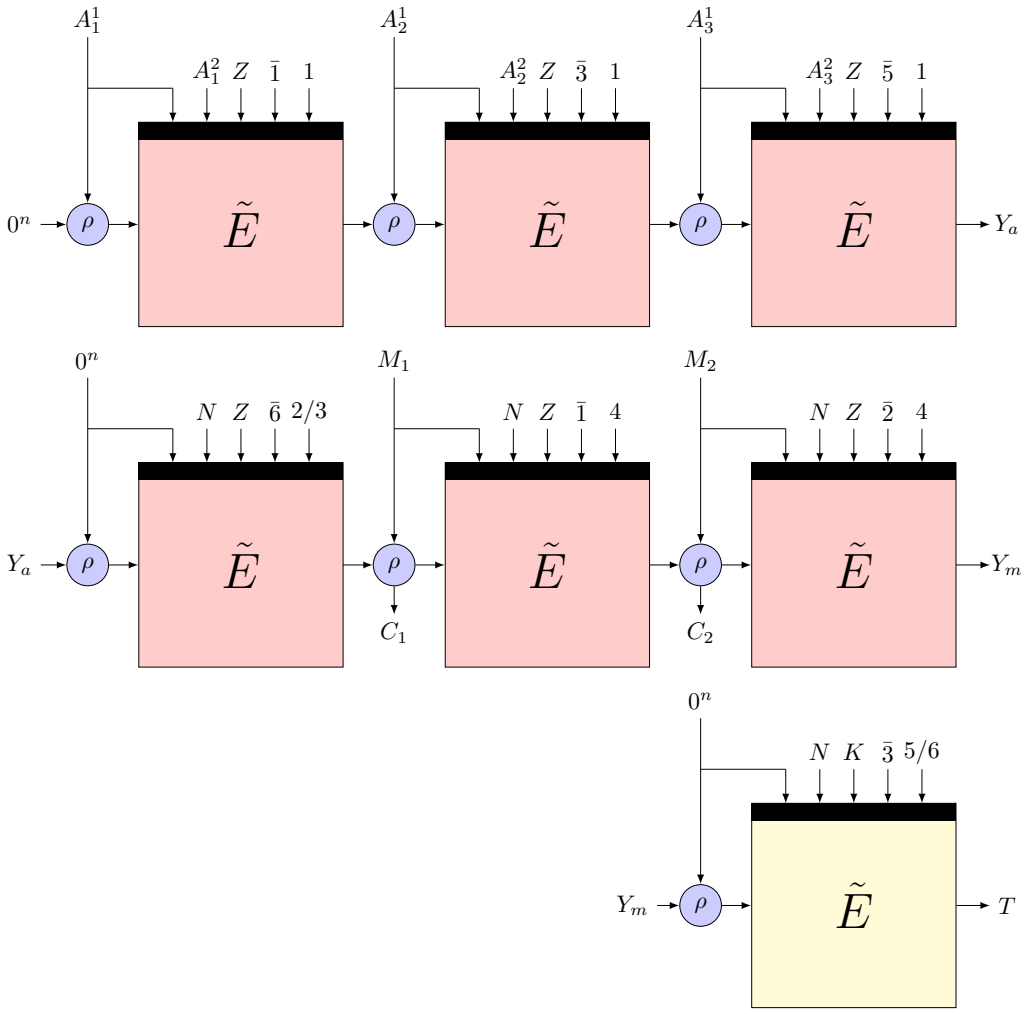

Figure 2: The AET-LRZ mode

As the first step of our proof, we replace all the occurrences of E_k^* and its inverse function by a truly tweakable random permutation $\widetilde{P}$ and its inverse, and this gives rise to G_1. The gap between $\Pr[E_0]$ and $\Pr[E_1]|$ is clearly bounded by the STPRP security of E. For this, we rely on [DS14, Theorem 6], which means

$$\big|\Pr[E_0] - \Pr[E_1]\big| \leqslant \frac{p + \sigma_{\mathsf{priv}}}{2^n}.$$

As the second step, we first define two bad events around the hash function H. In detail, for each encryption query (N_i, A_i, M_i), we associate the values v_i^1 and $\overrightarrow{c}_i$ computed from it via the process described by AET-LRZ. Then the first bad event CollH occurs, if there exists two calls to $\mathsf{H}(x)$ and $\mathsf{H}(x')$ such that $x \neq x'$ yet $\mathsf{H}(x) = \mathsf{H}(x')$. It's clear that by emulating the process of AET-LRZ, an adversary $\mathcal{B}_2$ against the collision intractability of H can be built from the adversary $\mathcal{A}$, and $\mathcal{B}_2$ makes at most $\sigma_{\mathsf{priv}} + p$ queries to E by our assumption. Therefore, by Theorem X, we have $\Pr[\mathsf{CollH}] \leqslant 3(\sigma_{\mathsf{priv}} + p)(\sigma_{\mathsf{priv}} + p + 1)/2^n$.

The other bad event PreH occurs, if there exists a decryption query (N, A, C, T) such that $H(N, A, C) = \widetilde{P}^{-1}(T)$, and T never appeared in the response to any earlier encryption query. To bound $\Pr[\mathsf{PreH}]$, assume that there are distinct tag values $T_1, \ldots, T$ appearing in the decryption query history. Then we note that in the game G_1, the (necessarily distinct) values $h^{1,*} = \widetilde{P}^{-1}(T_1), \ldots, h^{\cdot,*} = \widetilde{P}^{-1}(T)$ can be seen as random values in the view of any efficient algorithms (since they never appeared earlier due to encryption queries), which fits into the definition of range-oriented preimage resistance. Therefore, we could use an adversary $\mathcal{B}_3$ to emulate the game G_1 (the $O(q)$ answers from the random tweakable permutation $\widetilde{P}$ can be simulated by lazy sampling), and if PreH happens then $\mathcal{B}_3$ turns out a range-oriented preimage adversary against the hash function H^E. Since $\mathcal{B}_3$ makes at most $\sigma_{\mathsf{priv}} + p$ queries to E, $\leqslant q_d$, we have $\Pr[\mathsf{PreH}] \leqslant (2q_d + \sigma_{\mathsf{priv}} + p)(\sigma_{\mathsf{priv}} + p + 1)/2^n$ by Theorem 2.

Finally, we argue that if neither CollH nor PreH happens, then all decryption queries respond with $\perp$. To this end, let (N_i, A_i, C_i, T_i) be the valid decryption query with the smallest index on which $\mathcal{A}$ makes. If T_i did not appear in the response to any earlier encryption query, then this query being decided valid contradicts the condition that PreH_1 does not occur. Otherwise, consider this encryption query (N_j, A_j, M_j) and the response (C_j, T_i). By Definition 1, it has to be $(N_j, A_j, C_j) \neq (N_i, A_i, C_i)$, which essentially means $(N_j, A_j, M_j) \neq (N_i, A_i, M_i)$. Then the fact that $H^E(N_j, A_j, M_j) = \widetilde{P}^{-1}(T_i)$ and $H^E(N_i, A_i, M_i) = \widetilde{P}^{-1}(T_i)$ implies $H^E(N_j, A_j, M_j) = H^E(N_i, A_i, M_i)$, contradicting the condition that CollH does not occur.

By the above,

$$\begin{aligned}
&\Pr[E_0] \leqslant \big|\Pr[E_0] - \Pr[E_1]\big| + \Pr[\mathsf{CollH}] + \Pr[\mathsf{PreH}] \\
\leqslant\ &\frac{p + \sigma_{\mathsf{priv}}}{2^n} + \frac{3(\sigma_{\mathsf{priv}} + p)(\sigma_{\mathsf{priv}} + p + 1)}{2^n} + \frac{(2q_d + \sigma_{\mathsf{priv}} + p)(\sigma_{\mathsf{priv}} + p + 1)}{2^n} \\
\leqslant\ &\frac{6(\sigma_{\mathsf{priv}} + p)(\sigma_{\mathsf{priv}} + p + 1)}{2^n},
\end{aligned}$$

as claimed. □

Theorem 4 (INT-RUP Security of AET-LRZ)**.** *Assume that E is an ideal cipher with n-bit blocks and $3n$-bit tweakey, then*

$$\mathbf{Adv}^{\mathsf{INT\text{-}RUP}}_{\mathsf{AET-LRZ}}(\sigma_{\mathsf{priv}}, q_d, p) \leqslant \frac{6(\sigma_{\mathsf{priv}} + p)(\sigma_{\mathsf{priv}} + p + 1)}{2^n}.$$

Proof. Assume that there exists an adversary $\mathcal{A}$ against the INT-RUP security of AET-LRZ. We show that there exists an adversary $\mathcal{B}$ against the CIML2 security of AET-LRZ. In detail, $\mathcal{B}$ runs $\mathcal{A}$ and reacts as follows.

- Upon $\mathcal{A}$ making a query to E, $\mathcal{B}$ simply relays;
- Upon $\mathcal{A}$ making a query to $\widetilde{\mathcal{E}}(N, A, M)$, $\mathcal{B}$ queries $\mathsf{LEnc}(N, A, M)$ and passes the outputs (C, T) to $\mathcal{A}$;
- Upon $\mathcal{A}$ making a query to $\widetilde{\mathcal{D}}(N, A, C, T)$, $\mathcal{B}$ queries $\mathsf{LDec}(N, A, C, T)$. By the construction of $\mathsf{AET\text{-}LRZ}$, the mode will evaluate the decryption and leak the corresponding plaintext M due to the unbounded leakage assumption. By this, $\mathcal{B}$ is able to obtain M and answer $\mathcal{A}$;
- Upon $\mathcal{A}$ making a query to $\widetilde{\mathcal{D}}(N, A, C, T)$, $\mathcal{B}$ queries $\mathsf{LDec}(N, A, C, T)$, and answers 0 or 1 depends on the final return value of $\mathsf{LDec}(N, A, C, T)$.

Eventually, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. By the above, it can be seen

$$\mathbf{Adv}^{\mathsf{INT\text{-}RUP}}_{\mathsf{AET-LRZ}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{AET-LRZ}}(\mathcal{B}),$$

and thus the claim by Theorem 4. □

4.2 CIML2/INT-RUP security of AET-LR

We present the security results on $\mathsf{AET\text{-}LR}$ as follows. First is the $\mathsf{CIML2}$ security.

Theorem 5 ($\mathsf{CIML2}$ Security of $\mathsf{AET\text{-}LR}$)**.** *Assume that E is an ideal cipher with n-bit blocks and $3n$-bit tweakey, then*

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{AET-LR}^E}(\sigma_{\mathsf{priv}}, q_d, p) \leqslant \frac{6(\sigma_{\mathsf{priv}} + p + 1)(\sigma_{\mathsf{priv}} + p)}{2^n}.$$

Proof. We note that the mode $\mathsf{AET-LR}^E$ can be viewed as several independent hash-then-TBC constructions. In detail, assume that there are distinct nonces appearing in the interaction $N_1, ..., N$. Then, the KDF derive distinct initial keys $K_1^*, ..., K^*$, which result in independent tweakable random permutations $E_{K_1^*}, ..., E_{K^*}$ and further independent hash functions $H^{E_{K_1^*}}, ..., H^{E_{K^*}}$. Moreover, these nonces also give rise to independent TGFs $E_K^{N_1}, ..., E_K^N$.

For each nonce N_i, assume that the number of ideal cipher queries under the corresponding key K_i is q_i. Then $\sum_{i=1} q_i = \sigma_{\mathsf{priv}} + p$. By Theorem 3, the probability to forge for $E_K^{N_i}\left(H^{E_{K_i^*}}\right)$ is at most $6q_i(q_i + 1)/2^n$. Then the total forging probability is bounded to

$$\sum_{i=1} \frac{6q_i(q_i + 1)}{2^n} \leqslant \frac{6(\sigma_{\mathsf{priv}} + p + 1)\sum_{i=1} q_i}{2^n} = \frac{6(\sigma_{\mathsf{priv}} + p + 1)(\sigma_{\mathsf{priv}} + p)}{2^n}$$

as claimed. □

In a similar vein to Theorem 4, the $\mathsf{INT\text{-}RUP}$ security of $\mathsf{AET\text{-}LR}$ is also implied by Theorem 5.

Theorem 6 ($\mathsf{INT\text{-}RUP}$ Security of $\mathsf{AET\text{-}LR}$)**.** *Assume that E is an ideal cipher with n-bit blocks and $3n$-bit tweakey, then*

$$\mathbf{Adv}^{\mathsf{INT\text{-}RUP}}_{\mathsf{AET-LR}^E}(\sigma_{\mathsf{priv}}, q_d, p) \leqslant \frac{6(\sigma_{\mathsf{priv}} + p + 1)(\sigma_{\mathsf{priv}} + p)}{2^n}.$$

The proof is similar to the proof of Theorem 4 and is omitted.

4.3 Privacy of AET-LR with Leakage

In this section, we discuss the assumptions and privacy security notions targeted by AET-LR. We target two security notions; CCAml1 and CCAmL1 with an l-time assumption.

4.4 CCAml1 Security of AET-LR

The CCAml1 security of AET-LR is studied under the following assumptions:

(i) The Key Derivation Function (KDF) and Tag Generation Function (TGF) are leak-free. In practice, they are heavily protected against complex side-channel attacks, such as Differential Power Analysis (DPA).

(ii) The rest of the encryption operations of the mode suffer unbounded leakage.

(iii) The decryption operations are leak-free. In practice, they are heavily protected against complex side-channel attacks, such as Differential Power Analysis (DPA).

(iv) Any nonce can be repeated as many times as the adversary wants during the attack preparation phase.

(v) The challenge queries are leak-free and use fresh unique nonces.

The security of AET-LR under these assumptions can be reduced to the security of the KDF. Let a set $\Theta = \{(N_1, Z_1), (N_2, Z_2), \ldots, (N_q, Z_q)\}$, such that $Z_i = E_K(N_i)\ \forall 1 \geqslant i \geqslant q$. Let $\Theta_N = \{N_1, N_2, \ldots, N_q\}$ and $\Theta_Z = \{Z_1, Z_2, \ldots, Z_q\}$ and a nonce $N_c \notin \Theta_N$. Then, $Z_c \notin \Theta_N$ and all the TBC calls corresponding to (N_c, Z_c) are fresh, *i.e.* use permutations that have not appeared before in previous queries. Hence,

$$\mathbf{Adv}^{\mathsf{CCAml1}}_{\mathsf{AET-LR}^E}(\sigma_{\mathsf{priv}}, q_d, p) \leqslant \mathbf{Adv}^{\mathsf{TPRP}}_{E}(q_e + q_d) + \mathbf{Adv}^{\mathsf{NAE}}_{\mathsf{AET-LR}^E}(\sigma, q_e, q_d, p)$$

where $\mathbf{Adv}^{\mathsf{TPRP}}_{E}(q_e+q_d)$ refers to the security of the KDF function and $\mathbf{Adv}^{\mathsf{NAE}}_{\mathsf{AET-LR}^E}(\sigma, q_e, q_d, p)$ refers to the black box security of AET-LR in the nonce-respecting model.

4.5 CCAmL1 Security of AET-LR

The CCAml1 security of AET-LR is studied under the following assumptions:

(i) The Key Derivation Function (KDF) and Tag Generation Function (TGF) are leak-free. In practice, they are heavily protected against complex side-channel attacks, such as Differential Power Analysis (DPA).

(ii) The rest of the encryption operations of the mode are protected against SPA/cheap side-channel attacks.

(iii) The decryption operations are leak-free. In practice, they are heavily protected against complex side-channel attacks, such as Differential Power Analysis (DPA).

(iv) Any nonce can be repeated as many times as the adversary wants during the attack preparation phase.

(v) The challenge queries are leak-free and use fresh unique nonces.

(vi) The encryption queries can consist of at most l blocks where $l = m + \frac{a}{2}$, such that m is the number of plaintext blocks and the a is the number of associated data blocks (l-time assumption).

The security of AET-LR under these assumptions can be reduced to the security of the KDF and the security of the countermeasure used for the encryption operations, aside from the KDF and TGF. Let a set $\Theta = \{(N_1, Z_1), (N_2, Z_2), \ldots, (N_q, Z_q)\}$, such that $Z_i = E_K(N_i)\ \forall 1 \geqslant i \geqslant q$. Let $\Theta_N = \{N_1, N_2, \ldots, N_q\}$ and $\Theta_Z = \{Z_1, Z_2, \ldots, Z_q\}$ and a nonce $N_c \notin \Theta_N$. Then, $Z_c \notin \Theta_N$ and all the TBC calls corresponding to (N_c, Z_c) are fresh, *i.e.* use permutations that have not appeared before in previous queries. Hence,

$$\mathbf{Adv}^{\mathsf{CCAmL1}}_{\mathsf{AET-LR}^E}(\sigma_{\mathsf{priv}}, q_d, p) \leqslant \mathbf{Adv}^{\mathsf{TPRP}}_{E}(q_e + q_d) + \mathbf{Adv}^{\mathsf{NAE}}_{\mathsf{AET-LR}^E}(\sigma, q_e, q_d, p) + \mathbf{Adv}^{\mathsf{DPA}}_{E}(l)$$

where $\mathbf{Adv}^{\mathsf{DPA}}_{E}(l)$ refers to the security of the TBC protected using a cheap countermeasure against an l-time adversary.

5 Conclusions and Future Works

In this short article, we propose the AET-LR mode of operation targeted towards leakage-resilient implementations using low implementation cost. We show it achieves birthday-bound integrity notions even with unbounded leakage and nonce-misuse resistance, *e.g.* CIML2 and INT-RUP. We also discuss different privacy notions in the presence of leakage. We show it is assumed to be secure against CCAml1 adversaries and CCAmL1 adversaries for short messages. We emphasize that this mode is very close to the Romulus-N mode submitted to the NIST competition (basically an added KDF at initialisation, and a feed-forward of the message block into the tweak inputs of the TBC calls).

This article is not to be viewed as full version, but an extended abstract. While we provide solid security bounds against CIMl2 and INT-RUP adversaries. The discussion of privacy notion is less formal and reduces the security to the black box security of the mode. We intend to provide a full version of this article with black box security analysis, and more formal privacy analysis, and implementations.

Acknowledgements

We would like to thank François-Xavier Standaert, Olivier Pereira, Thomas Peters, Tetsu Iwata and Kazuhiko Minematsu on their insightful comments on this mode and leakage resilience security notions. The second and third authors are supported by Temasek Laboratories, Singapore.

References

[BBB+20] Davide Bellizia, Francesco Berti, Olivier Bronchain, Gaëtan Cassiers, Sébastien Duval, Chun Guo, Gregor Leander, Gaëtan Leurent, Itamar Levi, Charles Momin, Olivier Pereira, Thomas Peters, François-Xavier Standaert, Balazs Udvarhelyi, and Friedrich Wiemer. Spook: Sponge-based leakage-resistant authenticated encryption with a masked tweakable block cipher. *IACR Transactions on Symmetric Cryptology*, 2020(S1):295–349, Jun. 2020. https://tosc.iacr.org/index.php/ToSC/article/view/8623.

[BBC+20] Davide Bellizia, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Mode-level vs. implementation-level physical security in symmetric cryptography. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 369–400, Cham, 2020. Springer International Publishing.

[BDF+17] Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 535–566. Springer, Heidelberg, April / May 2017.

[BGP+19] Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Tedt, a leakage-resist aead mode for high physical security applications. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(1):256–320, Nov. 2019. https://tches.iacr.org/index.php/TCHES/article/view/8400.

[BKP+18] 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 *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, 2018.

[BPPS17] Francesco Berti, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. On leakage-resilient authenticated encryption with decryption leakages. *IACR Trans. Symm. Cryptol.*, 2017(3):271–293, 2017.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from pgv. In *Annual International Cryptology Conference*, pages 320–335. Springer, 2002.

[BRSS10] John Black, Phillip Rogaway, Thomas Shrimpton, and Martijn Stam. An analysis of the blockcipher-based hash functions from PGV. *Journal of Cryptology*, 23(4):519–545, October 2010.

[DS14] Yuanxi Dai and John Steinberger. Tight security bounds for multiple encryption. Cryptology ePrint Archive, Report 2014/096, 2014. http://eprint.iacr.org/2014/096.

[FPS12] Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical leakage-resilient symmetric cryptography. In Emmanuel Prouff and Patrick Schaumont, editors, *CHES 2012*, volume 7428 of *LNCS*, pages 213–232. Springer, Heidelberg, September 2012.

[GPPS19] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Authenticated encryption with nonce misuse and physical leakage: Definitions, separation results and first construction - (extended abstract). In *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings*, pages 150–172, 2019.

[IKMP20a] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Duel of the Titans: The Romulus and Remus Families of Lightweight AEAD Algorithms. *IACR Transactions on Symmetric Cryptology*, 2020(1):43–120, May 2020. https://tosc.iacr.org/index.php/ToSC/article/view/8560.

[IKMP20b] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Romulus v1.2. NIST Lightweight Cryptography Project, 2020. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/Romulus-spec-round2.pdf.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 278–296. Springer, Heidelberg, February 2004.

[RS06] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, Heidelberg, May / June 2006.

[YSPY10] Yu Yu, François-Xavier Standaert, Olivier Pereira, and Moti Yung. Practical leakage-resilient pseudorandom generators. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 2010*, pages 141–151. ACM Press, October 2010.

A Proof of Theorem 1

Proof. We define a directed graph $G = (V_G, E_G)$ with vertex set $V_G = \{0,1\}^n \times \{0,1\}^\kappa \times \{0,1\}^n$ and an arc $(x,k,y) \to (x',k',y')$ in E_G if and only if $\lfloor k' \rfloor_n \oplus x' = y$.

Let $A^{?,?}$ be an adversary attacking H. We analyze the behavior of A when its left oracle is instantiated by $E \xleftarrow{\$} \mathit{Bloc}(n,\kappa)$ and its right oracle is instantiated by E^{-1}. Assume that A asks its oracles at most p total queries. We must show that $\mathbf{Adv}^{\mathtt{coll}}_H(p) \leqslant 3p(p+1)/2^n$. Run the algorithm SimulateOracles(A,n). As A executes with its (simulated) oracle, color the vertices of G as follows:

- Initially, each vertex of G is uncolored.
- When A asks an E-query (k,x) and this returns a value y, or when A asks an E^{-1}-query of (k,y) and this returns x, then: if $x \oplus \lfloor k \rfloor_n = h_0$ then vertex (x,k,y) gets colored red; otherwise vertex (x,k,y) gets colored black.

Given A does not repeat queries (x,k,y), every query A asks results in exactly one vertex getting colored red or black, that vertex formerly being uncolored. A vertex of G is colored when it gets colored red or black. A path P in G is colored if all of its vertices are colored. Vertices (x,k,y) and (x',k',y') are said to collide if $y = y'$. Distinct paths P and P' are said to collide if all of their vertices are colored and they begin with red vertices and they end with colliding vertices. Let C be the event that, as a result of the adversary's queries, there are formed in G some two colliding paths.

Claim. $\mathbf{Adv}^{\mathtt{coll}}_H(p) \leqslant \Pr[\mathsf{C}]$.

Claim. $\Pr[\mathsf{C}] \leqslant 3p(p+1)/2^n$

The result for H now follows by combining the two claims. □

Proof. Claim A. Suppose that the adversary A outputs colliding messages $M = m_1 \cdots m_a$ and $M' = m'_1 \cdots m'_b$ that is, $H(M) = H(M')$ for the simulated oracle E. We show that, necessarily, there are two colliding paths.

Let $P = (x_1,k_1,y_1) \to \cdots \to (x_a,k_a,y_a)$ where, for each $i \in [1 \cdots a]$, $x_i = h_{i-1} \oplus \lfloor m_i \rfloor_n, k_i = m_i, y_i = E_{k_i}(x_i)$, and $h_i = y_i$. Similarily, let $P' = (x'_1,k'_1,y'_1) \to \cdots \to (x'_b,k'_b,y'_b)$ where, for each $i \in [1 \cdots b]$, $x'_i = h'_{i-1} \oplus \lfloor m'_i \rfloor_n, k'_i = m'_i, y'_i = E_{k'_i}(x'_i)$, and $h'_i = y'_i$, and where $h_0 = h'_0$ is a fixed constant. We claim that P and P' are colliding paths.

A makes all of the queries necessary to compute $H(M)$ and $H(M')$. So, for each $i \in [1 \cdots a]$, A must have made either an E-query (k_i,x_i) or an E^{-1}-query (k_i,y_i). Similarly, for each $i \in [1 \cdots b]$, A must have made either an E-query (k'_i,x'_i) or an E^{-1}-query (k'_i,y'_i). We can conclude then that P and P' are colored. Moreover, $x_1 \oplus \lfloor k_1 \rfloor_n = h_0$ and $x'_1 \oplus \lfloor k_1 \rfloor_n = h'_0$ so each of P and P' starts with a red node.

If $a \neq b$, then clearly P and P' are distinct. Consider $a = b$ and $M \neq M'$. There is some $i \in [1 \cdots a]$ such that $m_i \neq m'_i$, and so $(x_i, k_i, y_i) \neq (x'_i, k'_i, y'_i)$. Finally, if M and M' collide we have $h_a = h'_b$, and hence $y_a = y'_b$. □

Proof. Claim A. Let C_i be the event that C occurs by the i-th query. Define $\bar{C}_0$ to be the null event. Then $\Pr[sfC] = \Sigma_{i=1}^{q} \Pr[\mathsf{C}_i | \bar{\mathsf{C}}_{i-1} \wedge \cdots \wedge \bar{\mathsf{C}}_0]$. Given $\bar{\mathsf{C}}_{i-1} \wedge \cdots \wedge \bar{\mathsf{C}}_0$, the event C_i occurs in one of four ways. To make the discussion of these cases more clear, we define a little more notation. Let $\mathsf{Arc}(i, j)$ be the event that there exists in G vertices (x_i, k_i, y_i) and (x_j, k_j, y_j), and $y_i = x_j \oplus \lfloor k_j \rfloor_n$. Let $\mathsf{Red}(i)$ be the event that there exists in G vertex (x_i, k_i, y_i) and $x_i \oplus \lfloor k_i \rfloor_n = h_0$. Let $\mathsf{Collide}(i, j)$ be the event that there exists in G vertices (x_i, k_i, y_i) and (x_j, k_j, y_j), and $y_i = y_j$.

Case 1 A vertex (x_i, k_i, y_i) is colored on the i-th query, and there exists in G arcs $(x_r, k_r, y_r) \rightarrow (x_i, k_i, y_i)$ and $(x_i, k_i, y_i) \rightarrow (x_j, k_j, y_j)$, where (x_r, k_r, y_r) and (x_j, k_j, y_j) were colored on the r-th and j-th queries, respectively.

This event requires $\mathsf{Arc}(r, i) \wedge \mathsf{Arc}(i, j)$ be true, such that $y_r = x_i \oplus \lfloor k_i \rfloor_n$ and $y_i = x_j \oplus \lfloor k_j \rfloor_n$. If C_i occures via an E-query (k_i, x_i), then y_i is a random value from a set of size $2^n - A$, where A is the number of colored nodes where $k_a = k_i$, for all $a < i$, such that $A \leqslant i - 1$. There are $(i - 1 - A)$ unique possible choices for (x_j, k_j, y_j) such that $k_j \neq k_i$, with at most $(i - 1 - A)$ targets for $x_j \oplus \lfloor k_j \rfloor_n$. Then,

$$\Pr[\mathsf{Arc}(r, i) \wedge \mathsf{Arc}(i, j)] = \Pr[\mathsf{Arc}(i, j) | \mathsf{Arc}(r, i)] \Pr[\mathsf{Arc}(r, i)]$$

$$\leqslant \Pr[\mathsf{Arc}(i, j) | \mathsf{Arc}(r, i)] \leqslant \frac{i - 1 - A}{2^n - A} \leqslant \frac{i - 1}{2^n - (i - 1)}$$

Alternatively, if C_i occures via an E^{-1}-query (k_i, y_i), then x_i is a random value from a set of size at least $2^n - (i - 1)$. Similarily, there are at most $(i - 1)$ unique possible choices for (x_r, k_r, y_r) with at most $(i - 1)$ targets for $y_r = x_i \oplus \lfloor k_i \rfloor_n$. Then,

$$\Pr[\mathsf{Arc}(r, i) \wedge \mathsf{Arc}(i, j)] = \Pr[\mathsf{Arc}(r, i) | \mathsf{Arc}(i, j)] \Pr[\mathsf{Arc}(i, j)]$$

$$\leqslant \Pr[\mathsf{Arc}(r, i) | \mathsf{Arc}(i, j)] \leqslant \frac{i - 1}{2^n - (i - 1)}$$

Case 2 A vertex (x_i, k_i, y_i) is colored red on the i-th query, and there exists in G an arc $(x_i, k_i, y_i) \rightarrow (x_j, k_j, y_j)$, where (x_j, k_j, y_j) was colored on the j-th query, $j < i$.

This event requires $\mathsf{Arc}(r, i) \wedge \mathsf{Red}(i)$ be true. If C_i occures via an E-query (k_i, x_i), then y_i is a random value from a set of size at least $2^n - (i - 1)$. There are at most $(i - 1)$ unique possible choices for (x_j, k_j, y_j) with at most $(i - 1)$ targets for $x_j \oplus \lfloor k_j \rfloor_n$. Then,

$$\Pr[\mathsf{Arc}(i, j) \wedge \mathsf{Red}(i)] = \Pr[\mathsf{Arc}(i, j) | \mathsf{Red}(i)] \Pr[\mathsf{Red}(i)]$$

$$\leqslant \Pr[\mathsf{Arc}(i, j) | \mathsf{Red}(i)] \leqslant \frac{i - 1}{2^n - (i - 1)}$$

Alternatively, if C_i occures via an E^{-1}-query (k_i, y_i), then x_i is a random value from a set of size at least $2^n - (i - 1)$. Then,

$$\Pr[\mathsf{Red}(i) \wedge \mathsf{Arc}(i, j)] = \Pr[\mathsf{Red}(i) | \mathsf{Arc}(i, j)] \Pr[\mathsf{Arc}(i, j)]$$

$$\leqslant \Pr[\mathsf{Red}(i) | \mathsf{Arc}(i, j)] \leqslant \frac{i - 1}{2^n - (i - 1)}$$

Case 3 A vertex (x_i, k_i, y_i) is colored on the i-th query, and there exists in G an arc $(x_i, k_i, y_i) \rightarrow (x_j, k_j, y_j)$, and a vertex (x_r, k_r, y_r), where (x_j, k_j, y_j) and (x_r, k_r, y_r) were colored on the j-th and r-th queries, $r, j < i$, respectively, and (x_i, k_i, y_i) agrees with (x_r, k_r, y_r).

This event requires that $\mathsf{Arc}(j, i)$ and $\mathsf{Collide}(i, r)$. If this occurs via an E-query (k_i, x_i), then y_i is a random value from a set of size at least $2n - (i-1)$. Then,

$$\Pr[\mathsf{Arc}(i,j) \wedge \mathsf{Collide}(i,r)] = \Pr[\mathsf{Collide}(i,r)|\mathsf{Arc}(i,j)] \Pr[\mathsf{Arc}(i,j)]$$

$$\leqslant \Pr[\mathsf{Collide}(i,r)|\mathsf{Arc}(i,j)] \leqslant \frac{i-1}{2^n - (i-1)}$$

Alternatively, if C_i occures via an E^{-1}-query (k_i, y_i), then x_i is a random value from a set of size at least $2^n - (i-1)$. Then,

$$\Pr[\mathsf{Arc}(i,j) \wedge \mathsf{Collide}(i,r)] = \Pr[\mathsf{Arc}(i,j)|\mathsf{Collide}(i,r)] \Pr[\mathsf{Collide}(i,r)]$$

$$\leqslant \Pr[\mathsf{Arc}(i,j)|\mathsf{Collide}(i,r)] \leqslant \frac{i-1}{2^n - (i-1)}$$

Case 4 A vertex (x_i, k_i, y_i) is colored red on the i-th query, and there exists in G an arc $(x_i, k_i, y_i) \rightarrow (x_i, k_i, y_i)$.

This event requires that $\mathsf{Red}(i) \wedge \mathsf{Arc}(i, i)$ is true. If this occurs via an E-query (k_i, x_i), then y_i is a random value from a set of size at least $2n - (i-1)$. There is only one possible target for y_i, which is h_0. Then,

$$\Pr[\mathsf{Arc}(i,i) \wedge \mathsf{Red}(i)] = \Pr[\mathsf{Arc}(i,i)|\mathsf{Red}(i)] \Pr[\mathsf{Red}(i)]$$

$$\leqslant \Pr[\mathsf{Arc}(i,i)|\mathsf{Red}(i)] \leqslant \frac{1}{2^n - (i-1)}$$

Alternatively, if C_i occures via an E^{-1}-query (k_i, y_i), then x_i is a random value from a set of size at least $2^n - (i-1)$. Then,

$$\Pr[\mathsf{Red}(i) \wedge \mathsf{Arc}(i,i)] = \Pr[\mathsf{Red}(i)|\mathsf{Arc}(i,i)] \Pr[\mathsf{Arc}(i,i)]$$

$$\leqslant \Pr[\mathsf{Red}(i)|\mathsf{Arc}(i,i)] \leqslant \frac{1}{2^n - (i-1)}$$

Combining all cases, we have

$$\Pr[\mathsf{C}] \leqslant \sum_{i=1}^{p} \frac{3(i-1)}{2^n - (i-1)} + \frac{1}{2^n - (i-1)} \leqslant 3 \sum_{i=1}^{p} \frac{i}{2^n - (i-1)} \leqslant 3 \sum_{i=1}^{p} \frac{i}{2^n - (p-1)}$$

$$= \frac{3p(p+1)}{2(2^n - (p-1))} = \frac{3p(p+1)}{2^{n+1} - 2(p-1)} \leqslant \frac{3p(p+1)}{2^n}$$

□

B Proof of Theorem 2

Proof. Let $h_0 \in \{0,1\}^n$. Let A be an adversary with oracles E, E^{-1} and input σ. Assume that A asks its oracles q queries in total. We are interested in A's behavior when its left oracle is instantiated by $E \xleftarrow{\$} Bloc(n,\kappa)$ and its right oracle is instantiated by E^{-1}.

As in the proof of Theorem X, we define an undirected graph $G = (V_G, E_G)$ with vertex set $V_G = \{0,1\}^n$—corresponding to all 2^n possible chaining values—and initially an empty edge set $E_G = \varnothing$. We will dynamically add edges based on the queries to E and E^{-1}. In particular, we add an edge (h,z), labeled by m, if we know a message m such that $z = f^E(h,m)$ (or $h = f^E(z,m)$) and the relevant query to either E or E^{-1} has been made. We claim that to find a preimage would require finding a path between the (fixed) initial vector and the q (random) targets $\sigma_1, ..., \sigma_q$: suppose that $H(M) = \sigma$, where $M = (m_1, ..., m_\ell)$ and correspondingly $h_1, ..., h_\ell$ for the chaining values of the iterated hash. Noting that $h_\ell = \sigma$, we see that $h_{0..\ell}$ is a (possibly empty) path from initial vector to target.

Since we are dynamically adding edges to the graph, components in the graph will also grow dynamically. Let T_0 be the set of all nodes that are in the component containing h_0, and similarly let $T_{\sigma_1}, ..., T_{\sigma_q}$ be the set of all nodes connected to $\sigma_1, ..., \sigma_q$ respectively. Unless $h_0 = \sigma_i$ for $i \in \{1, ..., q\}$, which happens with probability $q/2^n$, T_0 and $T_{\sigma_1}, ..., T_{\sigma_q}$ are initially disjoint. However, when a path between h_0 and σ_i is present, we have $T_0 = T_{\sigma_i}$.

The first claim is that after j queries, the sets T_0 and $T_{\sigma_1}, ..., T_{\sigma_q}$ have combined cardinality at most $j + q + 1$. Indeed, either component has at most $j' + 1$ nodes when j' edges are used. By construction, a query (either forward or inverse) will add at most one edge to the graph, so after j queries, there are at most j edges in the entire graph and at most $j + q + 1$ nodes in T_0 plus $T_{\sigma_1}, ..., T_{\sigma_q}$.

The second claim is that to complete the path between h_0 and σ_i, an edge needs to be added with one end (node) in T_0 and the other in $T_{\sigma_1}, ..., T_{\sigma_q}$. Writing $T = T_0 \cup T_{\sigma_1} \cup ... \cup T_{\sigma_q}$, we need to find an edge with both nodes already part of T. Consider the j-th query. For a forward query, the distribution of $f^E(h,m) = z$ is uniform over a set of size at least $2^n - j$. For an inverse query, the distribution of h is uniform over a set of size at least $2^n - j$. Consequently, the probability that on the j-th query a preimage is found is upper bounded by $(j + q - 1)/(2^n - j)$. (The set T contains $j + q$ elements before the j-th query, the query itself also needs to specify a node in T, but a self-loop cannot possibly connect T_0 and $T_{\sigma_1}, ..., T_{\sigma_q}$.)

With a union bound we can bound the probability of finding a preimage within p or fewer queries: $\mathbf{Adv}^{\mathsf{rpre}}_H(q,p) \leqslant q/2^n + \sum_{j=1}^{p}(j + q - 1)/(2^n - j) \leqslant (q + p)(q + p + 1)/2^n$ leading to the stated upper bound. □