# RIV for Robust Authenticated Encryption

Farzaneh Abed[1(✉)], Christian Forler[2], Eik List[1], Stefan Lucks[1], and Jakob Wenzel[1]

[1] Bauhaus-Universität Weimar, Weimar, Germany
{farzaneh.abed,eik.list,stefan.lucks,jakob.wenzel}@uni-weimar.de
[2] Hochschule Schmalkalden, Schmalkalden, Germany
cforler@hs-schmalkalden.de

**Abstract.** Typical AE schemes are supposed to be secure when used as specified. However, they can – and often do – fail miserably when used improperly. As a partial remedy, Rogaway and Shrimpton proposed (nonce-)misuse-resistant AE (MRAE) and the first MRAE scheme SIV ("Synthetic Initialization Vector"). This paper proposes RIV ("Robust Initialization Vector"), which extends the generic SIV construction by an additional call to the internal PRF. RIV inherits the full security assurance from SIV, but unlike SIV and other MRAE schemes, RIV is also provably secure when releasing unverified plaintexts. This follows a recent line of research on *"Robust Authenticated Encryption"*, similar to the CAESAR candidate AEZ.

An AES-based instantiation of RIV runs at less than 1.5 cpb on current x64 processors. Unlike the proposed instantiation of AEZ, which gains speed by relying on reduced-round AES, our instantiation of RIV is provably secure under the single assumption of the AES being secure.

**Keywords:** Robustness · Subtle authenticated encryption · Provable security

## 1 Introduction

***Authenticated Encryption.*** A secure authenticated encryption (AE) scheme generates ciphertexts that can not be efficiently distinguished from random bitstrings of the same length as the ciphertext and are infeasible to forge. Typical AE schemes are nonce-based [45], i.e., the user is responsible to supply an additional input that must be unique for every encryption. If a nonce ever repeats, the scheme's security may fully forfeit. While the concept of unique nonces is simple in theory, it is hard to ensure in practice [19], which led to severe security breaches in the past. Rogaway and Shrimpton [46] defined (nonce-)misuse-resistant AE (MRAE) as notion with the goal of providing full authenticity, and privacy up to the detection of repeated encryptions of the same associated data and message under the same nonce and key. Since then, the topic received significant attention by the community, resulting in a large corpus of MRAE schemes, e.g., [6,10,16,20,22,27–30,33,43,46].


T. Peyrin (Ed.): FSE 2016, LNCS 9783, pp. 23–42, 2016.
DOI: 10.1007/978-3-662-52993-5_2

Robustness aspects of AE are not limited to nonce reuse. "One shortcoming of AE as commonly understood is its idealized, all-or-nothing decryption" [7]. Leaking any information about the message before its authentication has been verified breaks this assumption. At least five noteworthy recent works strengthened the existing security definitions of robustness.[1] Boldyreva et al. [15] (BDPS) studied the effects when multiple distinct error messages are distinguishable in probabilistic or stateful schemes. Andreeva et al. [4] formalized notions that capture the remaining security under *release of unverified plaintexts* (RUP). Hoang et al. [24] defined *robust AE* (RAE) as a notion for the *best achievable* security of an AE scheme with a user-chosen ciphertext expansion. Badertscher et al. [5] investigated RAE with the frameworks by Maurer and Renner [38,39]. Barwell et al. [7] defined *subtle AE* (SAE) as a reference framework for the BDPS, RUP, and RAE notions. The SAE definitions comprise leakage beyond information about the invalid plaintext, which allows to model leakage as a property of the decryption *implementation* rather than as a property of the scheme.

***Previous Robust AE Schemes.*** In spite of so much progress regarding stricter security definitions, the portfolio of dedicated robust AE schemes remains still modest. Among the 57 CAESAR submissions, only four candidates consider robustness against leakage of invalid plaintexts: Julius [6] lacks a security proof; POET [1] and APE [3] concern on-line confidentiality, which cannot provide nonce-misuse resistance in the strong sense of Rogaway and Shrimpton, as has been criticized, e.g., by [25]. Only AEZ [24] provides robust AE. Though, AEZ follows a "proof-then-prune" approach: while the security proof assumes a strong block cipher, the performant instantiation employs four-round AES instead. Since AEZ also defines a key schedule, it appears more as a primitive of its own right than as a block-cipher-based AE scheme.

Beyond CAESAR, Bertoni et al. [12] proposed Mr. Monster Burrito, a four-round Feistel network with the round-reduced Keccak-$f$ permutation in duplex-wrap mode, and the sponge in counter mode for encryption. Shrimpton and Terashima [47] proposed Protected IV (PIV), a framework of strong tweakable ciphers (STPRPs), which generalized the $\Psi_3$ construction by Coron et al. [17]. PIV is fast (comparable with the construction proposed in this work); though, it requires the block-cipher inverse for decryption. Note that theoretically, more robust AE schemes could be constructed. Hoang et al. [24] showed that the well-known Encode-then-Encipher (EtE) [9] approach achieves RAE security when (a hash of) nonce and associated data are used as tweak. In theory, this implies that a secure STPRP can be transformed into a robust AE scheme, which allows to choose from the schemes that have been developed over the previous decade, e.g., in the domains of full-disk and format-preserving encryption.

***Contribution.*** This work proposes a modular framework, called *Robust IV* (RIV), which provides provable SAE security. RIV is an extension of SIV [26,46]

[1] By robustness, we mean resistance against both nonce misuse and decryption leakage beyond the single error information.

that inherits both the simplicity and the naturally strong security properties of SIV and adds robustness against leakage of invalid plaintexts. We propose an instantiation which runs at less than 1.5 clock cycles per byte (cpb) on current x64 processors.

***Outline.*** The remainder of this work is structured as follows: after Sect. 2 recalls the preliminaries, Sect. 3 describes the generic RIV framework. Section 4 recalls the relevant notions. Section 5 summarizes our formal security analysis. Section 6 details our instantiation, and Sect. 7 concludes this work.

## 2 Preliminaries

We use lowercase letters $x, y$ for indices and integers, uppercase letters $X, Y$ for binary strings and functions, and calligraphic uppercase letters $\mathcal{X}, \mathcal{Y}$ for sets. By $\varepsilon$ we denote the empty string. We denote the concatenation of binary strings $X$ and $Y$ by $X \,\|\, Y$ and the result of their bitwise XOR by $X \oplus Y$. We indicate the length of $X$ in bits by $|X|$, and write $X_i$ for the $i$-th block, $X[i]$ for the $i$-th most significant bit of $X$, and $X[i..j]$ for the bit sequence $X[i], \ldots, X[j]$. $X \twoheadleftarrow \mathcal{X}$ denotes that $X$ is chosen uniformly at random from the set $\mathcal{X}$. We define two sets of particular interest: $\mathsf{Perm}(\mathcal{X})$ be the set of all permutations on $\mathcal{X}$ and $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ the set of all functions $F : \mathcal{X} \to \mathcal{Y}$. A uniform random function $\rho : \mathcal{X} \to \mathcal{Y}$ with domain $\mathcal{X}$ and range $\mathcal{Y}$ is a random variable uniformly distributed over $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$. We define by $X_1, \ldots, X_j \xleftarrow{x} X$ the injective splitting of the string $X$ into $x$-bit blocks such that $X = X_1 \,\|\, \cdots \,\|\, X_j$, with $|X_i| = x$ for $1 \leq i \leq j-1$, and $|X_j| \leq x$.

For an event $E$, we denote by $\Pr[E]$ the probability of $E$. We write $\langle x \rangle_m$ for the binary $m$-bit-string representation of an integer $x$ and $\langle x \rangle$ for the binary $n$-bit-string representation of $x$ for an integer $n$ that is clear from the context. If not stated otherwise, we assume representations to be encoded in big-endian manner, i.e., the decimal $\langle 135 \rangle$ is encoded to the $n$-bit string 000..010000111.

***Universal Hashing.*** Universal hash functions are well-known components for compressing a message while guaranteeing maximal probabilities about output relations. We briefly recall the definitions that are relevant in this work.

**Definition 1 ($\epsilon$-Almost-(XOR-)Universal Hash Functions).** *Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^*$. Let $\mathcal{H} = \{H \,|\, H : \mathcal{X} \to \mathcal{Y}\}$ denote a family of hash functions. $\mathcal{H}$ is called $\epsilon$-almost-universal ($\epsilon$-AU) iff for all distinct elements $X, X' \in \mathcal{X}$, it holds that $\Pr_{H \twoheadleftarrow \mathcal{H}}[H(X) = H(X')] \leq \epsilon$. $\mathcal{H}$ is called* $\epsilon$-almost-XOR-universal *($\epsilon$-AXU) iff for all distinct elements $X, X' \in \mathcal{X}$ and $Y \in \mathcal{Y}$, it holds that $\Pr_{H \twoheadleftarrow \mathcal{H}}[H(X) \oplus H(X') = Y] \leq \epsilon$.*

**Theorem 1 (Theorem 3 from [14]).** Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^*$. Further, let $\mathcal{H} = \{H \,|\, H : \mathcal{X} \to \mathcal{Y}\}$ be a family of $\epsilon$-AXU hash functions. Then, the family $\mathcal{H}' = \{H' \,|\, H' : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}\}$ with $H'(X, Y) := H(X) \oplus Y$, is $\epsilon$-AU.

***Nonce-Based Encryption Schemes.*** A *nonce-based encryption scheme* [45] is a tuple $\Pi = (\mathcal{E}, \mathcal{D})$ of deterministic encryption and decryption algorithms $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \to \mathcal{C}$ and $\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{C} \to \mathcal{M}$, with associated non-empty key space $\mathcal{K}$, non-empty nonce space $\mathcal{N}$, and $\mathcal{M}, \mathcal{C} \subseteq \{0,1\}^*$ denoting message and ciphertext space, respectively. We often write $\mathcal{E}_K^N(M)$ and $\mathcal{D}_K^N(C)$ as short forms of $\mathcal{E}(K, N, M)$ and $\mathcal{D}(K, N, C)$. An adversary that never repeats a nonce over its encryption queries is called *nonce-respecting*, and *nonce-ignoring* otherwise. We assume for all $K \in \mathcal{K}$, $N \in \mathcal{N}$, $M \in \mathcal{M}$, and $C \in \mathcal{C}$ *length-preservation*, i.e., $|\mathcal{E}_K^N(M)| = |M|$, *correctness*, i.e., $\mathcal{D}_K^N(\mathcal{E}_K^N(M)) = M$, and *tidiness*, i.e., $\mathcal{E}_K^N(\mathcal{D}_K^N(C)) = C$. We call a nonce-based encryption scheme $\Pi = (\mathcal{E}, \mathcal{D})$ *nonce-keystream-based* iff its encryption algorithm derives a keystream $\kappa_N \subseteq \{0,1\}^*$, with $|\kappa_N| = |M|$, from the given nonce $N$ and computes the ciphertext as $C \leftarrow \kappa_N \oplus M$. Naturally, the decryption algorithm of such an encryption scheme is identical to its encryption algorithm, i.e., $\mathcal{E}_K^N(M) := \mathcal{D}_K^N(M)$ for all $K \in \mathcal{K}$, $N \in \mathcal{N}$, and $M \in \mathcal{M}$.

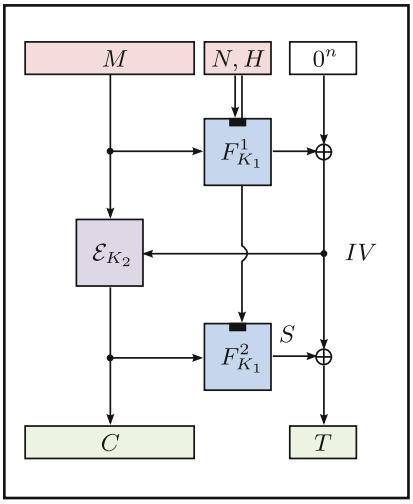
***Nonce-Based AE Schemes.*** A nonce-based authenticated encryption scheme (with associated data) [44] is a tuple $\widetilde{\Pi} = (\widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ of a deterministic encryption algorithm $\widetilde{\mathcal{E}} : \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$, and a deterministic decryption algorithm $\widetilde{\mathcal{D}} : \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}$, with associated non-empty key space $\mathcal{K}$, non-empty nonce space $\mathcal{N}$, and $\mathcal{H}$, $\mathcal{M}$, $\mathcal{C} \subseteq \{0,1\}^*$ denote the header, message, and ciphertext space, respectively. We define a tag space $\mathcal{T} = \{0,1\}^\tau$ for a fixed $\tau \geq 0$. We often write $\widetilde{\mathcal{E}}_K^{N,H}(M)$ and $\widetilde{\mathcal{D}}_K^{N,H}(C,T)$ as short forms of $\widetilde{\mathcal{E}}(K, N, H, M)$ and $\widetilde{\mathcal{D}}(K, N, H, C, T)$. If a given tuple $(N, H, C, T)$ is valid, $\widetilde{\mathcal{D}}_K^{N,H}(C,T)$ returns the corresponding plaintext $M$, and $\perp$ otherwise. We assume that for all $K \in \mathcal{K}$, $N \in \mathcal{N}$, $H \in \mathcal{H}$, and $M \in \mathcal{M}$ holds *stretch-preservation*: if $\widetilde{\mathcal{E}}_K^{N,H}(M) = (C,T)$, then $|C| = |M|$ and $|T| = \tau$, *correctness*: if $\widetilde{\mathcal{E}}_K^{N,H}(M) = (C,T)$, then $\widetilde{\mathcal{D}}_K^{N,H}(C,T) = M$, and *tidiness*: if $\widetilde{\mathcal{D}}_K^{N,H}(C,T) = M \neq \perp$, then $\widetilde{\mathcal{E}}_K^{N,H}(M) = (C,T)$, for all $C \in \mathcal{C}$ and $T \in \mathcal{T}$. Note that some notions (e.g., [41]) regard an authenticated ciphertext $C$ with $|C| = |M| + \tau$ instead of an explicitly separated tuple $(C,T)$.

***Subtle AE Schemes.*** Barwell et al. defined a subtle AE scheme $\widetilde{\Pi} = (\widetilde{\mathcal{E}}, \widetilde{\mathcal{D}}, \Lambda)$ as a tuple of deterministic encryption and decryption algorithms $\widetilde{\mathcal{E}}$ and $\widetilde{\mathcal{D}}$ as above[2], and an additional deterministic leakage algorithm $\Lambda : \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{C} \times \mathcal{T} \to \{\top\} \cup \mathcal{L}$, with a non-empty leakage space $\mathcal{L}$ and a symbol $\top \notin \mathcal{L}$ to indicate a valid input. This means, for all $K \in \mathcal{K}$, $N \in \mathcal{N}$, $H \in \mathcal{H}$, $C \in \mathcal{C}$, and $T \in \mathcal{T}$ holds: if $\Lambda_K^{N,H}(C,T) = \top$, then $\widetilde{\mathcal{D}}_K^{N,H}(C,T) \neq \perp$; moreover, it holds that if $\Lambda_K^{N,H}(C,T) \neq \top$, then $\widetilde{\mathcal{D}}_K^{N,H}(C,T) = \perp$.

## 3 Definition of RIV

**Definition 2** (RIV). *Let $d, n, \tau \geq 1$. Let $\mathcal{K}_1$, $\mathcal{K}_2$, and $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2$ be non-empty key sets, $\mathcal{N}$ a non-empty nonce space, $\{0,1\}^d$ the non-empty domain*

[2] Though, their definitions denote the authenticated ciphertext $(C,T)$ as $C$.




1: **function** $\widetilde{\mathcal{E}}_{K_1,K_2}(N,H,M)$
2: $IV \leftarrow F^1_{K_1}(N,H,M)$
3: $C \leftarrow \mathcal{E}_{K_2}(IV,M)$
4: $T \leftarrow F^2_{K_1}(N,H,C) \oplus IV$
5: **return** $(C,T)$

11: **function** $\widetilde{\mathcal{D}}_{K_1,K_2}(N,H,C,T)$
12: $IV \leftarrow F^2_{K_1}(N,H,C) \oplus T$
13: $M \leftarrow \mathcal{D}_{K_2}(IV,C)$
14: $IV' \leftarrow F^1_{K_1}(N,H,M)$
15: **if** $IV = IV'$ **then return** $M$
16: **return** $\perp$

21: **function** $\Lambda_{K_1,K_2}(N,H,C,T)$
22: $IV \leftarrow F^2_{K_1}(N,H,C) \oplus T$
23: $M \leftarrow \mathcal{D}_{K_2}(IV,C)$
24: $IV' \leftarrow F^1_{K_1}(N,H,M)$
25: **if** $IV = IV'$ **then return** $\top$
26: **return** $M$

**Fig. 1. Left:** Schematic illustration of the encryption of $\mathrm{RIV}_{F,\Pi}$ with a PRF $F$ and a nonce-based encryption scheme $\Pi = (\mathcal{E},\mathcal{D})$. **Right:** Definition of encryption and decryption algorithms of $\mathrm{RIV}_{F,\Pi}$, and definition of a plaintext-leaking oracle $\Lambda$ that will be used in our security analysis.

*space, and* $\mathcal{H},\mathcal{M},\mathcal{C} \subseteq \{0,1\}^*$ *header, message, and ciphertext spaces, respectively, and* $\mathcal{T} = \{0,1\}^\tau$ *a tag space. Let further* $F : \mathcal{K}_1 \times \{0,1\}^d \times \mathcal{N} \times \mathcal{H} \times \mathcal{M} \to \{0,1\}^n$ *be a function and* $\Pi = (\mathcal{E},\mathcal{D})$ *a nonce-based encryption scheme with associated key space* $\mathcal{K}_2$ *and nonce space* $\{0,1\}^\tau$*. Let* $F^i_K(\cdot,\cdot,\cdot)$ *denote* $F_K(\langle i\rangle_d,\cdot,\cdot,\cdot)$*. Then, we define the AE scheme* $\mathrm{RIV}_{F,\Pi} = (\widetilde{\mathcal{E}},\widetilde{\mathcal{D}})$ *with encryption algorithm* $\widetilde{\mathcal{E}} : \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$ *and decryption algorithm* $\widetilde{\mathcal{D}} : \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}$*, as given in Fig. 1.*

**Definition 3** ($\widehat{\mathrm{RIV}}$)**.** *We define the* SAE *scheme* $\widehat{\mathrm{RIV}}_{F,\Pi} = (\widetilde{\mathcal{E}},\widetilde{\mathcal{D}},\Lambda)$ *with an additional deterministic leakage algorithm* $\Lambda : \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \times \{\top\}$*, as given in Fig. 1.*

***Feistel Structure and Encode-then-Encipher (EtE).*** RIV can be seen as an application of the EtE [9] approach by Bellare et al. EtE can generically be used for constructing a robust AE scheme from a tweakable cipher, assuming its enciphering resists chosen-plaintext and chosen-ciphertext attacks [24]. The RIV cipher, however, is essentially an unbalanced three-round Feistel-network.[3] It is well-known that such ciphers are secure against chosen-plaintext, but vulnerable to chosen-ciphertext attacks [35] (see also [2,36,42]). RIV is robust *in spite* of its weak enciphering scheme, because its encoding operation has been chosen to specifically cover this weakness.

[3] If the used encryption scheme $\Pi = (\mathcal{E},\mathcal{D})$ is nonce-*keystream*-based, the RIV cipher *is* a three-round Feistel network.

## 4 Security Notions

***Adversaries and Advantages.*** An adversary $\mathbf{A}$ is an efficient Turing machine that interacts with a given set of oracles that appear as black boxes to $\mathbf{A}$. We use the notation $\mathbb{A}$ for the class of all computationally bounded adversaries and $\mathbf{A}^{\mathcal{O}}$ for the output of $\mathbf{A}$ after interacting with some oracle $\mathcal{O}$. We write $\Delta_{\mathbf{A}}(\mathcal{O}^L; \mathcal{O}^R) := \sup_{\mathbf{A}\in\mathbb{A}} |\Pr[\mathbf{A}^{\mathcal{O}^L} \Rightarrow 1] - \Pr[\mathbf{A}^{\mathcal{O}^R} \Rightarrow 1]|$ for the advantage of $\mathbf{A}$ to distinguish between oracles $\mathcal{O}^L$ and $\mathcal{O}^R$. All probabilities are defined over the random coins of the oracles and those of the adversary, if any. We write $\mathbf{Adv}^X_F(q, \ell, t) = \max_{\mathbf{A}\in\mathbb{A}}\{\mathbf{Adv}^X_F(\mathbf{A})\}$ to refer to the maximal advantage over all $X$-adversaries $\mathbf{A}$ on a given function $F$ that run in time at most $t$ and pose at most $q$ queries consisting of at most $\ell$ blocks in total to the available oracles. If $\mathbf{A}$ shall distinguish between two sets of oracles $(\mathcal{O}^L_1, \ldots, \mathcal{O}^L_k)$ and $(\mathcal{O}^R_1, \ldots, \mathcal{O}^R_k)$, we refer to the $i$-th oracle that $\mathbf{A}$ interacts with by $\mathcal{O}_i \in \{\mathcal{O}^L_i, \mathcal{O}^R_i\}$. By $\mathcal{O}_i \hookrightarrow \mathcal{O}_j$, we denote that $\mathbf{A}$ first queries $\mathcal{O}_i$ and later $\mathcal{O}_j$ with the output of $\mathcal{O}_i$. Wlog., we assume that $\mathbf{A}$ never asks queries to which it already knows the answer. In the case when $\mathbf{A}$ has access to multiple oracles $\mathcal{O}_1, \ldots, \mathcal{O}_k$, we denote by $q_i$ the number of queries and by $\ell_i$ the maximal number of blocks that $\mathbf{A}$ poses at most to oracle $\mathcal{O}_i$, $1 \le i \le k$.

If $\mathcal{O}_i$ and $\mathcal{O}_j$ represent a family of algorithms indexed by inputs, the indices must match, e.g., when $\widetilde{\mathcal{E}}^{N,H}_K(M)$ and $\widetilde{\mathcal{D}}^{N,H}_K(C)$ represent encryption and decryption algorithms with a fixed key $K$ and indexed by $N$ and $H$, then $\widetilde{\mathcal{E}}_K \hookrightarrow \widetilde{\mathcal{D}}_K$ says that $\mathbf{A}$ first queries $\widetilde{\mathcal{E}}^{N,H}_K(M)$ and later $\widetilde{\mathcal{D}}^{N,H}_K(C)$.

We define $\perp$, when in place of an oracle, to always return the invalid symbol $\perp$. We denote by $\$^{\mathcal{O}}$ an oracle that, given an input $X$, computes $Y \leftarrow \mathcal{O}(X)$, chooses uniformly at random a value $Y'$ from the space of all possible outputs with $|Y'| = |Y|$, and returns $Y'$. We assume that $\$^{\mathcal{O}}$ performs lazy sampling, i.e., $\$^{\mathcal{O}}(X)$ returns the same value when queried with the same input $X$. We often omit the key for brevity, e.g., $\$^{\widetilde{\mathcal{E}}}(X)$ will be short for $\$^{\widetilde{\mathcal{E}}_K}(X)$.

### 4.1 Security Definitions for Encryption Schemes

**Definition 4 (PRF Advantage).** *Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a function with non-empty key space $\mathcal{K}$, and $\mathbf{A}$ a computationally bounded adversary with access to an oracle, where $K \twoheadleftarrow \mathcal{K}$ and $\rho \twoheadleftarrow \mathsf{Func}(\mathcal{X}, \mathcal{Y})$. Then, the* PRF *advantage of $\mathbf{A}$ on $F$ is defined as $\mathbf{Adv}^{\mathrm{PRF}}_F(\mathbf{A}) := \Delta_{\mathbf{A}}(F_K; \rho)$.*

**Definition 5 (PRP Advantage).** *Let $n, k \ge 1$ be fixed. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher and $\mathbf{A}$ a computationally bounded adversary with access to an oracle. Further, let $K \twoheadleftarrow \{0,1\}^k$ and $\pi \twoheadleftarrow \mathsf{Perm}(\{0,1\}^n)$. Then, the* PRP *advantage of $\mathbf{A}$ on $E$ is defined as $\mathbf{Adv}^{\mathrm{PRP}}_E(\mathbf{A}) := \Delta_{\mathbf{A}}(E_K; \pi)$.*

Stinson [48] showed that one can construct an $(\epsilon_1 + \epsilon_2)$-AU family of hash functions from the consecutive application of an $\epsilon_1$-AU and an $\epsilon_2$-AU family of hash functions. From that we can derive the following theorem.

**Theorem 2.** *Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \{0,1\}^*$ and let $\mathcal{K}$ be a non-empty set. Further, let $\mathcal{H} = \{H : \mathcal{X} \to \mathcal{Y}\}$ be a family of $\epsilon$-AU hash functions and let $G : \mathcal{K} \times \mathcal{Y} \to \mathcal{Z}$ be a function. Then, we can define $F_K(X) := G_K(H(X))$, with independent $K \twoheadleftarrow \mathcal{K}$ and $H \twoheadleftarrow \mathcal{H}$. Let $\mathbf{A}$ be a* PRF *adversary on $F$ that asks at most $q$ queries of at most $\ell$ blocks in total, and runs in time at most $t$. Then, there exists a* PRF *adversary $\mathbf{A}_1$ on $G$ that asks at most $q$ queries and runs in time $O(t)$ such that*

$$\mathbf{Adv}_F^{\mathrm{PRF}}(\mathbf{A}) \leq \mathbf{Adv}_G^{\mathrm{PRF}}(\mathbf{A}_1) + \epsilon \cdot q^2/2.$$

Theorem 2 follows from the fact, that the PRF advantage of $F$ is upper bounded by the maximal PRF advantage on $G$ plus the maximal probability of output collisions of the form $H(X) = H(X')$ over $q$ queries.

**Definition 6 (nE Advantage [41]).** Let $\Pi = (\mathcal{E}, \mathcal{D})$ be a nonce-based encryption scheme and $K \twoheadleftarrow \mathcal{K}$. Let $\mathbf{A}$ be a nonce-respecting adversary with access to an oracle. Then, the nE advantage of $\mathbf{A}$ on $\Pi$ is defined as $\mathbf{Adv}_{\Pi}^{\mathrm{nE}}(\mathbf{A}) := \Delta_{\mathbf{A}}(\mathcal{E}_K; \$^{\mathcal{E}})$.

We adapt the definition of indistinguishability from random bits from [23] for nonce-based encryption schemes. Note that we strengthen it to adversaries that do not repeat nonces over *all encryption and decryption queries.*

**Definition 7 (SRND Advantage).** *Let be $\Pi = (\mathcal{E}, \mathcal{D})$ a nonce-based encryption scheme and $K \twoheadleftarrow \mathcal{K}$. Let $\mathbf{A}$ be a nonce-respecting adversary with access to two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$, s.t. $\mathbf{A}$ never asks for $\mathcal{O}_1 \hookrightarrow \mathcal{O}_2$ and never repeats a nonce over all its encryption and decryption queries. Then, we define the* SRND *advantage of $\mathbf{A}$ on $\Pi$ as $\mathbf{Adv}_{\Pi}^{\mathrm{SRND}}(\mathbf{A}) := \Delta_{\mathbf{A}}(\mathcal{E}_K, \mathcal{D}_K; \$^{\mathcal{E}}, \$^{\mathcal{D}})$.*

### 4.2 Security Definitions for Nonce-Based AE Schemes

For this subsection, let $\widetilde{\Pi} = (\widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ be a nonce-based AE scheme, $K \twoheadleftarrow \mathcal{K}$, and $\mathbf{A}$ be a computationally bounded adversary on $\widetilde{\Pi}$.

**Definition 8 (IND-CPA Advantage).** *Let $\mathbf{A}$ have access to an encryption oracle. Then, the* IND-CPA *advantage of $\mathbf{A}$ with respect to $\widetilde{\Pi}$ is defined as $\mathbf{Adv}_{\widetilde{\Pi}}^{\text{IND-CPA}}(\mathbf{A}) := \Delta_{\mathbf{A}}(\widetilde{\mathcal{E}}_K; \$^{\widetilde{\mathcal{E}}})$.*

**Definition 9 (INT-CTXT Advantage).** *Let $\mathbf{A}$ have access to two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ such that $\mathbf{A}$ never queries $\mathcal{O}_1 \hookrightarrow \mathcal{O}_2$. Then, the* INT-CTXT *advantage of $\mathbf{A}$ on $\widetilde{\Pi}$ is defined as $\mathbf{Adv}_{\widetilde{\Pi}}^{\text{INT-CTXT}}(\mathbf{A}) := \Pr[\mathbf{A}^{\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K} \text{forges}]$, where "forges" means that $\widetilde{\mathcal{D}}_K$ returns anything other than $\perp$ for a query of $\mathbf{A}$.*

**Definition 10 (nAE Advantage [41]).** Let $\mathbf{A}$ have access to two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ such that $\mathbf{A}$ never queries $\mathcal{O}_1 \hookrightarrow \mathcal{O}_2$. Then, the nAE advantage of $\mathbf{A}$ on $\widetilde{\Pi}$ is defined as $\mathbf{Adv}_{\widetilde{\Pi}}^{\mathrm{nAE}}(\mathbf{A}) := \Delta_{\mathbf{A}}(\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K; \$^{\widetilde{\mathcal{E}}}, \perp)$.

Bellare and Namprempre showed for probabilistic AE that chosen-ciphertext security results from IND-CPA and INT-CTXT security [8]. Fleischmann et al. proved in [19] a generalized theorem for nonce-based AE.

**Theorem 3 (Theorem 1 in [19]).** Let $\mathbf{A}$ be a computationally bounded nAE adversary on $\widetilde{\Pi}$ with access to two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ such that $\mathbf{A}$ never queries $\mathcal{O}_1 \hookrightarrow \mathcal{O}_2$; $\mathbf{A}$ makes at most $q$ queries of total length of at most $\ell$ blocks and runs in time at most $t$. Then, there exist an IND-CPA adversary $\mathbf{A}_1$ on $\widetilde{\Pi}$ and an INT-CTXT adversary $\mathbf{A}_2$ on $\widetilde{\Pi}$, both making at most $q$ queries of at most $\ell$ blocks and running in time $O(t)$ each, such that

$$\mathbf{Adv}^{\text{nAE}}_{\widetilde{\Pi}}(\mathbf{A}) \leq \mathbf{Adv}^{\text{IND-CPA}}_{\widetilde{\Pi}}(\mathbf{A}_1) + \mathbf{Adv}^{\text{INT-CTXT}}_{\widetilde{\Pi}}(\mathbf{A}_2).$$

### 4.3 Security Definitions for Subtle AE Schemes

Subtle AE (SAE) defines a compound security notion that provides guarantees for privacy and authenticity under the existence of a leakage oracle. It comprises the notions IND-CPA, INT-CTXT, and an additional notion ERR-CCA.

For this subsection, let $\widetilde{\Pi} = (\widetilde{\mathcal{E}}, \widetilde{\mathcal{D}}, \Lambda)$ be an SAE scheme, $K, K' \twoheadleftarrow \mathcal{K} \times \mathcal{K}$ independent keys, and $\mathbf{A}$ a deterministic adversary with access to three oracles $\mathcal{O}_1, \mathcal{O}_2$, and $\mathcal{O}_3$ such that $\mathbf{A}$ neither queries $\mathcal{O}_1 \hookrightarrow \mathcal{O}_2$ nor $\mathcal{O}_1 \hookrightarrow \mathcal{O}_3$.

**Definition 11 (ERR-CCA Advantage).** *The* ERR-CCA *advantage of* $\mathbf{A}$ *on* $\widetilde{\Pi}$ *is defined as* $\mathbf{Adv}^{\text{ERR-CCA}}_{\widetilde{\Pi}}(\mathbf{A}) := \Delta_{\mathbf{A}}(\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K, \Lambda_K; \widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K, \Lambda_{K'})$.

**Definition 12 (SAE Advantage).** *The* SAE *advantage of* $\mathbf{A}$ *on* $\widetilde{\Pi}$ *is defined as* $\mathbf{Adv}^{\text{SAE}}_{\widetilde{\Pi}}(\mathbf{A}) := \Delta_{\mathbf{A}}(\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K, \Lambda_K; \$^{\widetilde{\mathcal{E}}}, \perp, \Lambda_{K'})$.

In the full version of [7], Barwell et al. prove a statement equivalent to Theorem 4. We apply Theorem 3 to decompose their AE security advantage term into the separate advantages for IND-CPA and INT-CTXT.

**Theorem 4.** *Let* $\mathbf{A}$ *run in time at most* $t$ *and ask at most* $q$ *queries of at most* $\ell$ *blocks to its respective oracles. Then, there exist computationally bounded* IND-CPA, INT-CTXT, *and* ERR-CCA *adversaries* $\mathbf{A}_1$, $\mathbf{A}_2$, *and* $\mathbf{A}_3$, *respectively, on* $\widetilde{\Pi}$ *such that*

$$\mathbf{Adv}^{\text{SAE}}_{\widetilde{\Pi}}(\mathbf{A}) \leq \mathbf{Adv}^{\text{IND-CPA}}_{\widetilde{\Pi}}(\mathbf{A}_1) + \mathbf{Adv}^{\text{INT-CTXT}}_{\widetilde{\Pi}}(\mathbf{A}_2) + \mathbf{Adv}^{\text{ERR-CCA}}_{\widetilde{\Pi}}(\mathbf{A}_3),$$

*where* $\mathbf{A}_1$, $\mathbf{A}_2$, *and* $\mathbf{A}_3$ *each make at most* $q$ *queries of at most* $\ell$ *blocks and run in time* $O(t)$ *each.*

Since [4] omitted a compound notion for their security under release of unverified plaintexts, Barwell et al. defined RUPAE as $\Delta_{\mathbf{A}}(\widetilde{\mathcal{E}}_K, \widetilde{\mathcal{D}}_K, \mathcal{V}_K; \$^{\widetilde{\mathcal{E}}}, \widetilde{\mathcal{D}}_{K'}, \perp)$ [7, Theorem 3, Corollary 2]. They showed that the maximal SAE advantage on an AE scheme $\widetilde{\Pi}$ is, with a reduction term, also equivalent to the maximal RUPAE advantage. Moreover, they showed that – again with a reduction term – it is also equivalent to the maximal robust-AE advantage on $\widetilde{\Pi}$ with fixed stretch $\tau$.

## 5 Security Results for Generic RIV

This section summarizes our security results. For the remainder of this section, let $d, n, \tau \geq 1$ be integers, $\mathcal{K}_1$, $\mathcal{K}_2$ be non-empty key spaces, and $K_1, K_2 \twoheadleftarrow \mathcal{K}_1 \times \mathcal{K}_2$ be independent keys, $F : \mathcal{K}_1 \times \{0,1\}^d \times \mathcal{N} \times \mathcal{H} \times \mathcal{M} \to \{0,1\}^n$, and $\Pi = (\mathcal{E}, \mathcal{D})$ be a nonce-based encryption scheme with associated key space $\mathcal{K}_2$.

**Theorem 5.** *Let* $\mathbf{A}$ *be a computationally bounded* SAE *adversary on* $\widehat{\mathrm{RIV}}_{F,\Pi}$ *which asks at most* $q$ *queries of at most* $\ell$ *blocks in total and runs in time at most* $t$*. Then, there exists a computationally bounded* PRF *adversary* $\mathbf{A}_1$ *on* $F$ *that asks at most* $2q$ *queries of at most* $2(d + n\ell)$ *bits and runs in time* $O(t)$*, and a computationally bounded* SRND *adversary* $\mathbf{A}_2$ *on* $\Pi$ *that asks at most* $q$ *queries of at most* $\ell$ *blocks in total and runs in time* $O(t)$ *such that*

$$\mathbf{Adv}^{\mathrm{SAE}}_{\widehat{\mathrm{RIV}}_{F,\Pi}}(\mathbf{A}) \leq \frac{8q^2 + 3q}{2^n} + 4 \cdot \left( \mathbf{Adv}^{\mathrm{PRF}}_{F}(\mathbf{A}_1) + \mathbf{Adv}^{\mathrm{SRND}}_{\Pi}(\mathbf{A}_2) \right).$$

Due to space limitations, the proof can be found in the full version of this paper[4]. We can derive the following corollary for the nAE advantage on $\mathrm{RIV}_{F,\Pi}$ in the absence of a plaintext-leaking oracle.

**Corollary 1.** *Let* $\mathbf{A}$ *be a computationally bounded* nAE *adversary on* $\mathrm{RIV}_{F,\Pi}$ *which asks at most* $q$ *queries of at most* $\ell$ *blocks in total and runs in time at most* $t$*. Then, there exist a computationally bounded* PRF *adversary* $\mathbf{A}_1$ *on* $F$ *that asks at most* $2q$ *queries of at most* $2(d + n\ell)$ *bits and runs in time* $O(t)$*, and a computationally bounded* SRND *adversary* $\mathbf{A}_2$ *on* $\Pi$ *that asks at most* $q$ *queries of at most* $\ell$ *blocks in total and runs in time* $O(t)$*, such that*

$$\mathbf{Adv}^{\mathrm{nAE}}_{\mathrm{RIV}_{F,\Pi}}(\mathbf{A}) \leq \frac{2q^2 + q}{2^n} + 2 \cdot \left( \mathbf{Adv}^{\mathrm{PRF}}_{F}(\mathbf{A}_1) + \mathbf{Adv}^{\mathrm{SRND}}_{\Pi}(\mathbf{A}_2) \right).$$

The proof can be found can be found in the full version of this paper.

***Proof Ideas.*** The intuition of our proofs is the following: in encryption direction, for every fresh tuple of nonce, header, and message, $F$ will produce a fresh $IV \leftarrow F^1_K(N, H, M)$ that has not occurred before with overwhelming probability. Since $\Pi$ is SRND-secure, $\mathcal{E}$ will produce a randomly chosen ciphertext. The second invocation of $F$ with a fresh ciphertext then produces a random tag. To determine the privacy advantage of the scheme, we have to bound only the PRF-advantage on $F$, the SRND-security of $\mathcal{E}$, and the probabilities of random collisions of $IV$s from the birthday paradox.

In decryption direction, whenever the nonce, header, or ciphertext changes, $IV \leftarrow F^2_K(N, H, C)$ will be a random value up to the birthday bound. Since $\Pi$ is SRND-secure, a fresh $IV$ (regarded over all encryption *and* decryption queries) will produce a fresh pseudorandom plaintext. Thus, even when the adversary learns the decrypted (invalid) message, $M$ will provide it with no information

[4] The full version of this paper will soon appear on ePrint.

about other plaintexts as long as the $IV$ does not repeat. When an adversary changes $N$, $H$, or $C$ and manages to cancel the difference by a fresh tag, the second call to $F_K^1(N, H, M)$ will yield a random $IV'$ that differs from $IV$ with probability close to $1/2^n$. Thus, a similar argumentation as for the encryption also applies to the inverse direction. Finally, the domain separation from the first parameter to $F$ protects against choices of $(N, H, M) = (N, H, C)$.

## 6 Instantiation

***Pseudo-Dot-Product Hashing.*** Let $n, m \geq 1$ with even $m$ and let $\mathcal{X} = \bigcup_{i=1}^{m/2} \{0,1\}^{2in}$. Given a set of $m$ pair-wise independent key words $K = (K_1, \ldots, K_m)$ and an $m$-word input $M = (M_1, \ldots, M_m)$, with $M_i, K_i \in \{0,1\}^n$, $1 \leq i \leq m$, a *pseudo-dot-product* (PDP) family of hash functions $\mathcal{H} = \{H : \mathcal{X} \times \mathcal{X} \to \{0,1\}^{2n}\}$ is defined as

$$H_K(M) := \sum_{i=1}^{m/2} (M_{2i-1} + K_{2i-1}) \cdot (M_{2i} + K_{2i}).$$

Bernstein [11] credits it to Winograd [51] and classifies it as $(m, \lceil m/2 \rceil)$-design, i.e., it requires $m$ independent key words and $\lceil m/2 \rceil$ multiplications to process $m$ message words. If modular additions and multiplications are performed within the rings $\mathbb{Z}_{2^n}$ and $\mathbb{Z}_{2^{2n}}$, the construction is known as NH, to be $1/2^n$-AU, and is used in variants in UMAC [13], VMAC [18,32], and HS1 [33]. All these constructions employ a multi-stage hashing process: the input is first compressed with NH, before the results are used as inputs in a usual polynomial hash (and optionally further processed by an inner-product hash). To obtain a slightly higher security margin and efficiency, we consider a recently proposed variant, called CLHASH.

### 6.1 CLHASH

CLHASH [34] is a family of multi-stage hash functions that produces 64-bit hashes and employs a PDP family of hash functions CLNH, which resembles NH, but replaces modular additions and multiplications with XORs and carry-less multiplications in $\mathbb{GF}(2^{64})/p(\mathtt{x})$ with the irreducible polynomial $p(\mathtt{x}) = \mathtt{x}^{64} + \mathtt{x}^4 + \mathtt{x}^3 + \mathtt{x} + \mathtt{1}$. Therefore, CLNH can exploit the `vpclmulqdq` instruction for 64-bit carry-less multiplication which was originally introduced for boosting the performance of GCM [21].

CLHASH$[m]$ splits a given message $M$ into $(64m)$-bit blocks $(M_1, \ldots, M_s)$, and pads the final block with zeroes such that its length becomes a multiple of 128 bits. Each block $M_j$ is compressed with CLNH to a 128-bit value $A_j$. If the message consists of only a single block, the message length $|M|$ is multiplied with an independent key $K_L \in \{0,1\}^{64}$ and XORed to the result; the result is reduced to a 64-bit value modulo $p(\mathtt{x}) = \mathtt{x}^{64} + \mathtt{x}^4 + \mathtt{x}^3 + \mathtt{x} + \mathtt{1}$ and returned.

**Algorithm 1.** Definition of $\mathrm{CLHASH}^{\mathsf{T}}[m,t]$ with a hash length of $64t$ bits, a block length of $m/8$ bytes, and $t$ Toeplitz iterations.

**function** $\mathrm{CLHASH}^{\mathsf{T}}[m,t]_K(M)$
  $(K_N, K_P, K_A, K_L) \leftarrow \textsc{KeyGen}(K)$
  $s \leftarrow \max(\lceil 64 \cdot |M|/m \rceil, 1)$
  $(M_1, \ldots, M_s) \xleftarrow{64m} M$
  $M_s \leftarrow \textsc{pad}_{128}(M_s)$
  **for** $i \leftarrow 1$ to $t$ **do**
    **for** $j \leftarrow 1$ to $s$ **do**
      $K_j \leftarrow K_{N(2i-1)..m+2(i-1)}$
      $A_j \leftarrow \mathrm{CLNH}[m]_{K_j}(M_j)$
    **if** $s = 1$ **then**
      $H_1 \leftarrow A_1$
    **else**
      $K_{P\,i} \leftarrow K_{P\,i} \bmod 2^{126}$
      $O_i \leftarrow \textsc{Poly}_{K_{P\,i}}(A_1, \ldots, A_s)$
      $H_i \leftarrow \mathrm{CLNH}[2]_{K_{A\,i}}(O_i)$
    $H_i \leftarrow \textsc{HashLen}_{K_{L\,i}}(H_i, |M|)$
  **return** $(H_1 \parallel \cdots \parallel H_t)$

**function** $\mathrm{CLNH}[m]_{K_j}(M_j)$
  **return** $\bigoplus_{i=1}^{m} (M_{j\,2i-1} \oplus K_{j\,2i-1})$
    $\cdot (M_{j\,2i} \oplus K_{j\,2i})$

**function** $\textsc{KeyGen}(K)$
  $\kappa \leftarrow 64(m + 2t - 2)$
  $K_N \leftarrow K[1..\kappa]$
  $K_P \leftarrow K[(\kappa+1)..(\kappa+128t)]$
  $\kappa \leftarrow \kappa + 128t$
  $K_A \leftarrow K[(\kappa+1)..(\kappa+128t)]$
  $\kappa \leftarrow \kappa + 128t$
  $K_L \leftarrow K[(\kappa+1)..(\kappa+64t)]$
  **return** $(K_N, K_P, K_A, K_L)$

**function** $\textsc{Poly}_{K_P}(A_1, \ldots, A_s)$
  **return** $\bigoplus_{i=1}^{s} A_i \cdot K_P^{s-i}$
    $\bmod (2^{128} + 4 + 2)$

**function** $\textsc{HashLen}_{K_L}(H_i, |M|)$
  **return** $(H_i \oplus (K_L \cdot |M|))$
    $\bmod (2^{64} + 27)$

**function** $\textsc{pad}_n(X)$
  **if** $(|X| \bmod n = 0)$ **then**
    **return** $X$
  **return** $X \parallel 0^{n - |X| \bmod n}$

For longer messages, the values $A_j$ are processed by a polynomial hash with an independent key $K_P \in \{0,1\}^{128}$ and reduced modulo $q(\mathtt{x}) = \mathtt{x}^{127} + \mathtt{x} + \mathtt{1}$. For efficiency, the two most significant bits of $K_P$ are fixed to zero, and a lazy reduction modulo $\mathtt{x}^{128} + \mathtt{x}^2 + \mathtt{x}$ is used instead without affecting security.

The 128-bit result of the polynomial hash is then reduced to a 64-bit value by another application of CLNH with two further independent key words $K_{A_1}, K_{A_2} \in \{0,1\}^{64}$. The result $H$ is finally XORed with the hashed length to account for inputs of variable lengths, and is reduced to a 64-bit value.

In [34], the authors show that CLHASH is XOR-universal for messages of up to $b = 8m$ bytes, and $\epsilon$-AXU for messages of up to $N$ bytes.

**Theorem 6 (Lemma 9 in [34]).** *Let $N \geq 1$ denote the maximal message length in bytes, $m \geq 2$ be even, and $b = 8m$ the key size of* CLNH. *Then,* CLHASH *as defined above is $\epsilon$-AXU with*

$$\epsilon \leq \epsilon_{\mathrm{CLNH}[m]} + \epsilon_{\textsc{Poly}} + \epsilon_{\mathrm{CLNH}[2]} \leq \frac{1}{2^{64}} + \frac{N/b - 1}{2^{126}} + \frac{1}{2^{64}},$$

*where the terms stem from the facts that* CLNH$[m]$ *is an $\epsilon_{\mathrm{CLNH}[m]}$-AU, and the polynomial hash an $\epsilon_{\textsc{Poly}}$-AXU family of hash functions.*

The recommended values $N \leq 2^{64}$ and $b = 1024$ yield $\epsilon \leq 2.004/2^{-64}$. The construction requires $b + 40$ bytes of key material: $b$ bytes for CLNH, a 16-byte

value $K_P$ for the polynomial hash, two eight-byte values $K_A[1], K_A[2]$ for the final call to CLNH, and an eight-byte value $K_L$ for hashing the input length.

***Toeplitz Extension.*** To obtain a hash function with 128-bit security, one can process the same message twice under independent keys and concatenate the results. Doubling the key lengths of $K_P$, $K_A$, and $K_L$ increases their keys to 80 bytes. Since doubling the key length for CLNH would absurdly increase the key material, we use the Toeplitz extension [31,37] instead. Let $K_{i..j}$ be short for $K_i, \ldots, K_j$, $1 \le i \le j$. Given an $\epsilon$-AU family of hash functions $H : \{0,1\}^{mn} \times \{0,1\}^{mn} \to \{0,1\}^n$ which compresses an $m$-word input with an $m$-word key, one can derive a hash function $H^t : \{0,1\}^{(m+2t-2)n} \times \{0,1\}^{mn} \to \{0,1\}^{tm}$ by

$$H^t_{K_{1..(m+2t-2)}}(M) := H_{K_{1..m}}(M) \,\|\, H_{K_{3..(m+2)}}(M) \,\|\, \cdots \,\|\, H_{K_{(2t-1)..(m+2t-2)}}(M).$$

So, the $i$-th call to $H$ employs the key shifted by $2i-2$ words. In total, the key size increases slightly from $m$ to $m+2(t-1)$ words. We refer to the Toeplitz version of CLNH by $\mathrm{CLNH}^{\mathsf{T}}[m,t]$, and to that of CLHASH$[m]$ by $\mathrm{CLHASH}^{\mathsf{T}}[m,t]$. Algorithm 1 provides a specification. In total, $\mathrm{CLHASH}^{\mathsf{T}}[m,t]$ requires $(8m + 56t - 16)$ bytes of key material, which corresponds to $(8m + 96)$ bytes for $t = 2$.

**Definition 13 (Toeplitz** CLHASH**).** *Let $n = 64$, $t \ge 1$, $m \ge 2$ be even. Let $\mathcal{X} = \bigcup_{i=1}^{m/2} \{0,1\}^{2in}$. Let further $\mathcal{K}_N = \{0,1\}^{64m+128(t-1)}$, $\mathcal{K}_P = \{0,1\}^{128t}$, $\mathcal{K}_A = \{0,1\}^{128t}$, $\mathcal{K}_L = \{0,1\}^{64t}$, and $\mathcal{K} = \mathcal{K}_N \times \mathcal{K}_P \times \mathcal{K}_A \times \mathcal{K}_L$. The family of keyed hash functions $\mathrm{CLHASH}^{\mathsf{T}}[m,t] : \mathcal{K} \times \mathcal{X} \to \{0,1\}^{64t}$ is defined in Algorithm 1.*

**Theorem 7.** *For any fixed $n, t \ge 1$, and even $m \ge 2$, $\mathrm{CLNH}^{\mathsf{T}}[m,t]$ is $2^{-nt}$-AU on equal-length strings.*

The proof of Theorem 7 can be found in the full version of this paper.

**Theorem 8.** *Let $N \le 2^{64}$ be the maximal message length in bytes, $t \ge 1$, $m \ge 2$ be even, and $b = 8m$ the key length for* CLNH *in bytes. Then,* $\mathrm{CLHASH}^{\mathsf{T}}[m,t]$ *is an $\epsilon^t$-AXU family of hash functions with*

$$\epsilon \le \epsilon_{\mathrm{CLNH}[m]} + \epsilon_{\mathrm{POLY}} + \epsilon_{\mathrm{CLNH}[2]} \le \frac{1}{2^{64}} + \frac{N/b - 1}{2^{126}} + \frac{1}{2^{64}} \le \frac{3}{2^{64}}.$$

The proof of Theorem 8 follows from Theorem 7 and the fact that the keys for the individual iterations of polynomial, inner-product, and length hashing steps are chosen uniformly from their respective spaces and pairwise independently for each iteration. We can derive that $\mathrm{CLHASH}^{\mathsf{T}}[m,2]$ is $\epsilon$-AXU for $\epsilon \le 9/2^{128}$ when $m \ge 2$.

### 6.2 Constructing a PRF

Let $n, d \ge 1$, and $\mathcal{N}$, $\mathcal{H}$, $\mathcal{M}$ be as in Sect. 3. For brevity, we define $\mathcal{Y} := \{0,1\}^d \times \mathcal{N} \times \mathcal{H} \times \mathcal{M}$. Let ENCODE : $\mathcal{Y} \to \{0,1\}^*$ define an injective encoding function. Then, we can construct a PRF from the composition of ENCODE,

**Algorithm 2.** Encryption of nonce-based XOR-CTR, instantiated with a block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, with $n, k \geq 1$.

1: **function** XOR-CTR$[E].\mathcal{E}_K^N(M)$
2: $IV \leftarrow E_K(N)$
3: $m \leftarrow \lceil |M|/n \rceil$
4: $\kappa \leftarrow E_K(IV \oplus \langle 0 \rangle) \,\|\, \cdots \,\|\, E_K(IV \oplus \langle m-1 \rangle)$
5: **return** $C \leftarrow M \oplus \kappa[\text{first } |M| \text{ bits}]$

a family of $\epsilon$-AU hash functions $\mathcal{H}' = \{H' | H' : \{0,1\}^* \to \{0,1\}^n\}$, and a block cipher $E : \mathcal{K}_2 \times \{0,1\}^n \to \{0,1\}^n$, with independent keys $K_1 \in \mathcal{K}_1$ determining the hash function, and $K_2 \in \mathcal{K}_2$ for the cipher. We call the construction EHE$[\textsc{Encode}, \mathcal{H}', E] : \mathcal{Y} \to \{0,1\}^n$ (for *Encode-Hash-Encrypt*) and define it as

$$\text{EHE}[\textsc{Encode}, \mathcal{H}', E]_{K_1,K_2}(D, N, H, M) := E_{K_2}(\mathcal{H}'_{K_1}(\textsc{Encode}(D, N, H, M))).$$

We write EHE$[\mathcal{H}', E]$ or even EHE as short forms of EHE$[\textsc{Encode}, \mathcal{H}', E]$ when the components are clear from the context. The injective encoding excludes collisions between distinct inputs. From Theorem 2, and applying the PRF/PRP switching lemma, we can derive the following theorem.

**Theorem 9.** *Let* $\pi \twoheadleftarrow \mathsf{Perm}(\{0,1\}^n)$. *Further, let* EHE$[\textsc{Encode}, \mathcal{H}', \pi]$, $\mathcal{H}'$, *and* Encode *be defined as above. Let* $\mathbf{A}$ *be a computationally bounded adversary that asks at most* $q$ *queries of at most* $\ell$ *blocks and runs in time at most* $t$. *Then*

$$\mathbf{Adv}^{\text{PRF}}_{\text{EHE}[\textsc{Encode},\mathcal{H}',\pi]}(\mathbf{A}) \leq \binom{q}{2} \cdot \left(\frac{1}{2^n} + \epsilon\right).$$

### 6.3 Encryption

When starting counter-mode encryption from a random value and incrementing by modular addition, one has to either consider potential carry bits or to reduce the security by fixing a maximal message length. Wang et al. [50] proposed to replace modular addition by XOR, which avoids the need for concerning carry bits. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. We define XOR-CTR$[E] = (\mathcal{E}, \mathcal{D})$ as the nonce-based encryption scheme with encryption algorithm XOR-CTR$[E].\mathcal{E} : \{0,1\}^k \times \mathcal{N} \times \{0,1\}^* \to \{0,1\}^*$ and associated non-empty nonce-space $\mathcal{N}$, as defined in Algorithm 2.

We denote by XOR-CTR$[\pi, \pi']$ a version of XOR-CTR with two independent $n$-bit permutations $\pi$ and $\pi'$, where $\pi$ is used for encrypting the nonce and $\pi'$ for producing the keystream. Then, XOR-CTR$[\pi, \pi']$ is almost identical to the CTR2$[\pi, \pi']$ construction in [45], with the difference that the former replaces the addition of $IV$ and counter modulo $2^n$ by XOR. Since this change does not affect the probability of block-cipher inputs to repeat, the nE advantage of XOR-CTR is given by Theorem 10, which adapts Theorem 3 in [45].

**Theorem 10.** *Let* $\pi, \pi' \twoheadleftarrow \mathsf{Perm}(\{0,1\}^n) \times \mathsf{Perm}(\{0,1\}^n)$ *be independent permutations and* $\mathbf{A}$ *be a nonce-respecting* nE *adversary, which runs in time at most* $t$ *and poses at most* $q$ *queries to its oracles with at most* $\ell$ *blocks. Then*

$$\mathbf{Adv}^{\mathrm{nE}}_{\text{XOR-CTR}[\pi,\pi']}(\mathbf{A}) \leq \frac{\ell^2}{2^n}.$$

From the fact that encryption and decryption of XOR-CTR$[\pi, \pi']$ are identical operations, we can derive the following theorem.

**Theorem 11.** *There exists a reduction of a nonce-respecting* SRND *adversary* $\mathbf{A}$ *with access to two oracles on* XOR-CTR$[\pi, \pi']$ *to a nonce-respecting* nE *adversary* $\mathbf{A}'$ *on* XOR-CTR$[\pi, \pi']$ *such that*

$$\mathbf{Adv}^{\mathrm{SRND}}_{\text{XOR-CTR}[\pi,\pi']}(\mathbf{A}) \leq \mathbf{Adv}^{\mathrm{nE}}_{\text{XOR-CTR}[\pi,\pi']}(\mathbf{A}'),$$

*where both* $\mathbf{A}$ *and* $\mathbf{A}'$ *ask at most* $q$ *queries of at most* $\ell$ *blocks to their available oracle(s) and run in time* $O(t)$.

### 6.4 Instantiation of RIV

We instantiate $\text{RIV}_{F,\Pi}$ with EHE[Encode, $\mathcal{H}', E$] for $F$, with CLHASH$^{\mathsf{T}}[m, 2]$ as family of universal hash functions $\mathcal{H}'$, and XOR-CTR$[E]$ for $\Pi$, with the AES-128 as $E$. Algorithm 3 provides a specification. Our instantiation $\text{RIV}_{F,\Pi}$ expects a 128-bit user-supplied secret key $SK$, from which the remaining key material is derived by calling $E_{SK}(\cdot)$ iteratively in counter mode. The secret key is not used further. RIV uses $n = \tau = 128$, i.e., $n$-bit tags, and $n$-bit $IV$s for the counter mode. Moreover, the nonce space is fixed to 128 bits: $\mathcal{N} = \{0,1\}^n$. For $F$, it employs a four-bit domain separation, i.e., $d = 4$, and an injective encoding function Encode : $\{0,1\}^d \times \mathcal{N} \times \mathcal{H} \times \mathcal{M} \to \{0,1\}^*$, as defined in Algorithm 3. Header and message lengths are restricted to multiple of eight bits. The maximal number of header and message bytes to be encrypted under the same key are $2^{60}$ bytes each. So, the maximal number of bytes for RIV is less than $2^{62}$ bytes. We recommend that at most $2^{50}$ bytes be encrypted under the same key.

***Using a Single Key for the Block Cipher.*** There are four uses of the block cipher $E$ in RIV: in the first invocation of EHE, for encrypting the $IV$, for generating the keystream in XOR-CTR$[E]$, and in the second invocation of EHE. If four more calls to the AES key schedule would be tolerable, one could use four independent keys. Alternatively, we use a single key for the uses of $E$, and have to consider the security impact in the following theorem. Its proof can be found in the full version of this paper.

**Theorem 12.** *Let* $\text{RIV}_{F,\Pi}$ *be defined as in Algorithm 3. Let* $K_1, K_2 \twoheadleftarrow \mathcal{K}$ *be independent keys. We replace the calls to* $E$ *by independent random permutations* $\pi_1, \pi_2, \pi_3, \pi_4 \twoheadleftarrow \mathsf{Perm}(\{0,1\}^n)^4$. *Let* $\mathbf{A}$ *be a computationally bounded adversary that has access to three oracles* $\mathcal{O}_1$, $\mathcal{O}_2$, *and* $\mathcal{O}_3$ *for encryption, decryption, and leakage, respectively.* $\mathbf{A}$ *shall distinguish between a* real *setting of* $\text{RIV}_{F,\Pi}$ *as*

**Algorithm 3.** Definition of our instantiation $\mathrm{RIV}_{F,\Pi}$. Message and header lengths are restricted to multiple of eight bits, and nonces/IVs/tags are 128 bits: $n = \tau = 128$, and $d = 4$. Here, we leave the key size of $\mathrm{CLHASH}^{\mathsf{T}}[m, 2]$, $m$, as a parameter to study its impact on performance later.

```
101: function Ẽ_SK(N, H, M)
102:   (K1, K2) ← KeyGen(SK)
103:   IV ← EHE^1_{K1,K2}(N, H, M)
104:   C ← XOR-CTR[E].E_{K2}(IV, M)
105:   T ← EHE^2_{K1,K2}(N, H, C) ⊕ IV
106:   return (C, T)

201: function KeyGen(SK)
202:   K2 ← E_SK(⟨0⟩)
203:   κ ← (8m + 96)/16
204:   K1 ← E_SK(⟨1⟩) ‖ ··· ‖ E_SK(⟨κ⟩)
205:   return (K1, K2)

301: function EHE^D_{K1,K2}(N, H, X)
302:   Y ← Encode(D, N, H, X)
303:   return E_{K2}(H'_{K1}(Y))

401: function pad_n(X)
402:   if (|X| mod n = 0) then
403:     return X
404:   return X ‖ 0^{n−|X| mod n}

501: function D̃_SK(N, H, C, T)
502:   (K1, K2) ← KeyGen(SK)
503:   IV ← EHE^2_{K1,K2}(N, H, C) ⊕ T
504:   M ← XOR-CTR[E].D_{K2}(IV, C)
505:   IV' ← EHE^1_{K1,K2}(N, H, M)
506:   if (IV = IV') then
507:     return M
508:   return ⊥

601: function Encode(D, N, H, X)
602:   H̄ ← pad_128(H)
603:   X̄ ← pad_128(X)
604:   L̄ ← ⟨D⟩_d ‖ ⟨|H|/8⟩_60 ‖ ⟨|X|/8⟩_64
605:   return (H̄ ‖ N ‖ X̄ ‖ L̄)

701: function H'_{K1}(X)
702:   return CLHASH^T[m, 2]_{K1}(X)

801: function E_{K2}(X)
802:   return AES-128_{K2}(X)
```

*above with a single-keyed block cipher $E$, and $\mathrm{RIV}_{F,\Pi}$ which uses four independent uniformly chosen permutations $\pi^1, \pi^2, \pi^3, \pi^4 \leftarrow \mathsf{Perm}(\{0,1\}^n)$ with $\pi^1$ used in $\mathrm{EHE}^1$, $\pi^2$ used in $\mathrm{EHE}^2$, and $\pi^3$, $\pi^4$ used for $\mathrm{XOR\text{-}CTR}[\pi^3, \pi^4]$. $\mathbf{A}$ asks at most $q$ queries of at most $\ell$ blocks and runs in time at most $t$. Then, we can upper bound the distinguishing advantage of $\mathbf{A}$ by*

$$16.5\ell^2 \cdot \max\{\epsilon, 1/2^n\} + \mathbf{Adv}^{\mathrm{PRP}}_{E}(\ell + 3q, O(t)).$$

**Theorem 13.** *Let $d = 4$, $n = \tau = 128$, and $m \geq 2$ be even. Let $\mathrm{RIV}_{F,\Pi}$ be as given in Algorithm 3 and let $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ be computationally bounded* IND-CPA, INT-CTXT, *and* ERR-CCA *adversaries on $\mathrm{RIV}_{F,\Pi}$, respectively, which run each in time at most $t$ and ask at most $q$ queries of at most $\ell$ blocks in total. Then, it holds that*

$$\mathbf{Adv}^{\text{IND-CPA}}_{\mathrm{RIV}_{F,\Pi}}(\mathbf{A}) \leq \frac{2q^2 + \ell^2}{2^n} + q^2\epsilon + \delta_E,$$

$$\mathbf{Adv}^{\text{INT-CTXT}}_{\mathrm{RIV}_{F,\Pi}}(\mathbf{A}) \leq \frac{2q^2 + q + \ell^2}{2^n} + q^2\epsilon + \delta_E,$$

$$\mathbf{Adv}^{\text{ERR-CCA}}_{\mathrm{RIV}_{F,\Pi}}(\mathbf{A}) \leq \frac{8q^2 + 2q + 2\ell^2}{2^n} + 2q^2\epsilon + \delta_E,$$

*where $\delta_E = 16.5\ell^2 \cdot \epsilon + \mathbf{Adv}^{\mathrm{PRP}}_{E}(\ell + 3q, O(t))$ and $\epsilon \leq 9/2^{128}$.*

The proof follows from Theorems 5, 8, 9, and 11, and those of the lemmata from Sect. 5 that can be found in the full version of this paper.

### 6.5 Performance of RIV

We implemented reference and optimized versions of RIV in C.[5] Since the default key length for one iteration CLNH of $b = 1024$ bytes (which corresponds to CLHASH$^{\mathsf{T}}$[128, 2]) appeared high, we tested also a variant with a smaller key size of $b = 256$ bytes for CLNH (CLHASH$^{\mathsf{T}}$[32, 2]). Table 1 summarizes the results of our benchmarks. Our code was compiled using gcc v4.9.3 with options `-O3 -maes -mavx2 -mpclmul -march=native`, and run on (1) an Intel Core i5-4200M (Haswell) at 2.50 GHz, and (2) on an Intel i5-5200 (Broadwell) at 2.20 GHz, both with the TurboBoost, SpeedStep, and HyperThreading technologies *disabled*. For measuring, we used the median of 10000 encryptions, omitting the cost for key setup, using the `rdtsc` instruction.

Our results show that RIV can run at less than 1.5 cpb on Haswell. Interestingly, a *SIV-like* reduced version of RIV, which is an easily obtained byproduct that simply omits the second call to $F$, represents a performant MRAE scheme with $\leq 1.04$ cpb. This is slightly faster than the $4867/4096 \approx 1.17$ cpb reported for the manually assembly-optimized AES-GCM-SIV [22] and 1.06 cpb for the version of MRO with four-round BLAKE2b in [20], concerning messages of at least four KiB length on Haswell. Clearly, the reported performance of AEZv4 of about 0.7 cpb is unrivaled. Though, our construction provides a slightly higher security margin. Moreover, the security of AEZv4 bases on heuristic assumptions on four-round AES.

**Table 1.** Performance results on Intel Haswell and Broadwell, respectively, in cycles per byte for the encryption with optimized implementations of RIV and a reduced version, which omits the second call to $F$. $b$ denotes the key length for CLNH in bytes. Details regarding our setup are provided in the text.

| Platform | Instance | $b$ | Message length (bytes) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 128 | 256 | 512 | 1024 | 2048 | 4096 | 8192 | 16384 |
| Haswell | RIV | 256 | 3.81 | 2.78 | 2.14 | 1.81 | 1.62 | 1.48 | 1.40 | 1.37 |
| | RIV | 1024 | 3.53 | 2.13 | 1.81 | 1.49 | 1.37 | 1.29 | 1.25 | 1.22 |
| | RIV (2-pass) | 256 | 1.71 | 1.40 | 1.26 | 1.14 | 1.08 | 1.04 | 1.01 | 0.99 |
| | RIV (2-pass) | 1024 | 2.20 | 1.60 | 1.17 | 1.08 | 1.01 | 0.97 | 0.94 | 0.92 |
| Broadwell | RIV | 256 | 3.16 | 2.41 | 1.84 | 1.49 | 1.38 | 1.26 | 1.20 | 1.15 |
| | RIV | 1024 | 3.13 | 2.11 | 1.56 | 1.34 | 1.16 | 1.09 | 1.04 | 1.02 |
| | RIV (2-pass) | 256 | 2.16 | 1.67 | 1.30 | 1.09 | 1.03 | 0.95 | 0.92 | 0.90 |
| | RIV (2-pass) | 1024 | 2.19 | 1.50 | 1.14 | 1.01 | 0.92 | 0.86 | 0.84 | 0.82 |

[5] Our code is open to the public domain: https://github.com/medsec/riv.

## 7 Conclusion

This work described a modular framework RIV for the construction of provably secure subtle AE schemes by extending the SIV framework from two to three passes. The obvious strength of RIV resides in the simplicity of its structure: it allows a straight-forward transformation of existing SIV-based constructions into subtle AE schemes. We proved the security in the standard model under notions that strive for ideal security goals; a further step could be to prove *achievable* security in the RAE setting with fixed stretch. Moreover, since the generic RIV construction bases only on PRF assumptions, this leaves open the possibility for proofs in the indifferentiability setting [40]. RIV is slightly less efficient than earlier STPRP constructions, i.e., it employs three additional calls to an $n$-bit PRP, compared to e.g., a single call in HCTR-based [50] constructions. Since the use of a nonce-based encryption scheme $(\mathcal{E}, \mathcal{D})$ poses only the requirement on the $IV$ to be a nonce, it might look to be sufficient to have two calls to universal hash functions instead of to calls to a PRF $F$. Yet, at least the outputs from the first invocation of $F$, $F^1_{K_1}(\cdot,\cdot,\cdot)$ must be unpredictable in order to prevent leaking information about the message in the tag. A potential future work can be to further study reductions of the design to target even higher efficiency. Nevertheless, we proposed an instantiation that is highly efficient on current x64 platforms and avoids the weak-key issues that were reported for GHASH-based polynomials in HCTR instantiations [49].

**Acknowledgments.** We thank all reviewers of the FSE 2016 for their helpful comments, Daniel Lemire and Owen Kaser for valuable notes on CLHASH, and Guy Barwell, Daniel Page, and Martijn Stam for insights into their work on subtle authenticated encryption. Christian Forler received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement no. 307952 and from the Silicon Valley Community Foundation under the Cisco Systems project *Misuse-Resistant Authenticated Encryption for Complex and Low-End Systems* (MIRACLE).

## References

1. Abed, F., Fluhrer, S., Forler, C., List, E., Lucks, S., McGrew, D., Wenzel, J.: Pipelineable on-line encryption. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS, vol. 8540, pp. 205–223. Springer, Heidelberg (2015)
2. Anderson, R.J., Biham, E.: Two practical and provably secure block ciphers: BEAR and LION. In: Gollmann, D. (ed.) FSE 1996. LNCS, vol. 1039, pp. 113–120. Springer, Heidelberg (1996)
3. Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mendel, F., Mennink, B., Mouha, N., Wang, Q., Yasuda, K.: PRIMATES (2014). http://competitions.cr.yp.to/caesar-submissions.html
4. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: How to securely release unverified plaintext in authenticated encryption. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 105–125. Springer, Heidelberg (2014)

5. Badertscher, C., Matt, C., Maurer, U., Rogaway, P., Tackmann, B.: Robust authenticated encryption and the limits of symmetric cryptography. In: Groth, J., et al. (eds.) IMACC 2015. LNCS, vol. 9496, pp. 112–129. Springer, Heidelberg (2015). doi:10.1007/978-3-319-27239-9_7
6. Bahack, L.: Julius (2014). http://competitions.cr.yp.to/caesar-submissions.html
7. Barwell, G., Page, D., Stam, M.: Rogue decryption failures: reconciling AE robustness notions. In: Groth, J., et al. (eds.) IMACC 2015. LNCS, vol. 9496, pp. 94–111. Springer, Heidelberg (2015). doi:10.1007/978-3-319-27239-9_6
8. Bellare, M., Namprempre, C.: Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, p. 531. Springer, Heidelberg (2000)
9. Bellare, M., Rogaway, P.: Encode-then-encipher encryption: how to exploit nonces or redundancy in plaintexts for efficient cryptography. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 317–330. Springer, Heidelberg (2000)
10. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 389–407. Springer, Heidelberg (2004)
11. Bernstein, D.J.: Polynomial evaluation and message authentication (2007). http://cr.yp.to/papers, permanent ID: b1ef3f2d385a926123e1517392e20f8c, 2
12. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: Using Keccak technology for AE: Ketje, Keyak and more. In: SHA-3 2014 Workshop, UC Santa Barbara, 22 August 2014
13. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: fast and secure message authentication. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 216–233. Springer, Heidelberg (1999)
14. Boesgaard, M., Christensen, T., Zenner, E.: Badger – a fast and provably secure MAC. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 176–191. Springer, Heidelberg (2005)
15. Boldyreva, A., Degabriele, J.P., Paterson, K.G., Stam, M.: On symmetric encryption with distinguishable decryption failures. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 367–390. Springer, Heidelberg (2014)
16. Chakraborty, D., Sarkar, P.: On modes of operations of a block cipher for authentication and authenticated encryption. In: IACR Cryptology ePrint Archive, 2014/627 (2014)
17. Coron, J.-S., Dodis, Y., Mandal, A., Seurin, Y.: A domain extender for the ideal cipher. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 273–289. Springer, Heidelberg (2010)
18. Dai, W., Krovetz, T.: VHASH security. In: IACR Cryptology ePrint Archive: 2007/338 (2007)
19. Fleischmann, E., Forler, C., Lucks, S.: McOE: a family of almost foolproof on-line authenticated encryption schemes. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 196–215. Springer, Heidelberg (2012)
20. Granger, R., Jovanovic, P., Mennink, B., Neves, S.: Improved masking for tweakable blockciphers with applications to authenticated encryption. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 263–293. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49890-3_11
21. Kounavis, E.: Efficient implementation of the Galois Counter Mode using a carryless multiplier and a fast reduction algorithm. Inf. Process. Lett. **110**(14–15), 549–553 (2010)

22. Gueron, S., Lindell, Y.: GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM Conference on Computer and Communications Security, pp. 109–119. ACM (2015)
23. Halevi, S., Rogaway, P.: A parallelizable enciphering mode. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 292–304. Springer, Heidelberg (2004)
24. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 15–44. Springer, Heidelberg (2015)
25. Hoang, V.T., Reyhanitabar, R., Rogaway, P., Vizár, D.: Online authenticated-encryption and its nonce-reuse misuse-resistance. In: Gennaro, R., Robshaw, M., et al. (eds.) CRYPTO (1). LNCS, vol. 9215, pp. 493–517. Springer, Heidelberg (2015)
26. ISO/IEC. 19772: 2009, Information technology - Security techniques - Authenticated Encryption (2009)
27. Iwata, T., Yasuda, K.: BTM: a single-key, inverse-cipher-free mode for deterministic authenticated encryption. In: Jacobson Jr., M.J., Rijmen, V., Reihaneh, S.-N. (eds.) Selected Areas in Cryptography, pp. 313–330 (2009)
28. Iwata, T., Yasuda, K.: HBS: a single-key mode of operation for deterministic authenticated encryption. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 394–415. Springer, Heidelberg (2009)
29. Jean, J., Nikolić, I., Peyrin, T.: Deoxys (2014). http://competitions.cr.yp.to/caesar-submissions.html
30. Jean, J., Nikolić, I., Peyrin, T.: Joltik (2014). http://competitions.cr.yp.to/caesar-submissions.html
31. Krawczyk, H.: LFSR-based hashing and authentication. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 129–139. Springer, Heidelberg (1994)
32. Krovetz, T.: Message authentication on 64-bit architectures. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 327–341. Springer, Heidelberg (2007)
33. Krovetz, T.: HS1-SIV (2014). http://competitions.cr.yp.to/caesar-submissions.html
34. Lemire, D., Kaser, O.: Faster 64-bit universal hashing using carry-less multiplications. J. Crypt. Eng. **5**, 1–15 (2015)
35. Luby, M., Rackoff, C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM J. Comput. **17**(2), 373–386 (1988)
36. Lucks, S.: Faster Luby-Rackoff ciphers. In: Gollmann, D. (ed.) FSE 1996. LNCS, vol. 1039, pp. 183–203. Springer, Heidelberg (1996)
37. Mansour, Y., Nisan, N., Tiwari, P.: The computational complexity of universal hashing. In: Ortiz, H. (ed.) STOC, pp. 235–243. ACM (1990)
38. Maurer, U.: Constructive cryptography – a new paradigm for security definitions and proofs. In: Mödersheim, S., Palamidessi, C. (eds.) TOSCA 2011. LNCS, vol. 6993, pp. 33–56. Springer, Heidelberg (2012)
39. Maurer, U., Renner, R.: Abstract cryptography. In: Chazelle, B. (ed.) ICS, pp. 1–21. Tsinghua University Press, Beijing (2011)
40. Maurer, U.M., Renner, R.S., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)
41. Namprempre, C., Rogaway, P., Shrimpton, T.: Reconsidering generic composition. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 257–274. Springer, Heidelberg (2014)

42. Naor, M., Reingold, O.: On the construction of pseudorandom permutations: Luby-Rackoff revisited. J. Cryptol. **12**(1), 29–66 (1999)
43. Reyhanitabar, R., Vaudenay, S., Vizár, D.: Misuse-resistant variants of the OMD authenticated encryption mode. In: Chow, S.S.M., Liu, J.K., Hui, L.C.K., Yiu, S.M. (eds.) ProvSec 2014. LNCS, vol. 8782, pp. 55–70. Springer, Heidelberg (2014)
44. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM Conference on Computer and Communications Security, pp. 98–107 (2002)
45. Rogaway, P.: Nonce-based symmetric encryption. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 348–359. Springer, Heidelberg (2004)
46. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (2006)
47. Shrimpton, T., Terashima, R.S.: A modular framework for building variable-input-length tweakable ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 405–423. Springer, Heidelberg (2013)
48. Stinson, D.R.: Universal hashing and authentication codes. Des. Codes Crypt. **4**(4), 369–380 (1994)
49. Sun, Z., Wang, P., Zhang, L.: Weak-key and related-key analysis of hash-counter-hash tweakable enciphering schemes. In: Foo, E., Stebila, D. (eds.) ACISP 2015. LNCS, vol. 9144, pp. 3–19. Springer, Heidelberg (2015)
50. Wang, P., Feng, D., Wu, W.: HCTR: a variable-input-length enciphering mode. In: Feng, D., Lin, D., Yung, M. (eds.) CISC 2005. LNCS, vol. 3822, pp. 175–188. Springer, Heidelberg (2005)
51. Winograd, S.: A new algorithm for inner product. IEEE Trans. Comput. **100**(7), 693–694 (1968)

http://www.springer.com/978-3-662-52992-8

Fast Software Encryption
23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers
Peyrin, Th. (Ed.)
2016, XI, 592 p. 105 illus., Softcover
ISBN: 978-3-662-52992-8