Pipelineable On-Line Encryption

Farzaneh Abed[1], Scott Fluhrer[2], John Foley[2], Christian Forler[1,⋆]
Eik List[1], Stefan Lucks[1,⋆⋆], David McGrew[2], Jakob Wenzel[1]

[1] Bauhaus-Universität Weimar, Germany, [2] Cisco Systems, USA
farzaneh.abed@uni-weimar.de, christian.forler@uni-weimar.de, eik.list@uni-weimar.de, stefan.lucks@uni-weimar.de, jakob.wenzel@uni-weimar.de, sfluhrer@cisco.com, foleyj@cisco.com, mcgrewd@cisco.com

Abstract. Correct authenticated decryption requires the receiver to buffer the decrypted message until the authenticity check has been performed. In high-speed networks, which must handle large message frames at low latency, this behavior becomes practically infeasible. This paper proposes CCA-secure on-line ciphers as a practical alternative to AE schemes since the former provide some defense against malicious message modifications. Unfortunately, all published on-line ciphers so far are either inherently sequential, or lack a CCA-security proof.
This paper introduces POE, a family of on-line ciphers that combines provable security against chosen-ciphertext attacks with pipelineability to support efficient implementations. POE combines a block cipher and an ϵ-AXU family of hash functions. Different instantiations of POE are given, based on different universal hash functions and suitable for different platforms. Moreover, this paper introduces POET, a provably secure on-line AE scheme, which inherits pipelineability and chosen-ciphertext-security from POE and provides additional resistance against nonce-misuse attacks.

Keywords: on-line cipher, chosen-ciphertext security, authenticated encryption

1 Introduction

Authenticated Encryption (AE) schemes (such as EAX [7], GCM [32], OCB [29], etc.) perform an authentication check on the entire ciphertext before they output a decrypted message. This practice is inherent in the idea of authenticated encryption and part of its strength. However, it is incompatible with settings that pose demanding performance requirements (e.g., high speed, low latency, long messages).

⋆ The research leading to these results received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement no. 307952.

⋆⋆ A part of this research was done while Stefan Lucks was visiting the National Institute of Standards and Technologies (NIST), during his sabbatical.

One example for such settings are Optical Transport Networks (OTNs) [25], in which the links between multiple network channels must be capable of transmitting, multiplexing, and switching between immense data streams in a fast and secure manner. OTNs are characterized by high throughput rates of up to 100 Gbps, low latencies in the order of a few clock cycles, and large message frames of up to 64 kB. At that size, a mode of operation of a 128-bit block cipher would require over 4,096 clock cycles to complete a decryption – which exceeds the allowed latency in OTN systems by far.

In such uses of AE, implementations have to pass along (part of) a decrypted message before validating its authenticity; if the message later turns out to be invalid, this fact will be discovered and reported, but only after some information has been leaked. The literature calls this practice *decryption misuse* [20], and describes severe vulnerabilities for conventional AE schemes. A chosen-ciphertext adversary can exploit it to determine unknown plaintexts, or to introduce forged message fragments that may get passed to the application and are processed before the authentication check is completed. As a consequence, common existing AE schemes do not suit well in this environment. To overcome this issue, this work considers authenticated encryption schemes that provide robustness against decryption misuse through *on-line chosen-ciphertext security* (OPRP-CCA) [4]. Implementations of AE schemes that allow decryption misuse abound, even when latency is not a consideration. For example, many software libraries provide access to encryption and decryption operations through a stream-oriented interface that consists of functions for initialization, updating, and finalization. In these interfaces the decrypt-update function can be called multiple times.[1] *Every* invocation of this function performs decryption misuse, because it releases the would-be plaintext before completing the authentication check. This type of interface is incompatible with existing authenticated encryption schemes. But its use is widespread, well-established and will not easily go away.

Decryption-Misuse Resistance. An encryption scheme is called *non-malleable* if any change to a ciphertext causes its entire post-decryption plaintext to be pseudorandom [19]. We call such a scheme *decryption-misuse-resistant* since the decryption of manipulated ciphertext results in uncontrollable random noise. Unfortunately, non-malleability and on-line encryption are mutually exclusive: if an adversary manipulates the i-th block of a ciphertext, an on-line encryption scheme leaves the previous $(i-1)$ blocks unchanged. But OPRP-CCA-security is the strongest form of non-malleability and decryption-misuse resistance an on-line cipher can provide: if an adversary manipulates the i-th block, all plaintext blocks starting from the i-th one will become pseudorandom.

The concept of decryption-misuse-resistant AE schemes is controversial. During the *Dagstuhl Seminar* on Symmetric Cryptography in January 2014 some researchers were worried about the risk of *advertising* decryption-misuse resistance as a feature for AE schemes since it could *invite* application programmers

[1] For example, see the OpenSSL `EVP_DecryptUpdate` function [45].

to improperly implement authenticated decryption. Of course, misuse must be avoided where possible, e.g., by user education. Nevertheless, decryption misuse is common in practice,[2] as our example of OTNs illustrates. The choice for the cryptograph is to either deal with decryption misuse, or to abandon AE completely.

Support For Intermediate Tags. Beyond limiting the harm of decryption misuse OPRP-CCA-secure on-line ciphers allow another desirable feature: *Intermediate tags* [8] allow the receiver to early detect if parts of a decrypted message are invalid – which is handy when authenticating large messages. They can be integrated easily into an OPRP-CCA-secure on-line cipher by adding somne form of well-formed redundancy (e.g., fixed constants or non- cryptographic checksums) to the plaintexts. For example, the headers of IP, TCP, or UDP [38,39,37] packets already contain a 16-bit checksum each, which is *verified* by the receiver and/or network routers. In OTNs, a single 64-kB message frame consists of multiple IP packets. Due to the low-latency constraints, receiving routers cannot buffer incoming messages until the authenication check has finished and must forward the first packets to their destination. However, they can test the packets' checksums to detect forgery attempts early. Hence, OPRP-CCA-security ensures that false TCP/IP packets only pass with probability of at most 2^{-16}.

Previous Work and Contributions. An ideal on-line cipher should be both IND-CCA-secure *and* non- sequential, i.e., parallelizable or pipelineable.[3] Already in 1978 Campbell published an early on-line cipher, called *Infinite Garble Extension* (IGE), which is far from complying with current security goals. In 2000 Knudsen [27] proposed his *Accumulated Block Chaining* (ABC) mode. In their landmark paper from 2001 Bellare et al. [4] coined the term of and security notions for on-line ciphers, and presented two instances, HCBC-1 and HCBC-2, based on the combination of a block cipher and a keyed hash function. Both constructions are inheritly sequential – HCBC-2 was slightly slower than HCBC-1, but provided additional IND-CCA-security. In 2002 Rivest, Liskov and Wagner [30,31] presented a non-sequential, tweakable on-line cipher, called TIE. However, TIE could not provide CCA-security due to a counter-based tweak input. In 2003 Halevi and Rogaway [23] proposed the *EME* approach (encryption-mix-encryption), which has inspired several on-line cipher designs since then. EME is a symmetric design concept that consists of five layers: an initial whitening step, an ECB layer, a linear mixing, a second ECB layer, and a final whitening. In 2004 Boldyrea and Taseombut [11] proposed security notions for on-line authenthentication ciphers, and the *HPCBC* mode as an instantiation. In 2007 and 2008 Nandi proposed further on-line ciphers similar to that of Bellare et

[2] As is nonce misuse: considering security under nonce-misuse has been a novelty a few years ago [41], but has become an established design goal nowadays.

[3] We call an operation f *pipelineable* if it can be split into multiple parts $f = f_2 \circ f_1$, s.t. f_1 can process the $(i+1)$-th input block before f_2 has finished processing the i-th block.

	Sequential	Non-Sequential
CCA-insecure	ABC [27], CBC [35], CFB [35], HCBC-1 [4], IGE [12], OFB [35], TC [31], TC1 [42]	COPE [2], CTR [18], ECB [35], TIE [31], XTS [24]
CCA-secure	APE(X) [3], CMC [22], HCBC-2 [4], MCBC [34], McOE [20], MHCBC [34], TC2/3 [42]	**POE**

Table 1. Classification of on-line encryption schemes.

al. [33,34]. In the same year the IEEE standardized the XTS [24] mode of operation for disk encryption; however, which also lacked CCA-security. In 2011 Rogaway and Zhang [42] described methods to construct secure on-line ciphers from tweakable block ciphers. However, it is easy to see that all mentioned schemes until here are either inherently sequential or CCA-insecure. Table 1 shows a summarized classification.

Contribution. This paper introduces the *Pipelineable On-line Encryption* (POE, hereafter) family of on-line ciphers, which consists of an ECB layer that is wrapped by two chaining layers of a keyed family of ϵ-AXU hash functions. The resulting construction is provably IND-CCA-secure and pipelineable, i.e., POE allows to process neighboring input blocks efficiently. To address different platforms, this work proposes three instantiations of POE, based on the AES as cipher and different families of universal hash functions. Furthermore, we show that POE can be easily transformed into an OPRP-CCA-secure, robust on-line AE (OAE) scheme, called *Pipelineable On-line Encryption with Tag* (POET hereafter), using well-studied methods from [20].

Recent Related Work. To the best of our knowledge, only four nonce-misuse-resistant OAE schemes were published prior to this work:[4] (1) McOE [20], (2) APE(X) [3], (3) COPA [2], and (4) ELmE [16]. McOE is a TC3-like design that was introduced at FSE 2012, and pioneered nonce-misuse resistance as a considerable feature for OAE schemes; APE(X), COPA, and ELmE are recent designs, where APE(X) bases on the Sponge, and COPA as well as ELmE on the EME design. McOE and APE(X) provide OPRP-CCA-security, but work inherently sequential, COPA and ELmE are parallelizable, and may outperform POET when running on high- end hardware or multi-core systems. However, the EME structure implies that both require two block-cipher calls for each message block, whereas POE and POET employ only a single cipher and two hash-function calls. Hence, we expect POET to perform better than EME-based

[4] Regarding the state *before* the CAESAR submission deadline. The research inpired by the CAESAR competition brought multiple further constructions that can be added to this list, including but not limited to COBRA, ELmD, or AEZ.

designs on medium- and low-end systems with few cores and no native AES instructions. Moreover, we illustrate in Appendix B that EME- based designs lose the OPRP-CCA-security in the decryption-misuse setting, which disqualifies COPA and ELmE for the OTN application scenario. More generally, Datta and Nandi [17] showed recently that EME constructions with linear mixing can not provide IND-CCA-security. Therefore, POET represents the first non-sequential OAE scheme with resistance against *both* nonce *and* decryption misuse.

Outline. The remainder of this work is structured as follows. Section 2 recalls the preliminary information about universal hash functions, on-line ciphers, and AE schemes that is necessary for this work. In Section 3, we propose the POE family of on-line ciphers and prove its security against chosen-plaintext and chosen-ciphertext attacks. Thereupon, Section 4, introduces POET, and provides a proof for the security against chosen-ciphertext attacks. Section 5 proposes three practical instantiations for POE and POET. Finally, we draw a conclusion in Section 6.

2 Preliminaries

This section revisits the well- known definitions of universal hash-function families from Carter and Wegman [14,44], as well as notions for on-line ciphers from Bellare et al. [4,5]. Prior, Table 2 summarizes the general notions.

N	Nonce (initial value)
M	Plaintext Message
C	Ciphertext
K	User-given secret key
$\|X\|$	Length of X in bits
n	Block length in bits
k	Key length in bits
X_i	i-th block of a value X
$X \mid\mid Y$	Concatenation of two values X and Y
$\mathcal{X}$	Set X
$X \xleftarrow{\$} \mathcal{X}$	X is a uniformly at random chosen sample from $\mathcal{X}$.

Table 2. Notions used throughout this paper.

2.1 Notions For Universal Hash Functions.

Definition 1 (ϵ-Almost-(XOR-)Universal Hash Functions). *Let $m, n \geq 1$ be integers. Let $\mathcal{H} = \{H : \{0,1\}^m \to \{0,1\}^n\}$ be a family of hash functions. We call $\mathcal{H}$* ϵ-almost-universal *(ϵ-AU) if for all $X, X' \in \{0,1\}^m$, $X \neq X'$:*

$$\Pr_H[H \xleftarrow{\$} \mathcal{H} : \ H(X) = H(X')] \leq \epsilon.$$

We call $\mathcal{H}$ ϵ-almost-XOR-universal *(ϵ-AXU) if for all* $X, X' \in \{0,1\}^m$, $Y \in \{0,1\}^n$, $X \neq X'$*:*

$$\Pr_H[H \xleftarrow{\$} \mathcal{H} : \ H(X) \oplus H(X') = Y] \leq \epsilon.$$

Boesgaard et al. [10] showed that an ϵ-AXU family of hash functions can be reduced to a family of ϵ-AU hash functions by the following theorem:

Theorem 1 (Theorem 3 from [10]). *Let* $m, n \geq 1$ *be integers. Let* $\mathcal{H} = \{H : \{0,1\}^m \to \{0,1\}^n\}$ *be a family of ϵ-AXU hash functions. Then, the familiy* $\mathcal{H}' = \{H' : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n\}$ *with* $H'(X,Y) = H(X) \oplus Y$ *is ϵ-AU.*

2.2 Notions For On-Line Ciphers.

Block Ciphers. A block cipher is a keyed family of n-bit permutations $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ which takes a k-bit key K and an n-bit message M and outputs an n-bit ciphertext C. We define $\texttt{Block}(k,n)$ as the set of all (k,n)-bit block ciphers for $n > 0$. For any $E \in \texttt{Block}(k,n)$ and a fixed key $K \in \{0,1\}^k$, the encryption of a message M is defined by $E_K(M)$, and the decryption is defined as the inverse function, i.e., $E_K^{-1}(M)$. For any key $K \in \{0,1\}^k$, it applies that $E_K^{-1}(E_K(M)) = M$.

Definition 2 (On-line Cipher). *Let* $k, n \geq 1$ *be integers and let* $\Gamma : \{0,1\}^k \times (\{0,1\}^n)^* \to (\{0,1\}^n)^*$ *be a keyed family of n-bit permutations which takes a k-bit key K and a message M of an arbitrary number of n-bit blocks, and outputs a ciphertext C consisting of the same number of n-bit blocks as M. We call Γ an* on-line cipher *iff the encryption of message block M_i, for all $i \in [1, |M|/n]$, depends only on the blocks $M_1, \ldots, M_i$.*

A secure cipher should behave like a random permutation. It is easy to see that on-line ciphers cannot guarantee this property since the encryption of message block M_i does not depend on M_{i+1}. The on-line behavior implies that two messages M, M' that share an m-block common prefix are always encrypted to two ciphertexts C, C' that also share an m-block common prefix. Hence, an on-line cipher Γ is called secure iff no ciphertext reveals any further information about a plaintext than its length and the *longest common prefix* with previous messages. For a formal definition of the longest common prefix of two messages, we refer to [20].

Definition 3 (On-line Permutation). *Let* $i, j, \ell, n \geq 1$ *be integers. Let* $F_i : (\{0,1\}^n)^i \to \{0,1\}^n$ *be a family of indexed n-bit permutations, i.e., for a fixed index* $j \in (\{0,1\}^n)^{i-1}$ *it applies that* $F_i(j, \cdot)$ *is a permutation. We define an n-bit on-line permutation* $P : (\{0,1\}^n)^\ell \to (\{0,1\}^n)^\ell$ *as a composition of ℓ permutations* $F_1 \cup F_2 \cup \ldots \cup F_\ell$*. An ℓ-block message* $M = (M_1, \ldots, M_\ell)$ *is mapped to an ℓ-block output* $C = (C_1, \ldots, C_\ell)$ *by*

$$C_i = F_i(M_1 \ || \ \ldots \ || \ M_{i-1}, M_i), \quad \forall i \in [1, \ell].$$

Remark 1. For any two ℓ-block inputs M, M', with $M \neq M'$, that share an exactly m-block common prefix $M_1 \,||\, \ldots \,||\, M_m$, the corresponding outputs $C = P(M), C' = P(M')$ satisfy $C_i = C'_i$ for all $i \in [1, m]$ and $m \leq \ell$. However, it applies that $C_{m+1} \neq C'_{m+1}$, and all further blocks C_i, C'_i, with $i \in [m+2, \ell]$, are likely to be different.

In the following, we denote by OPerm_n the set of all n-bit on-line permutations. Furthermore, we denote by $P \xleftarrow{\$} \mathsf{OPerm}_n$ that P is chosen as a random on-line permutation. Note that a random on-line permutation can be efficiently implemented by lazy-sampling.

On-Line Authenticated Encryption Scheme (With Associated Data). An authenticated encryption scheme is a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. $\mathcal{K}$ denotes a key-generation procedure that returns a randomly chosen key K; the encryption algorithm $\mathcal{E}_\mathcal{K}(H, M)$ and its inverse decryption algorithm $\mathcal{D}_\mathcal{K}(H, C, T)$ are deterministic algorithms, where H denotes the header, M the message, T the authentication tag, and C the ciphertext, with $H, M, C \in (\{0,1\}^n)^*$ and $T \in \{0,1\}^n$. We define that the final header block is a nonce. $\mathcal{E}$ always outputs a ciphertext C, and $\mathcal{D}$ outputs either the plaintext M that corresponds to C, or $\perp$ if the authentication tag T is invalid. Note that we call an authenticated encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ on-line if $\mathcal{E}$ is an on-line cipher and $\mathcal{D}$ is its inverse operation.

3 The On-Line Cipher POE

This section introduces the POE family of on-line ciphers and shows that it is secure against chosen-plaintext and chosen-ciphertext attacks.

3.1 Definition of POE

Definition 4 (POE). *Let $k, n \geq 1$ be integers, $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ a block cipher, and $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ a family of keyed ϵ-AXU hash functions. Furthermore, let $F_i : \{0,1\}^{ni} \to \{0,1\}^n$ be $i\epsilon$-AXU family of hash functions defined as follows:*

$$F_0 = F(1); \quad F_i(M) = F(F_{i-1}(M_1, \ldots, M_{i-1}) \oplus M_i) \qquad i \in \mathbb{N}^+.$$

Let $K, K_1, K_2 \in \{0,1\}^k$ denote three pair-wise independent keys. Then, we define the encryption of POE and its inverse as shown in Algorithm 1.

A schematic illustration of the encryption algorithm is given in Figure 1.

3.2 Security Notions for On-Line Ciphers

The IND-SPRP-security of a block cipher E is defined by the success probability of an adversary $\mathcal{A}$ to distinguish the output of E, E^{-1} from that of an n-bit random permutation π.

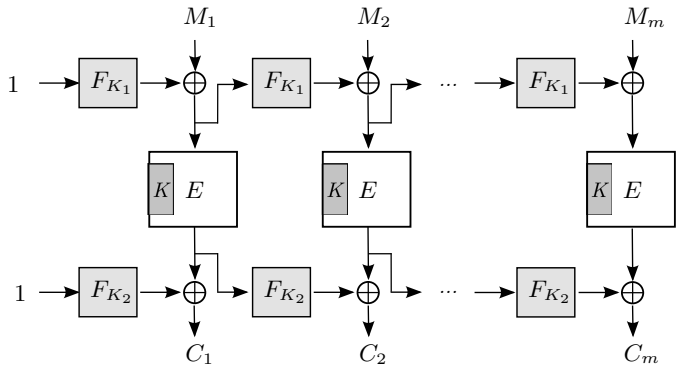

Fig. 1. The encryption process for an m-block message M with POE.

Algorithm 1 Procedures **Encrypt** and **Decrypt** for POE.

Encrypt(M)	**Decrypt**(C)
1: $m \leftarrow \lvert M\rvert/n$, $X_0 \leftarrow 1, Y_0 \leftarrow 1$	11: $m \leftarrow \lvert C\rvert/n$, $X_0 \leftarrow 1, Y_0 \leftarrow 1$
2: **for** $i = 1, \ldots, m$ **do**	12: **for** $i = 1, \ldots, m$ **do**
3: $\quad X_i \leftarrow F_{K_1}(X_{i-1}) \oplus M_i$	13: $\quad Y_i \leftarrow F_{K_2}(Y_{i-1}) \oplus C_i$
4: $\quad Y_i \leftarrow E_K(X_i)$	14: $\quad X_i \leftarrow E_K^{-1}(Y_i)$
5: $\quad C_i \leftarrow F_{K_2}(Y_{i-1}) \oplus Y_i$	15: $\quad M_i \leftarrow F_{K_1}(X_{i-1}) \oplus X_i$
6: **end for**	16: **end for**
7: **return** $(C_1 \parallel \ldots \parallel C_m)$	17: **return** $(M_1 \parallel \ldots \parallel M_m)$

Definition 5 (IND-SPRP-Security). *Let $E \in \mathtt{Block}(k, n)$ denote a block cipher and E^{-1} its inverse. Let Perm_n be the set of all n-bit permutations. The IND-SPRP advantage of $\mathcal{A}$ against E is then defined by*

$$\mathbf{Adv}^{\mathsf{IND\text{-}SPRP}}_{E,E^{-1}}(\mathcal{A}) \leq \left|\Pr\left[\mathcal{A}^{E(\cdot),E^{-1}(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\pi(\cdot),\pi^{-1}(\cdot)} \Rightarrow 1\right]\right|,$$

where the probabilities are taken over $K \xleftarrow{\$} \{0,1\}^k$ and $\pi \xleftarrow{\$} \mathsf{Perm}_n$. We define $\mathbf{Adv}^{\mathsf{IND\text{-}SPRP}}_{E,E^{-1}}(q,t)$ as the maximum advantage over all IND-SPRP-adversaries $\mathcal{A}$ on E that run in time at most t and make at most q queries to the available oracles.

We borrow the OPRP-CCA notion from Bellare et al. [4,5]. The OPRP-CCA-security specifies the maximal advantage of an adversary $\mathcal{A}$ with access to an encryption and decryption oracle to distinguish the outputs of a on-line cipher Γ under a randomly chosen key K from that of a random permutation.

Definition 6 (OPRP-CCA-Security). *Let K a k-bit key, P a random on-line permutation, and $\Gamma : \{0,1\}^k \times (\{0,1\}^n)^* \to (\{0,1\}^n)^*$ be an on-line cipher. Then, we define the OPRP-CCA-advantage of an adversary $\mathcal{A}$ by*

$$\mathbf{Adv}^{\mathsf{OPRP\text{-}CCA}}_{\Gamma}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{\Gamma_K(\cdot),\Gamma_K^{-1}(\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{P(\cdot),P^{-1}(\cdot)} \Rightarrow 1\right]\right|, \quad (1)$$

where the probabilities are taken over $K \xleftarrow{\$} \mathcal{K}$ *and* $P \xleftarrow{\$} \mathsf{OPerm}_n$*. Further, we define* $\mathbf{Adv}^{\mathsf{OPRP\text{-}CCA}}_{\Gamma}(q,\ell,t)$ *as the maximum advantage over all adversaries* $\mathcal{A}$ *that run in time at most* t*, and make at most* q *queries of total length of at most* ℓ *blocks to the available oracles.*

Bellare and Namprempre showed in [6] that IND-CCA-security implies *non-malleable chosen-ciphertext*-security (NM-CCA). Hence, OPRP-CCA implies *weak* non-malleability, i.e., an adversary that manipulates the i-th ciphertext block cannot distinguish the $(i+1)$-th, $(i+2)$-th, ... ciphertext blocks of Γ from random.

3.3 OPRP-CCA-Security of POE

Theorem 2. *Let* $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ *be a block cipher and* E^{-1} *its inverse operation. Let* $\pi \xleftarrow{\$} \mathsf{Perm}_n$ *denote an* n*-bit random permutation that was chosen uniformly from random, and let* π^{-1} *denote its inverse. Then, it holds that*

$$\mathbf{Adv}^{\mathsf{OPRP\text{-}CCA}}_{\mathsf{POE}_{E,E^{-1}}}(q,\ell,t) \leq \ell^2\epsilon + \frac{\ell^2}{2^n-\ell} + \mathbf{Adv}^{\mathsf{IND\text{-}SPRP}}_{E,E^{-1}}(\ell, O(t)). \tag{2}$$

Proof. Let $\mathcal{A}$ be an OPRP-CCA-adversary with access to an oracle $\mathcal{O}$, which responds either with real encryption/decryptions using $\mathsf{POE}_{E_K,E_K^{-1}}$ or a random on-line permutation P, as given in Definition 6. We say that $\mathcal{A}$ collects its queries and the corresponding oracle response as tuples (M,C) in a query history $\mathcal{Q}$. Wlog., we assume that $\mathcal{A}$ will not make queries to which it already knows the answer.
It is easy to see that we can rewrite Equation (1) as (cf. [20], Sec. 4):

$$\mathbf{Adv}^{\mathsf{OPRP\text{-}CCA}}_{\mathsf{POE}_{E,E^{-1}}}(\mathcal{A}) \leq \left|\Pr\left[\mathcal{A}^{\mathsf{POE}_E,\mathsf{POE}^{-1}_{E^{-1}}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathsf{POE}_\pi,\mathsf{POE}^{-1}_{\pi^{-1}}} \Rightarrow 1\right]\right| \tag{3}$$

$$+ \left|\Pr\left[\mathcal{A}^{\mathsf{POE}_\pi,\mathsf{POE}^{-1}_{\pi^{-1}}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{P(\cdot),P^{-1}(\cdot)} \Rightarrow 1\right]\right|. \tag{4}$$

It is easy to see that Equation (3) can be upper bounded by

$$\mathbf{Adv}^{\mathsf{IND\text{-}SPRP}}_{E,E^{-1}}(\ell, \mathrm{O}(t)).$$

It remains to study the difference in (4), which refers to the advantage of $\mathcal{A}$ to distinguish POE instantiated with an n-bit random permutation π from P. We can identify two cases from the structure of POE: (1) collisions between internal values of POE occur (COLL), or (2) no collisions occur (NOCOLL). From the law of total probability follows that we can rewrite (4) as

$$\begin{aligned}&\left|\Pr\left[\mathcal{A}^{\mathsf{POE}_\pi,\mathsf{POE}^{-1}_{\pi^{-1}}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{P(\cdot),P^{-1}(\cdot)} \Rightarrow 1\right]\right| \\ \leq\ &\Pr\left[\mathrm{COLL}\right] \cdot \Pr\left[\mathrm{COLLWIN}\right] + \Pr\left[\neg\,\mathrm{COLL}\right] \cdot \Pr\left[\mathrm{NOCOLLWIN}\right],\end{aligned}$$

with

$$\Pr[\text{COLLWIN}] = \left|\Pr\left[\mathcal{A}^{\mathsf{POE}_\pi, \mathsf{POE}_{\pi^{-1}}^{-1}} \Rightarrow 1 \,|\, \text{COLL}\right] - \Pr\left[\mathcal{A}^{P(\cdot), P^{-1}(\cdot)} \Rightarrow 1\right]\right|,$$
$$\Pr[\text{NOCOLLWIN}] = \left|\Pr\left[\mathcal{A}^{\mathsf{POE}_\pi, \mathsf{POE}_{\pi^{-1}}^{-1}} \Rightarrow 1 \,|\, \neg\,\text{COLL}\right] - \Pr\left[\mathcal{A}^{P(\cdot), P^{-1}(\cdot)} \Rightarrow 1\right]\right|.$$

For the sake of simplicity, we upper bound $\Pr[\text{COLLWIN}]$ and $\Pr[\neg\,\text{COLL}]$ by 1. Thus, we only have to look at $\Pr[\text{COLL}]$ and $\Pr[\text{NOCOLLWIN}]$.

***Case 1:* COLL.** In this case, $\mathcal{A}$ tries to distinguish POE from random by exploiting some collision between internal values. Since π is a random permutation, any *fresh* (i.e., not previously queried) input to $\pi(\cdot)$ or $\pi^{-1}(\cdot)$ produces a random output and therefore:

1. For any fresh X_i, the result of $\pi(X_i) \oplus F_{K_2}(Y_{i-1})$ will be random.
2. For any fresh Y_i, the result of $\pi^{-1}(Y_i) \oplus F_{K_1}(X_{i-1})$ will be random.

We obtain two possible subcases: a collision between internal values in the top row occurred (COLL_{top}), or a collision between in internal values in the bottom row occurred (COLL_{bot}). COLL then represents the event that either (or both) subcases occurred.

$$\text{COLL} \;=\; \text{COLL}_{\text{top}} \vee \text{COLL}_{\text{bot}}.$$

***Subcase 1.1:* $\mathbf{COLL_{top}}$.** By an internal collision in the top row, we refer to the event that $X_i = X'_j$ for two distinct tuples (X_{i-1}, M_i) and (X'_{j-1}, M'_j), with $i, j \geq 1$:

$$X_i = F_{K_1}(X_{i-1}) \oplus M_i, \quad \text{and} \quad X'_j = F_{K_1}(X'_{j-1}) \oplus M'_j.$$

Since F is an ϵ-AXU family of hash functions, the family F' of hash functions

$$F'_{K_1}(X_{i-1}, M_i) \;:=\; F_{K_1}(X_{i-1}) \oplus M_i$$

is ϵ-AU (cf. Theorem 1). Thus, the probability of a top-row collision for at most ℓ queried message blocks can be upper bounded by

$$\Pr[\text{COLL}_{\text{top}}] \;=\; \frac{\ell(\ell-1)}{2} \cdot \epsilon \;\leq\; \frac{\ell^2}{2}\epsilon.$$

***Subcase 1.2:* $\mathbf{COLL_{bot}}$.** We define a bottom-row collision as the event that two distinct tuples (Y_{i-1}, C_i) and (Y'_{j-1}, C'_j) produce the same values $Y_i = Y'_j$, with

$$Y_i = F_{K_2}(Y_{i-1}) \oplus E_K(X_i), \quad \text{and} \quad Y'_j = F_{K_2}(Y'_{j-1}) \oplus E_K(X'_j).$$

Due to the symmetric structure of POE, the analysis for bottom-row collisions is similar to that of top-row collisions. Thus, the probability for this event can also be upper bounded by

$$\Pr[\text{COLL}_{\text{bot}}] = \frac{\ell(\ell-1)}{2} \cdot \epsilon \;\leq\; \frac{\ell^2}{2}\epsilon.$$

Hence, we can upper bound $\Pr[\text{COLL}] \;\leq\; \Pr[\text{COLL}_{\text{top}}] + \Pr[\text{COLL}_{\text{top}}] \leq \ell^2\epsilon$.

***Case 2:* NOCOLLWIN.** Next, we regard the case that $\mathcal{A}$ shall distinguish $(\mathsf{POE}_\pi, \mathsf{POE}_{\pi^{-1}}^{-1})$ from $(P(\cdot), P^{-1}(\cdot))$ when no internal collisions occur. We can generalize that each pair of tuples $(M, C), (M', C') \in \mathcal{Q}$ shares a common prefix of 0 to $\min(|M|, |M'|)/n$ blocks. Wlog., say that the pair $M, M' \in \mathcal{Q}$ shares an i-block common prefix, i.e., $M_j = M'_j$, $\forall j \in [1, i]$, and $M_{i+1} \neq M'_{i+1}$. In the following, we study the difference in the behavior of POE and P for three subcases: (2.1) for the message blocks in the common prefix, $M_1, \ldots, M_i$, (2.2) for the $(i+1)$-th block, or (2.3) for the message blocks after the $(i+1)$-th one.

Subcase 2.1: Common Prefix. Since an OPERM is deterministic, input and output behaviors of $(\mathsf{POE}_\pi, \mathsf{POE}_{\pi^{-1}}^{-1})$ and $(P\cdot, P^{-1}(\cdot))$ are identical for the common prefix. Hence, the advantage for $\mathcal{A}$ in this subcase is 0.

Subcase 2.2: Directly After the Common Prefix. Since $M_j = M'_j$, $\forall j \in [1, i]$, it must hold in the real case that $Y_i = Y'_i$ and $X_i = X'_i$. From $M_{i+1} \neq M'_{i+1}$ follows

$$C_{i+1} = \pi(F_{K_1}(X_i) \oplus M_{i+1}) \oplus F_{K_2}(Y_i) \neq \pi(F_{K_1}(X'_i) \oplus M'_{i+1}) \oplus F_{K_2}(Y'_i) = C'_{i+1}.$$

Since π is a random permutation, C_{i+1}, C'_{i+1} are chosen uniformly at random in the real case. In the random case P is used with two different prefixes $M_1 \,||\, \ldots \,||\, M_{i+1}$ and $M'_1 \,||\, \ldots \,||\, M'_{i+1}$. Since P is an OPERM, $C_{i+1} \neq C'_{i+1}$ also must hold in this case. Hence, the advantage for $\mathcal{A}$ in this subcase is also 0.

Subcase 2.3: After the $(i+1)$-th Message Block. In the random case, each query output is chosen uniformly at random from $\{0,1\}^n$. However, in the real world each output of either an encryption or a decryption query is chosen uniformly at random from the set $\{0,1\}^n \setminus \mathcal{Q}$. This means that in the real case POE loses randomness with every query. We can upper bound the success probability of an adversar to distinguish POE from a random OPERM by

$$\frac{\ell^2}{2^n - \ell}.$$

Our claim in Equation (2) follows from summing up the individual terms. □

4 The On-Line AE Scheme POET

For McOE, Fleischmann et al. [20] showed that an OPRP-CCA-secure on-line cipher can be easily transformed into an on-line AEAD scheme that is resistant against nonce and decryption misuse. This section shows how to apply their approach to transform POE into a nonce- misuse-resistant AE scheme for messages whose lengths are a multiple of the block length.

4.1 Definition of POET

Definition 7 (POET). *Let $k, n \geq 1$ be integers. Let $\mathsf{POET} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE scheme, $E : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^*$ a block cipher, and $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ a family of keyed ϵ-AXU hash functions. Furthermore, let $F_i : \{0,1\}^{ni} \to \{0,1\}^n$ be $i\epsilon$-AXU family of hash functions defined as follows:*

$$F_0 = F(1); \quad F_i(M) = F(F_{i-1}(M_1, \ldots, M_{i-1}) \oplus M_i) \qquad i \in \mathbb{N}^+.$$

Let H be the header (including the nonce as its final block), M the message, T the authentication tag, and C the ciphertext, with $H, M, C \in (\{0,1\}^n)^*$ and $T \in \{0,1\}^n$. Then, we define encryption and decryption algorithms of POET as shown in Algorithm 2.

Algorithm 2 Procedures **Encrypt** and **Decrypt** for POET.

Encrypt(H, M)	**Decrypt**(H, C, T)
101: $X_0 \leftarrow Y_0 \leftarrow 1$, $m \leftarrow \frac{\lvert M\rvert}{n}$ $h \leftarrow \frac{\lvert H\rvert}{n}$	201: $X_0 \leftarrow Y_0 \leftarrow 1$, $m \leftarrow \frac{\lvert C\rvert}{n}$ $h \leftarrow \frac{\lvert H\rvert}{n}$
102: **for** $i \leftarrow 1, \ldots, h$ **do**	202: **for** $i \leftarrow 1, \ldots, h$ **do**
103: $\quad X_i \leftarrow F_{K_1}(X_{i-1}) \oplus H_i$	203: $\quad X_i \leftarrow F_{K_1}(X_{i-1}) \oplus H_i$
104: $\quad Y_i \leftarrow E_K(X_i)$	204: $\quad Y_i \leftarrow E_K(X_i)$
105: **end for**	205: **end for**
106: $\tau \leftarrow F_{K_2}(Y_{h-1}) \oplus Y_h$	206: $\tau \leftarrow F_{K_2}(Y_{h-1}) \oplus Y_h$
107: $M_m \leftarrow M_m \oplus E_K(\lvert M\rvert)$	207: **for** $i \leftarrow 1, \ldots, m$ **do**
108: **for** $i \leftarrow 1, \ldots, m$ **do**	208: $\quad j \leftarrow i + h$
109: $\quad j \leftarrow i + h$	209: $\quad Y_j \leftarrow F_{K_2}(Y_{j-1}) \oplus C_i$
110: $\quad X_j \leftarrow F_{K_1}(X_{j-1}) \oplus M_i$	210: $\quad X_j \leftarrow E_K^{-1}(Y_j)$
111: $\quad Y_j \leftarrow E_K(X_j)$	211: $\quad M_i \leftarrow F_{K_1}(X_{j-1}) \oplus X_j$
112: $\quad C_i \leftarrow F_{K_2}(Y_{j-1}) \oplus Y_j$	212: **end for**
113: **end for**	213: $M_m \leftarrow M_m \oplus E_K(\lvert C\rvert)$
114: $j \leftarrow m + h$	214: $j \leftarrow m + h$
115: $X_{j+1} \leftarrow F_{K_1}(X_j) \oplus \tau$	215: $X_{j+1} \leftarrow F_{K_1}(X_j) \oplus \tau$
116: $T \leftarrow F_{K_2}(Y_j) \oplus E_K(X_{j+1})$	216: $T' \leftarrow F_{K_2}(Y_j) \oplus E_K(X_{j+1})$
117: **return** $(C_1 \parallel \ldots \parallel C_m, T)$	217: **if** $T = T'$ **then**
	218: $\quad$ **return** $(M_1 \parallel \ldots \parallel M_m)$
	219: **end if**
	220: **return** $\perp$

A schematic illustration of the encryption algorithm is given in Figure 2.

Remark 2. POET uses the common 10*-padding for headers $|H|$ whose length is not a multiple of n. As a result, H consists of at least a single block, and the entire header can be seen as a nonce. For messages whose length is not a multiple of the block size, POET *borrows* the provably secure tag-splitting approach from McOE [20]. Therefore, it is sufficient to prove the OCCA3-security *only* for messages whose length is a multiple of the block size.

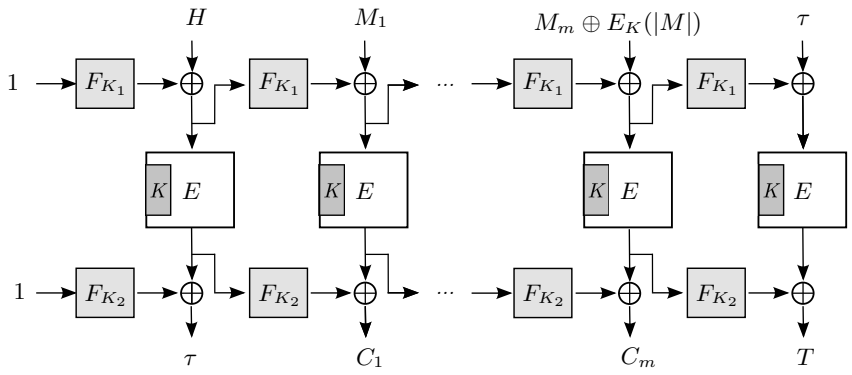

Fig. 2. The encryption process for an m-block message M of POET.

4.2 Security Notions for On-Line AE Schemes

We define an on-line authenticated encryption scheme Π to be OCCA3-secure iff it provides both OPRP-CPA *and* INT-CTXT security. Note that we explicitly regard nonce-ignoring adversaries which are allowed to use a nonce multiple times, similar to the security notions of integrity for authenticated encryption schemes in [20]. In the next part, we briefly revisit the formal definitions of INT-CTXT and OCCA3.

The INT-CTXT-advantage of an adversary $\mathcal{A}$ is given by the success probability of winning the game $G_{\mathsf{INT\text{-}CTXT}}$ that is defined in Figure 3. Thus, we obtain

$$\mathbf{Adv}_{\Pi}^{\mathsf{INT\text{-}CTXT}}(\mathcal{A}) \leq \Pr\left[\mathcal{A}^{G_{\mathsf{INT\text{-}CTXT}}} \Rightarrow 1\right], \tag{5}$$

where $\mathbf{Adv}_{\Pi}^{\mathsf{INT\text{-}CTXT}}(q, \ell, t)$ is the maximum advantage over all INT-CTXT adversaries $\mathcal{A}$ that run in time at most t, and make at most q queries with a total length of at most ℓ blocks to the available oracles.

```
1  Initialize()
2  K ← K(); Q ← ∅;
3  Finalize()
4  return win;
10 Encrypt(H, M)
11 (C, T) ← E_K(H, M);
12 Q ← Q ∪ {(H, C)};
13 return (C, T);
20 Verify(H, C, T)
21 M ← D_K(H, C, T);
22 if ((H, C) ∉ Q and M ≠ ⊥) then
23    win ← true;
24 return (M ≠ ⊥);
```

Fig. 3. The $G_{\mathsf{INT\text{-}CTXT}}$ game for an authenticated encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$.

Definition 8 (OCCA3-Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an on-line authenticated encryption scheme. Then, the OCCA3-advantage of an adversary $\mathcal{A}$ is upper bounded by*

$$\mathbf{Adv}_{\Pi}^{\mathsf{OCCA3}}(\mathcal{A}) \leq \mathbf{Adv}_{\Pi}^{\mathsf{OPRP\text{-}CPA}}(q, \ell, t) + \mathbf{Adv}_{\Pi}^{\mathsf{INT\text{-}CTXT}}(q, \ell, t). \tag{6}$$

The OCCA3-advantage of Π, $\mathbf{Adv}_{\Pi}^{\mathsf{OCCA3}}(q, \ell, t)$, is then defined by the maximum advantage of all adversaries $\mathcal{A}$ that run in time at most t, and make at most q queries of a total length of at most ℓ blocks to the available oracles.

Note that an OPRP-CPA-adversary $\mathcal{A}$ on some encryption scheme Γ can always be used by an OPRP-CCA-adversary $\mathcal{A}$' on Γ that inherits the advantage of $\mathcal{A}$. In reverse direction, an upper bound for the OPRP-CCA-advantage of Γ is always an upper bound for the OPRP-CPA-advantage of Γ.

4.3 OCCA3-Security of POET

Theorem 3. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a POET scheme as defined in Definition 7. Then, it applies that*

$$\mathbf{Adv}_{\Pi}^{OCCA3}(q, \ell, t) \leq 2\,(\ell + 2q)^2\,\epsilon + \frac{(\ell + 2q)^2 + q}{2^n - (\ell + 2q)} + 2\mathbf{Adv}_{E,E^{-1}}^{IND\text{-}SPRP}(\ell + 2q, O(t)).$$

Proof. The proof follows from Theorem 2 and Lemma 1 (see Appendix A). Since Theorem 2 yields an upper bound for the OPRP-CCA-advantage on POE, it also provides an upper bound for the OPRP-CPA-advantage on POET. Though, the number of encrypted message (and header) blocks ℓ from Theorem 2 must be replaced by $(\ell + 2q)$ since the tag-generation process of POET includes two additional block-cipher calls per query. □

5 Key Derivation and Instantiations

5.1 Key Derivation

POE and POET require three internal keys: K for the block cipher, and two keys K_1, and K_2 for the two instances of F. Since our goal was to not put further restrictions on the used hash function families, we borrowed the idea from [26] to obtain pair-wise independent keys for top and bottom row. At setup, the user supplies a k-bit secret key L. The further keys are then derived from L by encrypting three distinct constants $const_0$, $const_1$, $const_2$ with E:

$$K \leftarrow E_L(const_0), \qquad K_1 \leftarrow E_L(const_1), \qquad K_2 \leftarrow E_L(const_2).$$

For simplicity, we recommend $const_0 = 1, const_1 = 2, const_2 = 3$. Therefore, under the assumption that E is a PRP-secure block cipher, we can ensure to obtain independent keys for cipher and hash function calls.

5.2 ϵ-AXU Hash Functions

We recommend to instantiate POE/POET with AES-128 as block cipher. For the ϵ-AXU families of hash functions F, we propose three suitable instantiations in the following.

POE/POET with Four-Round AES. When trying to minimize the implementation footprint, it may be desirable to have an encryption scheme based on only a single primitive. Furthermore, maximizing the throughput is often critical. Therefore, POE/POET with the first four rounds of the AES as a family of keyed hash functions may be an excellent choice for restricted devices and/or devices with support for AES native instructions. The drawback of this solution would be a slightly lower number of message blocks that can be processed under the same key. As shown by Daemen et al. in [15], four-round AES is a family of ϵ-AXU hash functions – under the reasonable assumption that all used round keys are independent – with

$$\epsilon \leq 1.88 \cdot 2^{-114} \approx 2^{-113}.$$

This implies that at most $\ll 2^{56}$ message blocks can be encrypted or decrypted under the same key.

POE/POET with Full-Round AES. As a more conservative variant we propose the full AES-128 for the family of hash functions. Under the common PRF assumption – where we asssume that AES is indistinguishable from a random 128-bit permutation, this constructions yields $\epsilon \approx 2^{-128}$.

POE/POET with Galois-Field Multiplications. In addition, we propose to use a multiplication in $GF(2^{128})$, similar to that in AES-GCM [32], as universal hash function. This approach yields an $\epsilon \approx 2^{-128}$.
We stress that POE and POET can be fully parallelized with Galois-Field multiplications. For instance, consider a message of at least four blocks, $M_1 \,||\, \ldots \,||\, M_4$. Using Galois-Field multiplications, the input for the second block-cipher call is $K^2 + KM_1 + M_2$. Instead of sequentially multiplying with K, adding M_3, multiplying with K and adding M_4, one can compute in parallel:

- For the third block-cipher call: $K \cdot (K^2 + KM_1 + M_2) + M_3$.
- For the fourth block-cipher call: $K^2 \cdot (K^2 + KM_1 + M_2) + KM_3 + M_4$.

This approach increases the total number of multiplications, but decreases the latency. Given c cores, and c subsequent message blocks to process, this approach reduces the latency from c hash-function calls to $O(\log c)$. This approach is used, e.g., in carry-lookahead adders, GCM [32], or CWC [28].

When using multiplications in $GF(2^{128})$, one has to consider the risk of weak keys. At FSE'12 Saarinen [43] pointed out that, since $2^{128} - 1$ is not prime and produces 2^9 smooth-order multiplicative groups, one can obtain a weak key with probability 2^{-96}. This observation was generalized Procter and Cid at FSE'13 [40], who showed that one can choose a polynomial $q(x)$ with a preferably high degree and no repeated roots. Then, one can create two messages M, M' that collide with $p = \frac{\#\text{roots of } q(x)}{2^{128}}$. After the FSE'14, Abdelraheem et al. [1] applied the observations of Procter and Cid to the version of POET that was submitted to the CAESAR competition, and concluded that the risk of obtaining a weak hash-function key for GF multiplications is about 2^{-66}.

To reduce the risk of weak hash-function keys, one can perform a check on K_1 and K_2 right after the key-generation phase, and – in case one of them or both are weak – to repeat the generation for the affected key(s) K_i, with the respective constant $const_i$ incremented by two. So, one can ensure that K, K_1, and K_2 are pair-wise independent. This procedure can be repeated until neither key is weak. Since this additional security measure must be applied only at the time of key setup, and since only a small fraction of keys are weak, its effort is negligible in the long-term view.

6 Conclusion

This paper presented POE, the first family of on-line ciphers which is both non-sequential *and* provably OPRP-CCA-secure. Its design combines two layers of ϵ-AXU hashing and a wrapped layer of ECB encryption.
Most on-line AE schemes have a significant latency since they must buffer a would-be plaintext until the tag has been been verified. The latency can be significantly decreased when the would-be plaintext is passed beforehand – however, this approach raises security issues when applied to AE schemes that lack OPRP-CCA-security, i.e., an adversary could obtain partial control about the would-be plaintext, even when these include additional checksums. On the other hand, previous OPRP-CCA-secure encryption schemes were inherently sequential. POE is well-suited for high-speed networks that require performant, low-latency encryption of large message frames, especially when classical authenticated decryption would increase latency significantly. Our application scenario targets optical transport networks (OTNs), but the latency imposed by authenticated decryption is an issue for other applications as well. In general, POE is an option for such applications.
We proposed three instantiations, where we recommended the AES as block cipher and either four-round AES, full AES, or a multiplication in $GF(2^{128})$ as ϵ-AXU families of hash functions. Additionally, we presented POET, a state-of-the-art on-line authenticated encryption scheme, which inherits the chosen-ciphertext-security and pipelineability from POE. Concluding, POET combines pipelineability with misuse-resistance in a novel way, at the cost of only a single block-cipher and two additional hash-function calls per message block.

7 Acknowledgments

We thank all reviewers of the FSE 2014 for their helpful comments and Daniel J. Bernstein and Tetsu Iwata for fruitful discussions. Finally, we thank Jian Guo, Jérémy Jean, Thomas Peyrin, and Lei Wang who pointed out a mismatch between the specified and the analyzed version of POET in the pre-proceedings version [21].

References

1. Mohamed Ahmed Abdelraheem, Andrey Bogdanov, and Elmar Tischhauser. Weak-Key Analysis of POET. Cryptology ePrint Archive, Report 2014/226, 2014. `http://eprint.iacr.org/`.
2. Elena Andreeva, Andrey Bogdanov, Atul Luykx, B. Mennink, Elmar Tischhauser, and Kan Yasuda. Parallelizable and Authenticated Online Ciphers. In *Directions in Authenticated Ciphers*, 2013.
3. Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. APE(X): Authenticated Permutation-Based Encryption with Extended Security Features. In *Directions in Authenticated Ciphers*, 2013.
4. Mihir Bellare, Alexandra Boldyreva, Lars R. Knudsen, and Chanathip Namprempre. Online Ciphers and the Hash-CBC Construction. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 292–309. Springer, 2001.
5. Mihir Bellare, Alexandra Boldyreva, Lars R. Knudsen, and Chanathip Namprempre. On-line Ciphers and the Hash-CBC Constructions. *Journal of Cryptology*, 25(4):640–679, 2012.
6. Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.
7. Mihir Bellare, Phillip Rogaway, and David Wagner. The EAX Mode of Operation. In Bimal K. Roy and Willi Meier, editors, *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer, 2004.
8. Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.
9. John Black, Phillip Rogaway, and Thomas Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In Yung [46], pages 320–335.
10. Martin Boesgaard, Thomas Christensen, and Erik Zenner. Badger - A Fast and Provably Secure MAC. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *ACNS*, volume 3531 of *Lecture Notes in Computer Science*, pages 176–191, 2005.
11. Alexandra Boldyreva and Nut Taesombut. Online Encryption Schemes: New Security Notions and Constructions. In Okamoto [36], pages 1–14.
12. Carl Campbell. Design and Specification of Cryptographic Capabilities. In *Proceedings of the Conference on Computer Security and the Data Encryption Standard held at the National Bureau of Standards in Gaithersburg*, NBS special publication, Gaithersburg, Md., February 1978. U.S. National Bureau of Standards.
13. Anne Canteaut, editor. *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*. Springer, 2012.
14. Larry Carter and Mark N. Wegman. Universal Classes of Hash Functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.
15. Joan Daemen, Mario Lamberger, Norbert Pramstaller, Vincent Rijmen, and Frederik Vercauteren. Computational Aspects of the Expected Differential Probability of 4-Round AES and AES-Like Ciphers. *Computing*, 85(1-2):85–104, 2009.

16. Nilanjan Datta and Mridul Nandi. Misuse Resistant Parallel Authenticated Encryptions. Cryptology ePrint Archive, Report 2013/767, 2013. http://eprint.iacr.org/.
17. Nilanjan Datta and Mridul Nandi. Characterization of EME with Linear Mixing. Cryptology ePrint Archive, Report 2014/009, 2014. http://eprint.iacr.org/.
18. W. Diffie and M. E. Hellman. Privacy and Authentication: An Introduction to Cryptography. In *Proceedings of the IEEE*, volume 67, pages 397–427. IEEE, March 1979. Invited Paper.
19. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
20. Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In Canteaut [13], pages 196–215.
21. Jian Guo, Jérémy Jean, Thomas Peyrin, and Wang Lei. Breaking POET Authentication with a Single Query. Cryptology ePrint Archive, Report 2014/197, 2014. http://eprint.iacr.org/.
22. Shai Halevi and Phillip Rogaway. A Tweakable Enciphering Mode. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 482–499. Springer, 2003.
23. Shai Halevi and Phillip Rogaway. A Parallelizable Enciphering Mode. In Okamoto [36], pages 292–304.
24. IEEE. IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices. *IEEE Std. 1619-2007*, pages c1 – 32, 2008.
25. ITU-T. Interfaces for the Optical Transport Network (OTN). Recommendation G.709/Y.1331, International Telecommunication Union, Geneva, December 2009.
26. Tetsu Iwata and Kaoru Kurosawa. OMAC: One-Key CBC MAC. In Thomas Johansson, editor, *FSE*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153. Springer, 2003.
27. Lars R Knudsen. Block Chaining Modes Of Operation. *Symmetric-Key Block-Cipher Modes of Operation Workshop*, October 2000.
28. Tadayoshi Kohno, John Viega, and Doug Whiting. CWC: A High-Performance Conventional Authenticated Encryption Mode. In *FSE*, pages 408–426, 2004.
29. Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In Antoine Joux, editor, *FSE*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327. Springer, 2011.
30. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. In Yung [46], pages 31–46.
31. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. *Journal of Cryptology*, 24(3):588–613, 2011.
32. David McGrew and John Viega. The Galois/Counter Mode of Operation (GCM). *Submission to NIST. http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/-gcm/gcm-spec.pdf*, 2004.
33. Mridul Nandi. A Simple Security Analysis of Hash-CBC and a New Efficient One-Key Online Cipher. Cryptology ePrint Archive, Report 2007/158, 2007. http://eprint.iacr.org/.
34. Mridul Nandi. Two New Efficient CCA-Secure Online Ciphers: MHCBC and MCBC. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT*, volume 5365 of *Lecture Notes in Computer Science*, pages 350–362. Springer, 2008.

35. US Department of Commerce. DES Modes of Operation. Technical Report FIPS PUB 81, US Department of Commerce / National Bureau of Standards, December 1998.
36. Tatsuaki Okamoto, editor. *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*. Springer, 2004.
37. J. Postel. User Datagram Protocol. RFC 768 (INTERNET STANDARD), August 1980.
38. J. Postel. Internet Protocol. RFC 791 (INTERNET STANDARD), September 1981. Updated by RFCs 1349, 2474, 6864.
39. J. Postel. Transmission Control Protocol. RFC 793 (INTERNET STANDARD), September 1981. Updated by RFCs 1122, 3168, 6093, 6528.
40. Gordon Procter and Carlos Cid. *On Weak Keys and Forgery Attacks against Polynomial-based MAC Schemes*. Lecture Notes in Computer Science - LNCS. Springer, 2013.
41. Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.
42. Phillip Rogaway and Haibin Zhang. Online Ciphers from Tweakable Blockciphers. In Aggelos Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 237–249. Springer, 2011.
43. Markku-Juhani Olavi Saarinen. Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. In Canteaut [13], pages 216–225.
44. Mark N. Wegman and J. Lawrence Carter. New Hash Functions and Their Use in Authentication and Set Equality. *Journal of Computer and System Sciences*, 22(3):265–279, June 1981.
45. Eric A. Young and Tim J. Hudson. OpenSSL: The Open Source toolkit for SSL/TLS. `http://www.openssl.org/`, September 2011.
46. Moti Yung, editor. *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*. Springer, 2002.

A Integrity-Proof of POET

In this section, we prove an upper bound for the INT-CTXT-security of POET. Prior, we recall the notion for the *longest common prefix* of two messages [20].

Definition 9 (Length of Longest Common Prefix). *For integers $n, \ell, d \geq 1$, let $\mathcal{D}_n^d = (\{0,1\}^n)^d$ denote the set of all strings that consist of exactly d blocks of n bits each. Further, let $\mathcal{D}_n^* = \bigcup_{d \geq 0} \mathcal{D}_n^d$ denote the set which consists of all possible n-bit messages and $\mathcal{D}_{\ell,n} = \bigcup_{0 \leq d \leq \ell} \mathcal{D}_n^d$ the set of all possible messages which consist of 0 to ℓ n-bit blocks. For arbitrary $P \in \mathcal{D}_n^d$, let P_i denote the i-th block for all $i \in 1, \ldots, d$. For $P, R \in \mathcal{D}_n^*$, we define the* length of the longest common prefix of n-bit blocks *of P and R by*

$$\mathsf{LLCP}_n(P, R) = \max_i \{\forall j \in 1, \ldots, i : \ P_j = R_j\}\,.$$

For a non-empty set $\mathcal{Q}$ of strings in D_n^* we define $\mathsf{LLCP}_n(\mathcal{Q}, P)$ by

$$\max_{q \in \mathcal{Q}} \{\mathsf{LLCP}_n(q, P)\}\,.$$

For example, if $P \in \mathcal{Q}$, then $\mathsf{LLCP}_n(\mathcal{Q}, P) = |P|/n$.

Lemma 1. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a POET scheme as defined in Definition 7. Then it applies:*

$$\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\Pi}(q, \ell, t) \;\leq\; (\ell + 2q)^2 \cdot \epsilon + \frac{q}{2^n - (\ell + 2q)} + \mathbf{Adv}^{\mathsf{IND\text{-}SPRP}}_{E, E^{-1}}(\ell + 2q, O(t)).$$

Proof. The bound is computed by game-playing arguments from Theorem 2 in [20]. From Equation 5 we have

$$\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\Pi}(\mathcal{A}) \;\leq\; \Pr[\mathcal{A}^{G_{\mathsf{INT\text{-}CTXT}}} \Rightarrow 1].$$

We assume that $\mathcal{A}$ does not ask redundant queries, i.e., queries to which it already knows the answer. In the following, we transform $G_{\mathsf{INT\text{-}CTXT}}$ step by step until we obtain a game which represents the encryption and verification with POET, and for which we can simply upper bound the winning probability. In total, we consider four games G_1 to G_4 which are listed in Figures 4 and 5.

```
 1 Initialize()   Game G_1
 2 L ← K
 3 L ← E_L(1), K_1 ← E_L(2), K_2 ← E_L(3)
 4 Q ← ∅, win ← false

100 Encrypt(H, M)   Game G_1
101 X_0 ← Y_0 ← 1^n, m ← |M|/n, h ← |H|/n
102 for i ← 1,...,h do
103     X_i ← F_{K_1}(X_{i-1}) ⊕ H_i
104     Y_i ← E_K(X_i)
105 end for
106 τ ← F_{K_2}(Y_{h-1}) ⊕ Y_h
107 M_m ← M_m ⊕ E_K(|M|)
108 for i ← 1,...,m do
109     j ← i + h
110     X_j ← F_{K_1}(X_{i-1}) ⊕ M_i
111     Y_j ← E_K(X_i)
112     C_i ← F_{K_2}(Y_{j-1}) ⊕ Y_j
113 end for
114 j ← m + h
115 X_{j+1} ← F_{K_1}(X_j) ⊕ τ
116 T ← F_{K_2}(Y_j) ⊕ E_K(X_{j+1})
117 return (C_1 || ... || C_m, T)

 5 Finalize()   Game G_1
 6 return win

118 Verify(H, C, T)   Game G_1
119 X_0 ← Y_0 ← 1^n, m ← |C|/n, h ← |H|/n
120 for i ← 1,...,h do
121     X_i ← F_{K_1}(X_{i-1}) ⊕ H_i
122     Y_i ← E_K(X_i)
123 end for
124 τ ← F_{K_2}(Y_{h-1}) ⊕ Y_h
125 for i ← 1,...,m do
126     j ← i + h
127     Y_j ← F_{K_2}(Y_{j-1}) ⊕ C_i
128     X_j ← E_K^{-1}(Y_j)
129     M_i ← F_{K_1}(X_{j-1}) ⊕ X_j
130 end for
131 M_m ← M_m ⊕ E_K(|C|)
132 j ← m + h
133 X_{j+1} ← F_{K_1}(X_j) ⊕ τ
134 T' ← F_{K_2}(Y_j) ⊕ E_K(X_{j+1})
135 if T = T' and {(H, C, M)} ∉ Q then
136     win ← true
137 end if
138 Q ← Q ∪ {(H, C, M)}
139 return T = T'
```

Fig. 4. Game G_1 for the proof of Lemma 1.

Game G_1 is almost identical to $G_{\mathsf{INT\text{-}CTXT}}$, and replaces only the generic `Encrypt` and `Verify` procedures with the concrete procedures from the definition of

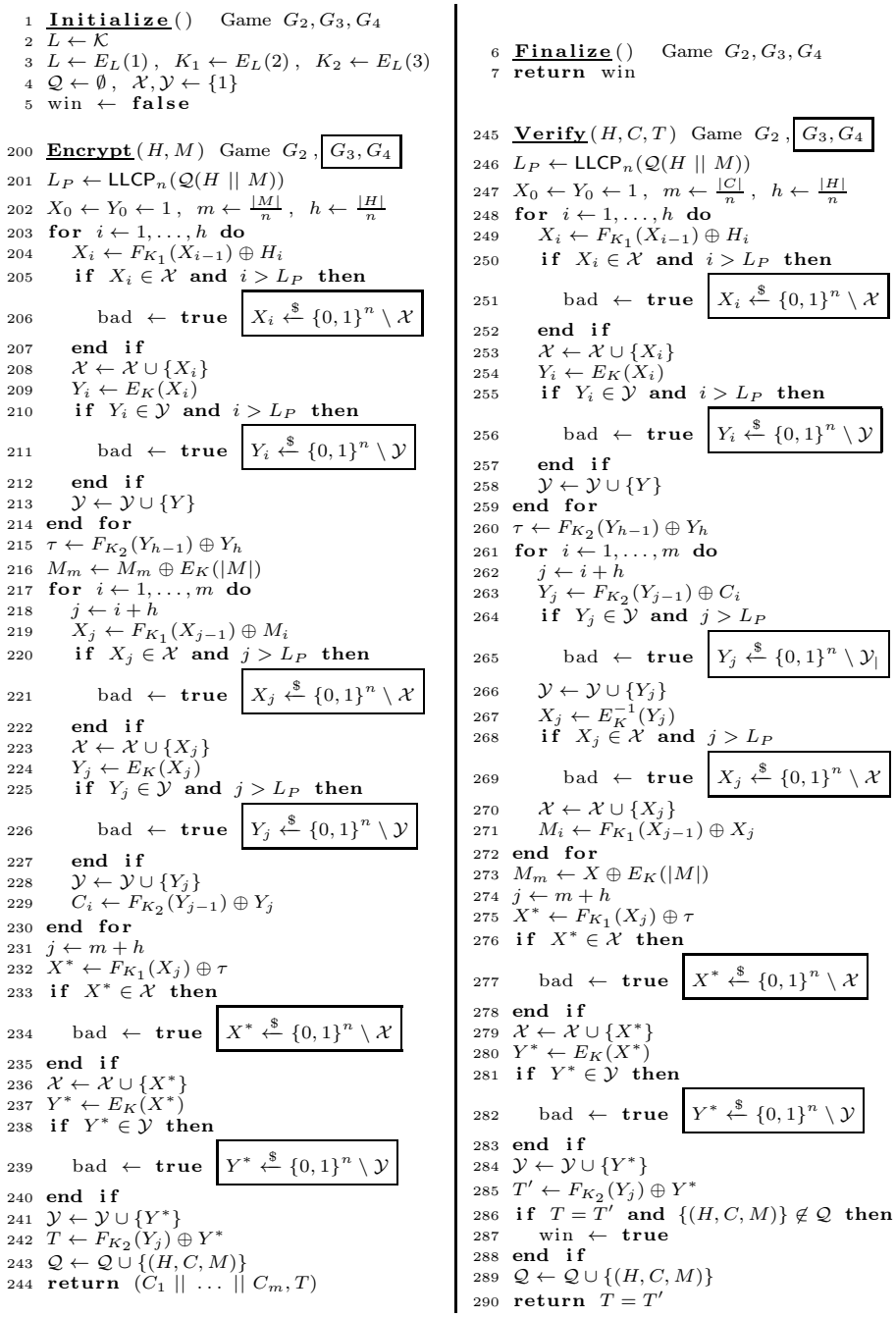

Fig. 5. Games G_2-G_4 for the proof of Lemma 1. G_3 and G_4 include the code in the boxes whereas G_2 does not.

POET. Hence, it applies that

$$\Pr[\mathcal{A}^{G_1} \Rightarrow 1] = \Pr[\mathcal{A}^{G^{\textsf{INT-CTXT}}} \Rightarrow 1].$$

G_2 bases on G_1, but adds the following steps:

- G_2 collects all values X and Y that occur during encryption and verification in two sets $\mathcal{X}$ and $\mathcal{Y}$, respectively.
- G_2 looks up the length of the longest common prefix L_P of the current input $(H \mid\mid M)$ (or $(H \mid\mid C)$ for decryption, respectively) and all previous messages (which are stored in $\mathcal{Q}$).
- Each time a non-trivial collision (i.e., a collision that is not due to a common prefix of two queried messages) between a value of X and a value in $\mathcal{X}$ occurs, G_2 sets `bad`. Similarly, each time a non-trivial collision between the current value Y and a value in $\mathcal{Y}$ occurs, it also sets `bad`.

Though, these changes do not affect the values obtained by the adversary, and thus

$$\Pr[\mathcal{A}^{G_2} \Rightarrow 1] = \Pr[\mathcal{A}^{G_1} \Rightarrow 1].$$

In the following, we investigate the success probability of $\mathcal{A}$ to win the games G_3 and G_4. G_3 is similar to G_2, except, when `bad` is set for a collision in $\mathcal{X}$, the current value of X is replaced by a new random value of X. Similarly, when a collision occured in $\mathcal{Y}$, the current value Y is replaced by a new random value. G_4 is then almost identical to G_3, and only replaces E with a random permutation, where K denotes the index. So, one can see from the listing of these games that

$$\begin{aligned}
\Pr[\mathcal{A}^{G_2} \Rightarrow 1] \;&=\; \Pr[\mathcal{A}^{G_3} \Rightarrow 1] + \left|\Pr[\mathcal{A}^{G_2} \Rightarrow 1] - \Pr[\mathcal{A}^{G_3} \Rightarrow 1]\right| \\
&\leq\; \Pr[\mathcal{A}^{G_3} \Rightarrow 1] + \Pr[\mathcal{A}^{G_3} \mathit{sets}\ \texttt{bad}] \\
&\leq\; \Pr[\mathcal{A}^{G_4} \Rightarrow 1] + \left|\Pr[\mathcal{A}^{G_3} \Rightarrow 1] - \Pr[\mathcal{A}^{G_4} \Rightarrow 1]\right| \\
&\quad + \Pr[\mathcal{A}^{G_3} \mathit{sets}\ \texttt{bad}]. \qquad (7)
\end{aligned}$$

Hence, it suffices to upper bound the success probability of the individual terms in the bottom row of Equation 7. We upper bound the terms from right to left. The difference of an adversary to win Game G_3 but not Game G_2 results from the probability that G_3 *sets* `bad`. This implies, that $\mathcal{A}$ must have found a collision between either the current X and some $X' \in \mathcal{X}$, or between the current Y and some $Y' \in \mathcal{Y}$. For both cases, Theorem 1 states that the probability to find a collision for F_{K_i} is at most ϵ. Since there are in total ℓ blocks from messages and headers plus $2q$ blocks for the tag-generation, it follows from [9] that

$$\Pr[\mathcal{A}^{G_3} \mathit{sets}\ \texttt{bad}] \;\leq\; 2\epsilon \cdot \frac{(\ell + 2q)^2}{2} \;=\; (\ell + 2q)^2 \cdot \epsilon.$$

We proceed with the term that describes the difference in the advantage of winning G_3 and G_4. It is easy to see that this term can be upper bounded by the advantage of adversaries to distinguish E, E^{-1} from a random PRP:

$$\left|\Pr[\mathcal{A}^{G_3} \Rightarrow 1] - \Pr[\mathcal{A}^{G_4} \Rightarrow 1]\right| \;\leq\; \mathbf{Adv}^{\textsf{IND-SPRP}}_{E,E^{-1}}(\ell + 2q, O(t)).$$

Finally, we have to upper bound the advantage of $\mathcal{A}$ to win the Game G_4. Prior, we have to introduce a few notions. We use the notion $X^* = F_{K_1}(X_{h+m}) \oplus \tau$ as listed in Line 232 of the `Encrypt` and in Line 275 of the `Verify` procedure, respectively. Further, we denote $Y^* = E_K(X^*)$ in Line 237 of the `Encrypt` and in Line 280 of the `Verify` procedure in Figure 4. Further, we will call a value *old* if it already is stored in a certain set, and *fresh* otherwise.
$\mathcal{A}$ can win G_4 only if the condition in Line 286 is fulfilled. We can see that X^* must be fresh; otherwise, a bad event would have happened before. Since E_K is a random permutation, it applies that the output Y^* is a random value out of a set of $2^n - (\ell + 2q)$ elements, and hence, T' is also a random value out of a set of the same size. Therefore, the probability to win G_4 can be upper bounded by

$$\frac{q}{2^n - (\ell + 2q)}.$$

The bound from Lemma 1 results from summing up the individual terms. □

B Observations on COPE

COPE is a parallelizable on-line cipher designed by Andreeva et al. [2] and is the underlying construction of the AE scheme COPA. A formal description is given as follows. Let $E \in \texttt{Block}$ and a fixed key $K \in \{0,1\}^k$. Then, COPE and its inverse are defined as shown in Algorithm 3.

Algorithm 3 Definition of COPE following [2].

Encrypt(M)	**Decrypt**(C)
1: $L \leftarrow E_K(0), \Delta_0 \leftarrow 3L, \Delta_1 \leftarrow 2L$	11: $L \leftarrow E_K(0), \Delta_0 \leftarrow 3L, \Delta_1 \leftarrow 2L$
2: $Y_0 \leftarrow L$	12: $X_0 \leftarrow L$
3: **for** $i = 1, \ldots, \ell$ **do**	13: **for** $i = 1, \ldots, \ell$ **do**
4: $\quad X_i \leftarrow E_K(M_i \oplus \Delta_0)$	14: $\quad Y_i \leftarrow E_K^{-1}(C_i \oplus \Delta_1)$
5: $\quad Y_i \leftarrow X_i \oplus Y_{i-1}$	15: $\quad X_i \leftarrow Y_i \oplus X_{i-1}$
6: $\quad C_i \leftarrow E_K(Y_i) \oplus \Delta_1$	16: $\quad M_i \leftarrow E_K^{-1}(X_i) \oplus \Delta_0$
7: $\quad \Delta_0 \leftarrow 2\Delta_0,\ \Delta_1 \leftarrow 2\Delta_1$	17: $\quad \Delta_0 \leftarrow 2\Delta_0,\ \Delta_1 \leftarrow 2\Delta_1$
8: **end for**	18: **end for**
9: **return** $(C_1 \parallel \ldots \parallel C_\ell)$	19: **return** $(M_1 \parallel \ldots \parallel M_\ell)$

In the following we show that COPE is not OPRP-CCA-secure.

OPRP-CCA-Attack. Let $\mathcal{A}$ be an OPRP-CCA adversary that communicates with two oracles $\mathcal{E}_K$ and $\mathcal{D}_K$. Let $M_a \neq M_b$ two distinct message blocks. Then, we denote $Y_a = E_K(M_a \oplus \Delta_0) \oplus L$ and $Y_b = E_K(M_b \oplus \Delta_0) \oplus L$.

1. First, $\mathcal{A}$ sends the encryption query (M_a, M_c) to $\mathcal{E}$, which responds with $(C_a, C_{(a,c)})$. In the "real" setting, it holds that

$$X_c = E_K(M_c \oplus 2\Delta_0), \qquad Y_{(a,c)} = Y_a \oplus X_c, \qquad C_{(a,c)} = E_K(Y_{(a,c)} \oplus 2\Delta_1).$$

2. Next, $\mathcal{A}$ requests the encryption of (M_b, M_c) and obtains $(C_b, C_{(b,c)})$. It holds that

$$X_c = E_K(M_c \oplus 2\Delta_0), \qquad Y_{(b,c)} = Y_b \oplus X_c, \qquad C_{(b,c)} = E_K(Y_{(b,c)} \oplus 2\Delta_1).$$

3. Then, $\mathcal{A}$ requests the decryption of the tuple $(C_a, C_{(b,c)})$, and $\mathcal{D}$ responds with $(M_a, M_{(a,bc)})$.

$$Y_{(b,c)} = E_K^{-1}(C_{(b,c)} \oplus 2\Delta_1), \quad X_{(a,bc)} = Y_{(b,c)} \oplus Y_a = Y_b \oplus X_c \oplus Y_a.$$

4. Finally, $\mathcal{A}$ sends the decryption query $(C_b, C_{(a,c)})$ and obtains $(M_b, M_{(b,ac)})$. It applies that

$$Y_{(a,c)} = E_K^{-1}(C_{(a,c)} \oplus 2\Delta_1), \quad X_{(b,ac)} = Y_{(a,c)} \oplus Y_b = Y_a \oplus X_c \oplus Y_b = X_{(a,bc)}.$$

From $X_{(a,bc)} = X_{(b,ac)}$ follows that $M_{(a,bc)} = M_{(b,ac)}$ in the real case. Hence, $\mathcal{A}$ returns `true` if $M_{(a,bc)} = M_{(b,ac)}$ and `false` otherwise, and can distinguish COPE from a random OPERM with probability $1 - 2^{-n}$.

Discussion. Our observation is also applicable to the on-line cipher of ELmE – or more generally to any on-line EME cipher with linear mixing layer. We want to stress that the shown attack does not invalidate any of the stated security claims of COPE or ELmE. However, our observation points out the importance of keeping the would-be plaintexts secret – otherwise, linear EME schemes can neither protect the data privacy of messages anymore. Therefore, one can not consider such designs secure in the decryption-misuse setting, which makes them a suboptimal choice for high-speed networks with low-latency requirements. Nevertheless, it remains an open research question if this property is undesired for further practical use cases.