ELmE: A Misuse Resistant Parallel Authenticated Encryption

Nilanjan Datta and Mridul Nandi

Indian Statistical Institute, Kolkata, India
nilanjan_isi_jrf@yahoo.com, mridul.nandi@gmail.com

Abstract. The authenticated encryptions which resist misuse of initial value (or nonce) at some desired level of privacy are two-pass or Mac-then-Encrypt constructions (inherently inefficient but provide full privacy) and online constructions, e.g., McOE, sponge-type authenticated encryptions (such as duplex) and COPA. Only the last one is almost parallelizable with some bottleneck in processing associated data. In this paper, *we design a new online secure authenticated encryption, called* ELmE *or Encrypt-Linear mix-Encrypt, which is completely (two-stage)* **parallel** *(even in associated data) and* **pipeline implementable**. It also provides full privacy when associated data (which includes initial value) is not repeated. The basic idea of our construction is based on EME, an Encrypt-Mix-Encrypt type SPRP constructions (secure against chosen plaintext and ciphertext). But unlike EME, we have used an online computable efficient **linear mixing** instead of a non-linear mixing. Our construction optionally supports **intermediate tags** which can be verified faster with less buffer size. Intermediate tag provides security against block-wise adversaries which is meaningful in low-end device implementation.

Keywords: Authenticated Encryption, Privacy, Misuse Resistant, EME.

1 Introduction

The common application of cryptography is to implement a secure channel between two or more users and then exchanging information over that channel. These users can initially set up their one-time shared key. Otherwise, a typical implementation first calls a key-exchange protocol for establishing a shared key or a session key (used only for the current session). Once the users have a shared key, either through the initial key set-up or key-exchange, they use this key to authenticate and encrypt the transmitted information using efficient symmetric-key algorithms such as a *message authentication code* $\mathsf{Mac}(\cdot)$ and (symmetric-key) *encryption* $\mathsf{Enc}(\cdot)$. The encryption provides **privacy** or **confidentiality** (hiding the sensitive data M, we call it *plaintext* or *message*) resulting a ciphertext C, whereas a message authentication code provides **data-integrity** (authenticating the transmitted message M or the ciphertext C) resulting a tag T. An authenticated encryption or AE is an integrated scheme which provides both privacy

W. Susilo and Y. Mu (Eds.): ACISP 2014, LNCS 8544, pp. 306–321, 2014.

of plaintext and authenticity or data integrity of message or ciphertext. An authenticated encryption scheme F_K takes associated data D (which may include initial value or nonce) and message M and produces tagged-ciphertext (C, T). Its inverse F_K^{-1} returns $\perp$ for all those (D, C, T) for which no such M exists, otherwise it returns M. Note that the associated data D must be sent along with tagged-ciphertext to decrypt correctly.

1.1 Examples of Authenticated Encryptions

So far, cryptography community put a lot of effort of designing different authenticated encryptions. CAESAR [1], a competition for Authenticated Encryption is going on, which will identify a portfolio of authenticated ciphers that offer advantages over AES-GCM and are suitable for widespread adoption. We have submitted ELmD v1.0 [1], a variant of ELmE (main difference is in the masking) in the competition and believe that it would be a strong candidate for this competition. We quickly mention some of the popularly known competitive constructions putting into different categories based on construction types.

Encrypt-and-MAC and Encrypt-then-MAC. It relies on non-repeating IV (or nonce), e.g. CCM [16], EAX [4], GCM [35], CHM [17], Sarkar's generic construction [34] and dedicated Stream Ciphers like Grain [15], Zuc [2] etc. All these constructions combine counter type encryption and a Mac.

MAC-then-Encrypt. It is a two-pass IV misuse resistant category e.g., SIV [33], BTM [19], HBS [18]. These compute a tag first and then based on this tag, counter type encryption is used to encrypt.

Online Feed Back Encryption. It uses feedback type encryption, e.g. IACBC [21], XCBC [8], CCFB [24], McOE [11], sponge-type constructions (Duplex [6]). These constructions have a bottleneck that they are not fully parallelizable. Our construction ELmE and COPA [3] also fall in this category which use basic structure of completely parallel EME, Encrypt-Mix-Encrypt constructions [14] with linear mixing in the middle layer, and hence parallelizable.

Encrypt-then-Checksum. It uses IV-based block-wise encryption (non-repeating IV is required) and then finally checksum is used to compute tag. For example, different versions of OCB [5,30,22] and IAPM [21].

1.2 Encrypt Mix Encrypt

Encrypt Mix Encrypt or EME [14] is a block-cipher mode of operation, that turns a block cipher into a tweakable enciphering scheme. The mode is parallelizable, and as serial-efficient as the non-parallelizable mode CMC [13]. EME algorithm entails two layers of ECB encryption and a non-linear mixing in between. In the non-linear mixing, the blockcipher is again used. EME is proved to provide SPRP [23] security in the standard, provable security model assuming that the underlying block cipher is SPRP secure. Moreover, the designers of EME showed a CCA-distinguisher if non-linear mixing is replaced by a binary linear mixing.

1.3 Our Contribution

In this paper, we have observed that replacing non-linear mixing by an efficient online linear mixing actually helps to have faster and parallel implementation of the construction and gives online prp [23] security. (We know that, an online function is a function whose i^{th} block output is determined by the first i blocks of input) the Based on this observation, we have designed an online authenticated cipher ELmE based on Encrypt Mix Encrypt structure where the non-linear mixing is replaced by efficient online linear mix. ELmE has the following advantages over other popular authenticated schemes :

Nonce Misuse Resistant. Most of the IV based authenticated encryption schemes [31] like all the versions of OCB [5], GCM [35] needed to ensure that nonce must be distinct for every invocation of the tagged-encryption. Failure to do so, leads easy attacks on the privacy of the scheme. In practice, it is challenging to ensure that the nonce is never reused. For example, in lightweight applications, it is quite challenging to generate distinct nonce as it either needs to store a non-tamperable state or require some hardware source of randomness. Apart from that, there are various issues like flawed implementations or bad management by the user, for example where users with same key uses the same nonce. Our construction ELmE does not have the distinct nonce requirement, instead it generates an IV from the associated data. In section 4, we prove that, ELmE provides **online privacy** under IV repeation and **full privacy** when distinct IVs are used.

Fully Pipeline Implementable. Most of the popular online constructions like McOE [11] (uses MHCBC [25], later generalized and called TC3 [32]) has a hardware bottleneck of not being fully pipelined (see the bottom layer of McOE in Figure 1. It has CBC like structure, which is sequential and hence can not be pipelined). Our construction ELmE has a Encrypt-Linear mix-Decrypt type structure, making it fully parallel and pipeline implementable.

Efficient. Deterministic AE Schemes (for example : SIV, BTM, HBS) doesn't use any nonce. Instead it uses a derived IV using the message and the

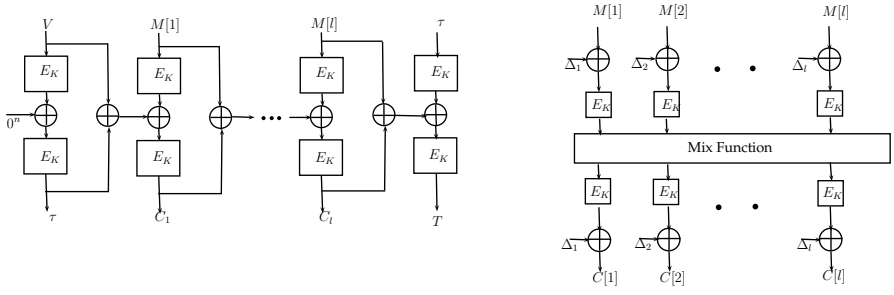

Fig. 1. (1) McOE-D construction : cannot be pipelined. (2) Encrypt-Mix-Encrypt : completely parallel and pipeline implementable.

associated data, which ensures that it is distinct for each different associated data-message tuples but such constructions are two passed, and hence not efficient. Having Encrypt- Linear mix-Encrypt type layered design, makes our construction single pass and efficient.

Minimized Area in Combined Implementation. The construction of ELmE ensures that encryption and decryption behave almost in a similar fashion (see figure 3 and remark 2 in section 3). This helps us to implement both encryption and decryption in hardware with a smaller area. Nowadays in all application environment, both encryption and decryption of blockciphers to be implemented and hence we can share the architectures to have a compact combined hardware implementation of it.

Secure against Block-wise Adaptive Adversaries. Due to limited memory in some environment such as low end devices the decryption oracle has to release a part of the plaintext before it authenticates. That raises some attacks on popular constructions [20]. We consider similar advantages such as privacy and authenticity, however the adversaries (called blockwise adaptive adversary) would have access of partial decryption oracles for authenticity security. To resist such attacks, intermediate tags can be used. In section 5, we have shown that ELmE can be extended to incorporates intermediate tags, hence it provides security against Block-wise adaptive adversaries.

2 Preliminaries

Definitions and Notation. By convention, $\mathbb{B} = \{0,1\}^n$ where n is the block size of the underlying blockcipher. An ℓ-tuple $x \in \mathbb{B}^\ell$ is denoted by $(x[1], x[2], \ldots, x[\ell])$. We call $\ell := \|x\|$ block-length of x. For $0 \leq a \leq b < \ell$ we denote $x[a..b] := (x[a], x[a+1], \ldots, x[b])$, $x[..b] = x[1..b]$. A plaintext P is represented as a tuple (D, M) where M is the message and D is the associated data and the corresponding ciphertext is represented as (C, T) where C is the ciphertext and T is the generated tag.

2.1 Full and Online Privacy

We give a particularly strong definition of privacy, one asserting indistinguishability from random strings. Consider an adversary A who has access of one of two types of oracles: a "real" encryption oracle or an "ideal" authenticated encryption oracle. A real authenticated encryption oracle, F_K, takes as input (D, M) and returns $(C, T) = F_K(D, M)$. Whereas an ideal authenticated encryption oracle \$ returns a random string R with $\|R\| = \|M\| + 1$ for every fresh pair (D, M). Given an adversary A (w.l.o.g. throughout the paper we assume a **deterministic adversary**) and an authenticated encryption scheme F, we define the (full) **privacy-advantage** of A by the distinguishing advantage of A distinguishing F from \$. More formally,

$$\mathbf{Adv}_F^{\text{priv}}(A) := \mathbf{Adv}_F^{\$}(A) = \mathsf{Pr}_K[A^{F_K} = 1] - \mathsf{Pr}_{\$}[A^{\$} = 1].$$

We include initial value IV as a part of associated data D and so for nonce-respecting adversary A (never repeats a nonce or initial value and hence the view obtained by the adversary is nonce-respecting) the response of ideal oracle for every query is random as all queries are fresh. Similarly, we define online privacy for which the the ideal online authenticated encryption oracle $\$_{ol}$ responses random string keeping the online property. The online privacy advantage of an adversary A against F is defined as $\mathbf{Adv}_F^{\text{opriv}}(A) := \mathbf{Adv}_F^{\$_{ol}}(A)$.

View and A-realizable. We define view of a deterministic adversary A interacting with an oracle $\mathcal{O}$ by a tuple $\tau(A^{\mathcal{O}}) := (Q_1, R_1, \ldots, Q_q, R_q)$ where Q_i is the i^{th} query and R_i is the response by $\mathcal{O}$. It is also called $\mathcal{O}$-view. A tuple $\tau = (Q_1, R_1, \ldots, Q_q, R_q)$ is called A-realizable if it makes query Q_i after obtaining all previous responses $R_1, \ldots, R_{i-1}$. As A is assumed to be deterministic, given $R_1, \ldots, R_q$, there is an unique q-tuple $Q_1, \ldots, Q_q$ for which the combined tuple is A-realizable. Now we describe the popular coefficient H-technique [27] which can be used to bound distinguish advantage. Suppose f and g are two oracles and V denotes all possible A-realizable views while A interacts with f or g (they have same input and output space).

Lemma 1 (Coefficient H Technique). *If $\forall v \in V_{good} \subseteq V$ (as defined above), $\mathsf{Pr}[\tau(A^g(\cdot)) = v] \geq (1 - \epsilon)\mathsf{Pr}[\tau(A^f(\cdot)) = v]$, then the distinguishing advantage $\mathbf{Adv}_g^f(A)$ of A is at most $\epsilon + \mathsf{Pr}[\tau(A^f(\cdot)) \notin V_{good}]$.*

We skip the proof as it can be found in many papers, e.g. [27,36].

2.2 Authenticity

We say that an adversary A **forges** an authenticated encryption F if A outputs (D, C, T) where $F_K(D, C, T) \neq \perp$ (i.e. it accepts and returns a plaintext), and A made no earlier query (D, M) for which the F-response is (C, T). It can make s attempts to forge after making q queries. We define that A forges if it makes at least one forges in all s attempts and the **authenticity-advantage** of A by

$$\mathbf{Adv}_F^{\text{auth}}(A) = \mathsf{Pr}_K[A^{F_K} \text{ forges}].$$

Suppose for any valid tuple of associate data and tagged ciphertext (D, C, T), the tag T can be computed from (D, C). We write $T = T_K(D, C)$. So (D, C, T) is a valid tagged ciphertext if and only if $T_K(D, C) = T$. Almost all known authenticated encryptions F (including those following encrypt-then-mac paradigm) have this property for a suitably defined ciphertext C and tag function T. We know that PRF implies Mac. We use similar concept to bound authenticity. More formally, for any forgery B, there is a distinguisher A such that

$$\mathbf{Adv}_F^{\text{auth}}(B) \leq \mathbf{Adv}_{(F,T)}^{\mathcal{O},\$}(A) + \frac{s}{2^n} \tag{1}$$

where $\mathcal{O}$ and $\$$ are independent oracles and $\$$ is a random function. This can be easily seen by defining A as follows:

- A first makes the q many F-queries (D_i, M_i) which are made by B and obtains responses (C_i, T_i), $1 \leq i \leq q$.
- Then it makes s many T-queries (D_j, C_j), $q < j \leq q+s$ where (D_j, C_j, T_j)'s are returned by B.
- A returns 1 (interpreting that interacting with real) if and only if $T(D_j, C_j) = T'_j$ for some j.

The distinguishing advantage of A is clearly at least $\mathsf{Pr}[B \text{ forges}] - \frac{s}{2^n}$ and hence our claim follows.

Trivial Queries. As $F(D, M) = (C, T)$ implies that $T(D, C) = T$, we call such T-query (D, C) trivial (after obtaining response (C, T) response of the F-query (D, M)). The repetition of queries are also called trivial. Without loss of generality, we assume that all adversaries A is **deterministic and does not make any trivial query**. These assumptions are useful to simplify the analysis.

3 ELmE: An Online Authenticated Encryption Algorithm

In this section, we demonstrate our new construction ELmE. It is an online authenticated encryption which takes an associated data $D \in \mathbb{B}^d$ and a messages $M \in \mathbb{B}^e$ and returns a tagged-ciphertext $C \in \mathbb{B}^{e+1}$ for all integers $d \geq 1$, $e \geq 1$. We assume associated data to be non-empty. The case when the associated data is empty, is taken care in the remark 1. To process incomplete blocks, one can either apply an injective padding rule (e.g., first pad 1 and then a sequence of zeros to make the padded message or associate data size multiple of n) or some standard methods (e.g., ciphertext stealing [9], the method used in Hash Counter Hash type constructions [10], XLS [29] etc.). It uses Encrypt-Mix-Encrypt type construction with a specified simple linear mixing (see in Algorithm 1) and a keyed block cipher $E_k : \mathbb{B} \to \mathbb{B}$ for the ECB layers. The ECB layers are masked by separate keys L_1 (for associated data), L_2 (for the message) and L_3 (for the ciphertext) chosen uniformly from $\mathbb{B}$. However, L_1, L_2, L_3 can be simply computed from E_k as $E_K(0) = L_1$, $E_K(1) = L_2$, $E_k(2) = L_3$ and can be preprocessed. The complete construction is described below in Algorithm 1 and illustrated in Fig. 2 below.

Remark 1 (Case when Associated data is empty). Here we consider the case when the associated data is non empty, using the initial value of the sequence $W[0] = 0$, one can have a trivial attack against the privacy of the construction : Query any message M_1 with $M_1[1] = 0$. It produces the ciphertext with $C_1[1] = L_2 + L_3$. Now querying any message M_2 with $M_2[1] = C_1[1]$ will produce $C_2[1] = 0$ with probability 1.

Note that, Algorithm 1 is defined for non-empty associated data. One can ensure associated data to be non-empty by including a non-empty public message number, in the first block of the associated data. Still, if we want to incorporate empty associated data in our algorithm, we make a small modification and initialize the value $W[0]$ to 1, to resist against any attack. The rest computations, to generate the tagged ciphertext, are identical to the above algorithm.

Input: $(D, M) \in \mathbb{B}^d \times \mathbb{B}^e$
Output: $Z = (C, T) \in \mathbb{B}^e \times \mathbb{B}$

Algorithm ELmE(D, M) (**Key**: (L_1, L_2, L_3, K))
`parse` D and M into n-length blocks.
1 $D = D[1] \| \cdots \| D[d]$
2 $M = M[1] \parallel M[2] \parallel \cdots \parallel M[e]$
3 $W[0] = 0$
4 $M[e+1] = D[1] + \cdots + D[d] + M[1] + \cdots + M[e]$ (**checksum**)
`process` D
5 **For all** $j = 1$ **to** d
6 $DD[j] = D[j] + \alpha^{j-1}.L_1$ **(Masking the associate data blocks)**
7 $Z[j] = E_K(DD[j])$ **(Layer-I Encryption)**
8 $(Y'[j], W[j]) \leftarrow \rho(Z[j],\, W[j-1])$ **(Linear Mixing)**
`process` M
9 **For all** $j = 1$ **to** e
10 $MM[j] = M[j] + \alpha^{j-1}.L_2$ **(Masking the message blocks)**
11 $X[j] = E_K(MM[j])$ **(Layer-I Encryption)**
12 $(Y[j], W[d+j]) \leftarrow \rho(X[j],\, W[d+j-1])$ **(Linear Mixing)**
13 $CC[j] = E_K^{-1}(Y[j])$ **(Layer-II Encryption)**
14 $C[j] = CC[j] + \alpha^{j-1}.L_3$ **(Masking the ciphertext blocks)**
`Tag generation`
15 $MM[e+1] = M[e+1] + \alpha^{e}.L_2$
16 $X[e+1] = E_K(MM[e+1])$
17 $(Y[e+1], W[d+e+1]) \leftarrow \rho(X[d+e+1],\, W[d+e])$
18 $TT = E_K^{-1}(Y[e+1] + 0^{n-1}1)$
19 $T \;=\; TT + \alpha^{e}.L_3$
20 **Return** $(C = C[1] \parallel C[2] \parallel \cdots \parallel C[e], T)$

Subroutine $\rho(x, w)$ `Onlinear Linear Mixing Function`
21 $y = x + (\alpha + 1) \cdot w$
22 $w = x + \alpha \cdot w$
23 **Return** (y, w)

Algorithm 1. ELmE Authenticated Encryption Algorithm. Here α is a primitive element of the binary field $(GF(2^n), +, .)$.

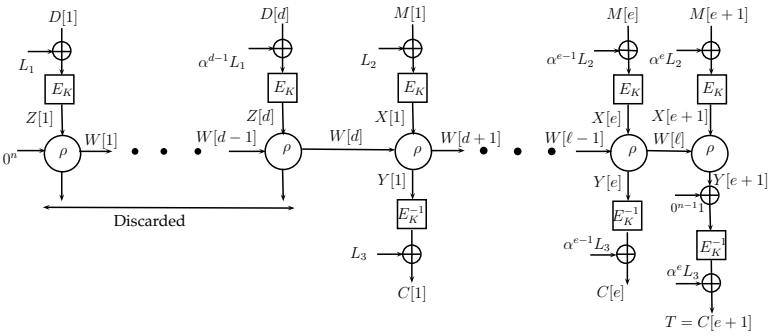

Fig. 2. Construction of ELmE Authenticated Encryption

Remark 2 (Similarity in Encryption and Decryption). Observe that, the second ECB layer is based on blockcipher decryption instead of encryption. Due to this, both encryption and decryption behave almost in a similar fashion (only with few changes in masking layers due to different keys and in linear mixing which should be inverse of the forward mixing). This helps us to implement both encryption and decryption in hardware with a smaller area.

Note that, from the definition of ρ, we see that the following online linear mixing has been performed :

When d is non-empty :

$$Y[i] = \alpha^{d+i-2}(\alpha+1)Z[1] + \cdots + \alpha^{i-1}(\alpha+1)Z[d] \\ +\alpha^{i-2}(\alpha+1)X[1] + \alpha^{i-3}(\alpha+1)X[2] + \cdots + (\alpha+1)X[i-1] + X[i]$$

When d is empty :

$$Y[i] = \alpha^{i-2}(\alpha+1)X[1] + \alpha^{i-3}(\alpha+1)X[2] + \cdots + (\alpha+1)X[i-1] + X[i] + \alpha^{i-1}(\alpha+1)$$

4 Privacy and Authenticity of ELmE

To prove the online Privacy of ELmE, let A be an adversary which makes at most q queries $\{(D_i, M_i)\}_{1 \le i \le q}$ in order to distinguish it from an online function, with same domain and range size chosen uniformly at random. Assume $\|D_i\| = d_i$, $\|M_i\| = e_i$. Let $\sigma_{\text{priv}} = \sum_{i=1}^{q}(d_i + e_i + 1)$ (the total number of blocks processed). Let $\$_{perm}$ denotes the random n-bit permutation and $\eta_{priv} := \max_B \mathbf{Adv}^{\$_{perm},\$_{perm}^{-1}}_{E,E^{-1}}(B)$ denotes the maximum advantage over all adversaries B making at most σ_{priv} queries and running in time T_0 which is about time of the adversary A plus some overhead which can be determined from the hybrid technique. The advantage of A is given by,

Theorem 1

$$\mathbf{Adv}^{\text{opriv}}_{\mathsf{ELmE}_{\Pi,\mathbf{L}}}(A) \;\leq\; \frac{5\sigma^2_{\text{priv}}}{2^n}, \qquad \mathbf{Adv}^{\text{opriv}}_{\mathsf{ELmE}_{E_K,\mathbf{L}}}(A) \;\leq\; \eta_{priv} + \frac{5\sigma^2_{\text{priv}}}{2^n}.$$

On the other hand, to show authenticity of the construction, let A be an adversary which makes q queries $\{(D_i, M_i)\}_{1\leq i\leq q}$ and tries to forge against the construction at most s times with queries $\{(D_i, C_i, T_i)\}_{q+1\leq i\leq q+s}$. For all i, let us denote $\|D_i\| = d_i$, $\|M_i\| = \|C_i\| = e_i$. Suppose $\sigma_{\mathbf{auth}} = \sum_{i=1}^{q+s}(d_i + e_i + 1)$. The forging advantage of ELmE is given by:

Theorem 2

$$\mathbf{Adv}^{\text{forge}}_{\mathsf{ELmE}_{\Pi,\mathbf{L}}}(A) \;\leq\; \frac{9\sigma^2_{\text{auth}}}{2^n} + \frac{s}{2^n}, \qquad \mathbf{Adv}^{\text{forge}}_{\mathsf{ELmE}_{E_K,\mathbf{L}}}(A) \;\leq\; \eta_{auth} + \frac{9\sigma^2_{\text{auth}}}{2^n} + \frac{s}{2^n}.$$

where η_{auth} is exactly same to η_{priv} except that it can make atmost σ_{auth} queries.

4.1 Proof of Theorem 1

First part of the theorem follows using the coefficient H technique (see Lemma 1) and following Propositions 1 and 2. Second part follows from the standard hybrid argument.

Let us fix q message and associate data pairs $P_1 = (D_1, M_1), \ldots, P_q = (D_q, M_q)$ with $\|D_i\| = d_i, \|M_i\| = e_i$, $\ell_i = d_i + e_i$ and $\sigma = \sum_i \ell_i$. We denote $(P_1, \ldots, P_q)$ by τ_{in}. We assume that all P_i's are distinct.

Definition 1 (Good views). *A tagged ciphertext tuple $\tau_{out} = (C_1, \ldots, C_q)$ (also the complete view $\tau = (\tau_{in}, \tau_{out})$) is called* **good** *online view (belongs to τ_{good}) w.r.t. τ_{in} if (τ_{in}, τ_{out}) is an online view (i.e., it must be realized by an online cipher, see section 2) and the following conditions hold:*

1. *$C_i[j] = C_{i'}[j]$ implies that $D_i = D_{i'}$, $M_i[..j] = M_{i'}[..j]$ and*
2. *$\forall\ (i, l_i + 1) \neq (i', j')$, $T_i \neq C_{i'}[j']$.*

The first condition says that we can have collision of ciphertext blocks in a position only if they are ciphertexts of two messages with same prefixes up to that block. The second conditions says that all tag blocks are fresh as if these are independently generated. It is easy to check that, in case of ideal online cipher, generating a bad view (i.e. not a good view) has negligible probability:

Proposition 1 (Obtaining a Good view has high probability)

$$\mathsf{Pr}[\tau(A^{\$_{ol}}) \notin \tau_{good}] \;\leq\; \frac{\sigma^2_{\text{priv}}}{2^n}.$$

We Now Fix a Good View $\tau = (\tau_{in}, \tau_{out})$ as Mentioned above. The tagged ciphertext of P_i is given by C_i which has $e_i + 1$ blocks where the last block $T_i := C_i[e_i + 1]$ denotes the tag. In the following result, we compute the interpolation probability, i.e. $\mathsf{Pr}[\tau(A^F) = \tau]$.

Proposition 2 (High interpolation probability of ELmE). $\forall \tau \in V_{good}$,

$$\mathsf{Pr}[\tau(A^{\mathsf{ELmE}_{\Pi,\mathbf{L}}}) = \tau] \geq (1 - \tfrac{4\sigma_{\text{priv}}^2}{2^n}) \times \mathsf{Pr}[\tau(A^{\$^{ol}}) = \tau].$$

Note that $\mathsf{Pr}[\tau(A^{\$^{ol}}) = \tau] = 2^{-nP}$ where P denotes the number of non-empty prefixes of (D_i, M_i), $1 \leq i \leq q$ as for every different prefixes, $\$^{ol}$ assigns an independent and uniform ciphertext blocks. Proof of the above proposition can be found in the full version [7].

Remark 3. If associated datas are distinct for all the q messages, then $P = \sigma_{priv}$ and hence, we'll have full privacy i.e. the construction becomes indistinguishable from a random cipher with same domain and range.

4.2 Proof of Theorem 2

First part of theorem 2 follows using the coefficient H technique (see Lemma 1) and following Propositions 3 and 4 and then using equation 1. Second part follows from the standard hybrid argument.

Let $\mathbf{L} = (L_1, L_2, L_3)$ be the triple of masking keys and Π be the uniform random permutation. For notational simplicity, we write $\mathsf{ELmE}_{\Pi,\mathbf{L}}$ by F. Note that for a valid tuple of associate data and tagged ciphertext (D, C, T), the tag T can be computed from C and the key. We write $T = T_{\Pi,\mathbf{L}}(D, C) := T(D, C)$. So (D, C, T) is a valid tagged ciphertext if and only if $T(D, C) = T$. As we have observed in Eq. 1, we only need to show indistinguishability for which we apply the coefficient H technique again. For this, we need to identify set of good views for which we have high interpolation probability.

Good Forge View. A (F, T)-forge view of a distinguisher A is the pair $\tau = (\tau_F, \tau_T)$ where $\tau_F = (D_i, M_i, C_i, T_i)_{1 \leq i \leq q}$ is an q-tuple of F-online view and $\tau_F = (D_j, C_j, T_j)_{q < j \leq q+s}$ is an s-tuple non-trivial T-view. τ is called **good forge view** (belongs to τ_{good}) if τ_F is good (as defined in Definition 1) and for all $q < j \leq q + s$, T_j's are fresh - distinct and different from all other T_i's and $C_i[j]$'s. We recall the notation $|M_i| = e_i$, $|D_i| = d_i$ and $\ell_i = d_i + e_i$. Let $\sigma_{\text{auth}} = \sum_{i=1}^{q+s}(\ell_i + 1)$. Since F is online function we consider pair of independent oracles $(\$_{ol}, \$)$ where $\$_{ol}$ denotes the random online function and $\$$ is simply a random function.

Proposition 3 (Obtaining a good forge view has high probability)

$$\mathsf{Pr}[\tau(A^{\$_{ol},\$}) \in \tau_{good}] \leq \frac{(q + \sum_{i=1}^{q} e_i)^2}{2^{n+1}} + \frac{s(q + s + \sum_{i=1}^{q+s} e_i)}{2^n} \leq \frac{2\sigma_{\text{auth}}^2}{2^n}.$$

The first summand takes care the collisions in $C_i[j]$'s (i.e., the bad view for τ_F as in Proposition 1) and the second summand takes care the collision between T_i's $(q < i \leq q + s)$ and all other $C_i[j]$'s.

Now we fix a good view $\tau = (\tau_F, \tau_T)$ as defined above (following same notations). It is easy to see that obtaining τ interacting with $(\$_{ol}, \$)$ has probability

$2^{-ns} \times 2^{-n\sigma_{pf}} = 2^{-n(s+\sigma_{pf})}$ where σ_{pf} denotes the number of non-empty prefixes of (C_i, T_i), $1 \leq i \leq q$ (at those blocks random online function returns randomly). Now, one can show the following result :

Proposition 4 (Good forge view has high interpolation probability). *For any good (F,T)-view τ and $\epsilon' = \frac{7\sigma^2_{\text{auth}}}{2^n}$, we have*

$$\mathsf{Pr}[F(D_i, M_i) = (C_i, T_i), 1 \leq i \leq q,\ T(D_j, C_j) = T_j, q < j \leq q+s] \ \geq\ (1-\epsilon')2^{-n(\sigma_{pf}+s)}.$$

Proof of this proposition can be found in the full version [7].

5 ELmE Incorporating Intermediate Tags

Intermediate tags can be used in authenticated encryption to provide quick rejection of invalid decryption queries. This also helps in low-end implementation where the message has to be released depending on buffer size. If we have an intermediate tag in appropriate positions so that we can reject before we release some message blocks. Our construction can be easily extended to produce intermediate tags also, as described in the figure below. Suppose, we want ELmE with intermediate tags generated after each k blocks. In this case, for a message $M \in \mathbb{B}^e$, ELmE generates a ciphertext $C \in \mathbb{B}^e$ and $T \in \mathbb{B}^h$ where $h = \lceil \frac{e}{k} \rceil$. Processing of D remains same. For Processing of M, the calculation of $C[j]$ is changed to $CC[j] + \alpha^{j-1+\lfloor \frac{j-1}{k} \rfloor}.L_3$. $\forall\ j < e$ s.t. $k|j$, the intermediate tags are generated by $T[\frac{j}{k}] = E_K^{-1}(W[d+j]) + \alpha^{j-1+\lceil \frac{j-1}{k} \rceil}.L_3$. Final tag $T[h]$ is generated similar to the generation of T in the case of ELmE without intermediate tags (Here $\alpha^{e+h-1}L_3$ is used as the mask). Tag T is given by $T[1] \,||\, T[2] \,||\, \cdots \,||\, T[h]$. For verification during decryption, each $T[i]$ is verified and as soon as, a $T[i]$ doesn't matched with it's calculated value, the ciphertext gets rejected. Here, we have used intermediate tags after processing of each $k < n$ blocks of message. Let $\mathbf{F}$ be our construction incorporating intermediate tags after each $k < n$ blocks. In the following subsection, we show the privacy and authenticity of $\mathbf{F}$.

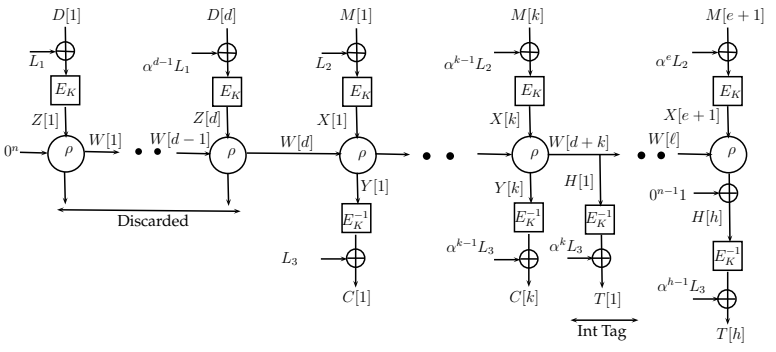

Fig. 3. ELmE with intermediate tags

Remark 4. Sponge duplex [6], is another authenticated encryption that incorporates intermediate tags but the dependency is such that, during decryption, the plaintext depends on the values of the intermediate tags. In our construction, during decryption, the plaintext does not depend on the intermediate tags and hence the extra computations required for the intermediate tags, can be skipped, if intermediate verifications are not required.

5.1 Online Privacy and Authenticity of F

Let A be an adversary which makes q queries (D_i, M_i) and obtains responses (C_i, T_i), $1 \leq i \leq q$. We denote $\|D_i\| = d_i$, $\|M_i\| = \|C_i\| = e_i$ and $\|T_i\| = h_i$. Let $\sigma_{\mathbf{priv}} = \sum_{i=1}^{q}(d_i + e_i + h_i)$ (the total number of ciphertext blocks with the tag blocks). The online Privacy of $\mathbf{F}$ is given by:

Theorem 3

$$\mathbf{Adv}^{\text{opriv}}_{\mathbf{F}^{\Pi,\mathbf{L}}}(A) \leq \frac{5\sigma^2_{\mathbf{priv}}}{2^n}, \quad \mathbf{Adv}^{\text{opriv}}_{\mathbf{F}^{E_K,\mathbf{L}}}(A) \leq \eta_{\mathbf{priv}} + \frac{5\sigma^2_{\mathbf{priv}}}{2^n}.$$

On the other hand, let A be an adversary which makes q queries $\{(D_i, M_i)\}_{1\leq i\leq q}$ and tries to forge against the construction at most s times with queries $\{(D_i, C_i, T_i)\}_{q+1\leq i\leq q+s}$. For all i, let us denote $\|D_i\| = d_i$, $\|M_i\| = \|C_i\| = e_i$ and $\|T_i\| = h_i$. Suppose $\sigma_{\mathbf{auth}} = \sum_{i=1}^{q+s}(d_i + e_i + h_i)$ (the total number of ciphertext blocks with the tag blocks). The forging advantage of F is given by:

Theorem 4

$$\mathbf{Adv}^{\text{forge}}_{\mathbf{F}^{\Pi,\mathbf{L}}}(A) \leq \frac{10\sigma^2_{\mathbf{auth}}}{2^n} + \frac{s}{2^n}, \quad \mathbf{Adv}^{\text{forge}}_{\mathbf{F}^{E_K,\mathbf{L}}}(A) \leq \eta_{\mathbf{auth}} + \frac{10\sigma^2_{\mathbf{auth}}}{2^n} + \frac{s}{2^n}.$$

The proofs of Theorem 3 and 4 are skipped due to page limit and can be found in the full version of the paper [7].

5.2 Including Intermediate Tags : Comparison with COPA

Intermediate tags are used to provide block-wise security. Suppose we consider a construction with intermediate tag size of k blocks. At each k blocks, we check the intermediate tag, hold the k block message and finally release the k blocks of the message if the tag is verified. For that, we need to store all the intermediate computations and the already computed messages in order to perform the verification. As we are using low end device, we need to minimize the buffer size.

Now, generating intermediate tags for COPA is not as straight forward as ELmE as similar approach won't provide any security because identical last two blocks will produce same intermediate tag.

Moreover, we claim that even if intermediate tags are produced for COPA as if the final tag, then it also has the disadvantage of requiring additional buffer storage. Now we compare the 20 round pipeline implementations which is keeping

Table 1. Comparative study on the performance of block-cipher based Authenticated Encryptions. Here #BC AD, #BC M and #BC T denotes no. of block-cipher call per associated data, message and tag block respectively.

Construction	#BC AD	#BC M	#BC T	speed up	Misuse Resistance	Bottleneck
OCB	1	1	1	p	No	Nonce Processing
McOE-D	1	2	2	2	Yes	Lower level Processing
CoPA	1	2	2	p	Yes	Associated data Processing
ELmE	1	2	1	p	Yes	None

computing the messages even after intermediate tag to keep the pipeline full. For each k block of intermediate tags, the pipelined implementation of 20 round AES for COPA requires to store k block messages and in addition 20 blocks of intermediate values for the subsequent ciphertext blocks. On the other hand ELmE requires k blocks messages and 10 blocks of intermediate computation for next 10 next subsequent ciphertext. We save 10 blocks in buffer mainly due to faster verification (ELmE verifies after one layer, whereas COPA verifies after two layers). It has great advantage for low-end devices (keeping in mind that, block-wise adversaries are considered only when buffer size is limited implying low-end device). Keeping the above benefits into consideration, we opt for the linear mix ρ function rather than using a simple xor operation, as used in COPA.

6 Conclusion and Future Works

In the following paragraph, we mainly provide theoretical comparisons of OCB3, McOE-D, COPA and our construction ELmE. All the constructions have same key size and similar number of random mask (which can be preprocessed) for masking layers. The number of blockcipher calls for processing every message, associate data and tag blocks are given in the Table 1. The speed up for OCB, COPA and ELmE is p with parallel implementations by p processors as their construction support parallel execution. Due to the sequential nature of the lower level of McOE-D, the speed up factor can be at most 2.

Now, we briefly discuss bottlenecks issues of the other constructions, that our construction overcome.

OCB versions are IV based constructions and require distinct nonce in each invocation, hence not misuse resistant. Moreover OCB3 (which has minimum bottleneck among all versions) has a bottleneck in the nonce processing. As the encryption of the IV is needed in the masking of the messages, hence the encryption of the messages can start only after the encryption of IV, hence has the bottleneck of having additional clock cycles required for one block encryption.

As already mentioned in section 1, McOE-D uses TC3 type encryption and it's lower level has a CBC type structure which can not be executed in parallel implying the construction can not be pipelined. Hence it has a hardware bottleneck.

COPA has the bottleneck during the processing of associated data, as the last blockcipher input depends on the previous blockcipher outputs. Hence, the last block cipher invocation must be done after the completion of all the block-cipher invocations, making it sequential for one block-cipher invocation. So, complete parallelization can not be achieved.

On the other hand, our construction ELmE is completely parallel with no such bottleneck as described above. Moreover the construction treats the additional data and message exactly in a similar way (except with different masking keys). The encryption and decryption also behave similarly and hence ensures less chip area in combined hardware implementation. Moreover, to resist against blockwise adversaries, ELmE can incorporate intermediate tags very efficiently, which the other constructions do not take care of and could be hard to generate.

Note that, the above comparison is given from theoretical point of view. Experimental measurements to support these claim is a possible future scope. We've planned to implement a portable reference software implementation of our cipher as well as include a reference hardware design in verilog.

Acknowledgement. This work is supported by the Centre of Excellence in Cryptology (CoEC), Indian Statistical Institute, Kolkata.

References

1. (no editor), CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness, `http://competitions.cr.yp.to/caesar.html`, Citations in this document: §1.1, §1.1
2. (no editor), Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 and 128-EIA3. Document 2: ZUC Specification. ETSI/SAGE Specification, Version: 1.5 (2011), Citations in this document: §1.1
3. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 424–443. Springer, Heidelberg (2013), Citations in this document: §1.1
4. 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), Citations in this document: §1.1
5. Bellare, M., Blake, J., Rogaway, P.: OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption 6, 365–403 (2005), Citations in this document: §1.3
6. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Duplexing the Sponge: Single Pass Authenticated Encryption and Other Applications. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 320–337. Springer, Heidelberg (2012), Citations in this document: §1.1, §4

7. Datta, N., Nandi, M.: Misuse Resistant Parallel Authenticated Encryptions, IACR Cryptology ePrint Archive (2013), http://eprint.iacr.org/2013/767.pdf, Citations in this document: §4.1, §4.2, §5.1
8. Gligor, V.D., Donescu, P.: Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 92–108. Springer, Heidelberg (2002), Citations in this document: §1.1
9. Dworkin, M.: Recommendation for block cipher modes of operation: three variants of ciphertext stealing for CBC mode. Addendum to NIST Special Publication 80038A (2010), Citations in this document: §3
10. 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), Citations in this document: §3
11. 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), Citations in this document: §1.1, §1.3
12. Fouque, P.-A., Joux, A., Martinet, G., Valette, F.: Authenticated On-Line Encryption. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 145–159. Springer, Heidelberg (2004)
13. Halevi, S., Rogaway, P.: A Tweakable Enciphering Mode. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 482–499. Springer, Heidelberg (2003), Citations in this document: §1.2
14. Halevi, S., Rogaway, P.: A parallelizable enciphering mode. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 292–304. Springer, Heidelberg (2004), Citations in this document: §1.1, §1.2
15. Hell, M., Johansson, T., Maximov, A., Meier, W.: A Stream Cipher Proposal: Grain-128, eSTREAM, ECRYPT Stream Cipher Project, Report 2006/071 (2005), http://www.ecrypt.eu.org/stream, Citations in this document: §1.1
16. Housley, R., Whiting, D., Ferguson, N.: Counter with CBC-MAC, CCM, RFC 3610 (Informational) (2003), Citations in this document: §1.1
17. Iwata, T.: New blockcipher modes of operation with beyond the birthday bound security. In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 310–327. Springer, Heidelberg (2006), Citations in this document: §1.1
18. 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), Citations in this document: §1.1
19. Iwata, T., Yasuda, K.: A Single-Key, Inverse-Cipher-Free Mode for Deterministic Authenticated Encryption. In: Jacobson Jr., M.J., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 313–330. Springer, Heidelberg (2009), Citations in this document: §1.1
20. Joux, A., Martinet, G., Valette, F.: Blockwise-Adaptive Attackers: Revisiting the (In)Security of Some Provably Secure Encryption Models: CBC, GEM, IACBC. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 17–30. Springer, Heidelberg (2002), Citations in this document: §1.3
21. Jutla, C.S.: Encryption Modes with Almost Free Message Integrity. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 529–544. Springer, Heidelberg (2001), Citations in this document: §1.1, §1.1
22. Krovetz, T., Rogaway, P.: The Software Performance of Authenticated-Encryption Modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer, Heidelberg (2011)

23. Luby, M., Rackoff, C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal of Computing, 373–386 (1988), Citations in this document: §1.2, §1.3
24. Lucks, S.: Two Pass Authenticated Encryption Faster than Generic Composition. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 284–298. Springer, Heidelberg (2005), Citations in this document: §1.1
25. Nandi, M.: Two new efficient CCA-secure online ciphers: MHCBC and MCBC. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 350–362. Springer, Heidelberg (2008), Citations in this document: §1.3
26. Nandi, M.: A Generic Method to Extend Message Space of a Strong Pseudorandom Permutation. Computacin y Sistemas 12 (2009)
27. Patarin, J.: The "Coefficients H" technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 328–345. Springer, Heidelberg (2009), Citations in this document: §2.1
28. Preneel, B., Wu, H.: AEGIS: A Fast Authenticated Encryption Algorithm, Cryptology ePrint Archive: Report 2013/695
29. Ristenpart, T., Rogaway, P.: How to Enrich the Message Space of a Cipher. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 101–118. Springer, Heidelberg (2007), Citations in this document: §3
30. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer, Heidelberg (2004)
31. Rogaway, P.: Nonce-based symmetric encryption. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 348–359. Springer, Heidelberg (2004), Citations in this document: §1.3
32. Rogaway, P., Zhang, H.: Online Ciphers from Tweakable Blockciphers. In: CT-RSA, pp. 237–249 (2011), Citations in this document: §1.3
33. 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), Citations in this document: §1.1
34. Sarkar, P.: On Authenticated Encryption Using Stream Ciphers Supporting an Initialisation Vector. IACR Cryptology ePrint Archive, 299–299 (2011), `http://eprint.iacr.org/2011/299.pdf`; capsulating Security Payload (ESP), Citations in this document: §1.1
35. Viega, J., McGraw, D.: The use of Galois/Counter Mode (GCM) in IPsec En, RFC 4106 (2005), Citations in this document: §1.1, §1.3
36. Vaudenay, S.: Decorrelation: A Theory for Block Cipher Security. Journal of Cryptology, 249–286 (2003)