Efficient Lattice-based Authenticated Encryption: A Practice-Oriented Provable Security Approach

Ahmad Boorghany, Siavash Bayat-Sarmadi, *Member, IEEE,* and Rasool Jalili

Abstract—Lattice-based cryptography has been received significant attention in the past decade. It has attractive properties such as being a major post-quantum cryptography candidate, enjoying worst-case to average-case security reductions, and being supported by efficient implementations. In recent years, lattice-based schemes have achieved enough maturity to become interesting also for the industry. Additionally, authenticated encryption (AE) is another important topic in the community of cryptography. In this paper, considering two above-mentioned subjects, we propose three lattice-based AEs with an acceptable practical efficiency. These schemes are provably secure assuming the hardness of elementary lattice problems. That is in contrast to the other practical provably-secure AEs, which are based on the hardness assumption of another cryptographic primitive, such as AES. Moreover, we analyze the exact security of these schemes in the paradigm of practice-oriented provable security, while the security proofs of almost all previous lattice-based schemes are asymptotic. The implementation results show that one of the proposed schemes becomes even faster than an AES-256-GCM implementation to encrypt messages of length 64 bytes or longer. Particularly, for a 1500-byte message, this scheme is 34% faster than AES-256-GCM.

Index Terms—Lattice-based cryptography, post-quantum cryptography, authenticated encryption, practice-oriented provable security, exact security analysis

1 Introduction

Lattice-based cryptography is one of the main candidates for post-quantum cryptography [1]. The cryptographic schemes based on lattice hard problems are conjectured to resist against the attacks by a large-scale quantum computer [2], while the schemes based on the hardness of integer factorization, integer discrete logarithm, and also elliptic curve discrete logarithm are completely vulnerable in this setting [3, 4]. Other advantages of the lattice-based schemes, in comparison to widely-used ones based on number theory (such as RSA and ECDSA), are as follows. Most of the state-of-the-art lattice-based schemes enjoy a worst-case to average-case security reduction (e.g., [5, 6, 7]). Thus, random instances of these schemes (i.e., with random keys and/or parameters) are provably as hard as *worst-case* instances of fundamental lattice problems. Finally, computations in the lattice-based schemes usually deal with simple operations on small integer vectors. This way, engineers become enabled to design efficient implementations on the hardware and software, and taking more advantage from hardware parallelism, processor pipelines, multiple cores, and the single-instruction multiple-data (SIMD) feature. As a result, modern implementations of the lattice-based schemes are much more efficient than the traditional alternatives [7, 8, 9, 10, 11, 12].

- *A. Boorghany and R. Jalili are with the Data and Network Security Lab (DNSL), Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. E-mail: boorghany@ce.sharif.edu, jalili@sharif.edu.*
- *S. Bayat-Sarmadi is with the Hardware Security and Trust (HST) Lab, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. E-mail: sbayat@sharif.edu.*

Additionally, authenticated encryption (AE) has been recently received significant attention in the community of cryptography. AE is a symmetric two-in-one solution for secure communications, as it provides both confidentiality and authenticity of the message. The CAESAR competition [13], started in 2013, is ongoing to select the best secure, applicable, and robust AE.

The majority of the current lattice-based schemes are asymmetric. That is mainly due to the computational and memory complexities of these schemes, which are usually not acceptable in the symmetric cryptography. Specifically, to the best of our knowledge, there is no report of any AE based on lattices. Moreover, the security assumptions of current practical provably-secure AEs are based on the security of another cryptographic primitive. For instance, AES-OCB [14] is an efficient provably-secure AE; however, it relies on the security of AES, for which the security is maintained heuristically.

Banerjee et al. [15] introduce a pseudorandom function (PRF) with a security proof based on lattice problems. PRFs are widely used in the design of symmetric cryptosystems. Later, Banerjee et al. [16] propose an efficient instantiation of this PRF called SPRING. They use SPRING in the counter (CTR) mode of operation to build a lattice-based symmetric encryption. Despite the use of the SIMD feature in high-end processors, their report of the SPRING implementation shows that the SPRING performance is interestingly comparable to the performance of AES.

In this paper, we introduce three authenticated encryption schemes with the security proofs based on the worst-case hardness of lattice problems. The proposed schemes, referred to as LAE1, LAE2, and LAE3, use the SPRING pseudorandom function as a building block. These AEs have the following particular advantages over the previous AE

constructions and also over other lattice-based schemes.

- Firstly, the proposed schemes are provably-secure not based on assuming the security of another cryptographic primitive, but on the hardness of well-studied mathematical problems. As far as we know, all previous AEs enjoying a security proof, are of the former type.
- Secondly, the proposed AEs are resistant to quantum attacks. The best known quantum attack to symmetric encryption schemes is based on the work of Grover [17]. In this attack, any generic symmetric encryption with an n-bit key can be broken in time $O(2^{n/2})$ using a quantum computer. Thus, to protect widely-used symmetric encryptions (and AEs) against large-scale quantum computers, it is only required to double the key size. However, the proposed lattice-based schemes are supported by another quantum resistance conjecture, which preserves even if a better quantum attack is found against other symmetric cryptosystems.
- Thirdly, as presented in Section 5, the performance of the best proposed scheme is comparable with conventional provably-secure AEs.

Finally, we analyze and prove the exact security of these lattice-based AEs in the paradigm of practice-oriented provable security. That is in contrast to the security proofs of most lattice-based schemes, which are asymptotic. Exact security analysis has important advantages. For instance, a large security gap is common in an asymptotic analysis, which may cause a provably secure scheme to be vulnerable in practice. Moreover, an exact security analysis helps practitioners to instantiate and parameterize a scheme more reliably.

The first proposed AE scheme is an encrypt-and-MAC (E&M) composition. On the encryption part, SPRING is used in a nonce-based variant of CTR mode. The authentication part, which ensures the authenticity of both nonce and message, is a CBC-MAC like XCBC [18], TMAC [19], and OMAC [20]. There are some challenges to adopt SPRING to be used in such cryptographic schemes. Most of these schemes are designed to use a pseudorandom permutation (PRP), which is the formal model of a block cipher. Some schemes, such as OCB mode of operation [14], are fundamentally dependent on the bijection property of PRPs, and cannot accept a PRF. Another challenge to utilize SPRING in cryptosystems is that the SPRING input and output sizes are not the same. It accepts 128 bits as input and provides 127 bits as output. That introduces a number of padding and truncation operations, which complicates the security proof. Moreover, the output is not a multiple of bytes and causes some obstacles in the implementation (see Section 5). Another challenge impacting the design of a scheme based on SPRING is that most significant optimizations can be achieved only by a certain configuration of SPRINGs (see Section 3.2).

The second proposed AE scheme is an encrypt-then-MAC (EtM) composition. The encryption part of the second scheme is similar to the previous one; however, the use of SPRING in the authentication part is minimized. Both of these AEs are two-pass, i.e., they are as inefficient as running encryption and MAC separately. The third proposed scheme is designed to be single-pass. Until recently, all single-pass AEs could not essentially use a PRF instead of a PRP. For instance, IACBC, IAPM [21], OCB variants [22, 23, 14], and stateful XECB and ECBC [24] cannot accept a PRF. The design of the third scheme is inspired by OTR [25]. Minematsu [25] proposes two variants of OTR, which accept both PRF and PRP. However, the third scheme cannot directly follow the PRF-compatible version of OTR because it requires a variable-input-length PRF (VILPRF), while the SPRING's input length is fixed. In this scheme, a tweakable PRF is created using SPRING.

The implementation results on the Intel Sandy Bridge and Haswell microarchitectures show that the second proposed scheme is efficient enough to be used in practice and compete widely-used AEs. On Sandy Bridge, although this scheme is 12% slower than AES-256-GCM [26] for encrypting 40-byte messages, it becomes faster for the messages of length 64 bytes or longer. Particularly, for a 1500-byte message, this scheme is 34% faster than AES-256-GCM. A similar result is also achieved on Haswell. The second proposed scheme is 24% slower for 40-byte messages in comparison with AES-256-GCM; however, it becomes faster for 128-byte messages and longer. Additionally, its performance is 15% better than AES-256-GCM for 1500-byte messages. The main comparison is performed with AES-256-GCM because the security of this scheme is 128 bits in the post-quantum setting. The performance comparison with AES-128-GCM is also presented in Section 5.

For the sake of simplicity, we have not considered common complementary AE features in the three proposed AE schemes, such as the support of associated data (AEADs) [27], nonce misuse-resistance [28], and online environment requirements.

1.1 Organization

Section 2 introduces the preliminary cryptographic definitions and notations used in this paper. The proposed two-pass AE schemes and the single-pass one are described in Section 3 and Section 4, respectively. These sections also include the design rationale and security proofs of the constructions. Section 5 is devoted to the implementations of the proposed schemes. The efficiency and performance results are also presented in this section. Finally, Section 6 concludes the paper. Some omitted proofs are presented in Appendix A.

2 Preliminaries

2.1 Notations

$\mathsf{msb}_\ell(x)$ is the string consisting of ℓ most significant bits of x. $\mathsf{pad}_n(x)$ is the n-bit string obtained by concatenating a 1 and optionally a few 0's to the right of string x. The notation $\xleftarrow{n}$ is the operator which splits the right-hand string into a list of n-bit strings, except the last one which may be shorter. The concatenation of two bit strings a and b is denoted by $a\|b$. By $x \xleftarrow{\$} \mathcal{S}$, we mean a uniform sampling of x from the finite set $\mathcal{S}$. $\mathrm{Func}(n, \ell)$ is the set of all functions from n-bit to ℓ-bit strings. A family of functions $F = \{F_K\}$ is a set of functions indexed by the key K. $\mathcal{A}^{\mathcal{O}_K(\cdot,\cdot)}$ denotes

a machine $\mathcal{A}$ with oracle access to $\mathcal{O}_K$. The oracle $\mathcal{O}_K(\cdot,\cdot)$ has two inputs under the control of $\mathcal{A}$, however, she cannot access the parameter K embedded inside.

Polynomial additions and multiplications in $\mathrm{GF}(2^{128})$ and $\mathrm{GF}(2^{127})$ are performed modulo $x^{128}+x^{127}+x^{126}+x^{121}+1$ and $x^{127}+x^{126}+1$, respectively, which are typical irreducible polynomials. The elements of $\mathrm{GF}(2^n)$ may be represented in three ways and be interchanged frequently. An n-bit string $Y = y_{n-1}...y_1y_0$ is equivalent to the polynomial $y_{n-1}x^n - 1 + ... + y_1x + y_0$ in $\mathrm{GF}(2^n)$, and is also equivalent to the number obtained by base-2 interpretation of Y. For instance, $2Y$, $3Y$, and $4Y$ are the multiplication of polynomials x, $x+1$, and x^2, respectively, with the polynomial equivalent to the string Y. Multiplication of a small constant with a field element can be performed very efficiently using a few shift and XOR operations.

The success probability or advantage of an adversary $\mathcal{A}$ is denoted by $\mathbf{Adv}^{\mathsf{G}}_{S[\mathcal{F}]}(\mathcal{A})$, where S is the cryptographic scheme being attacked by $\mathcal{A}$, $\mathcal{F}$ is the (family of the) building block used in S, and G is the security model or security game in which $\mathcal{A}$ is modeled. $\$(\cdot)$ and $\$(\cdot,\cdot)$ are oracles that, on each query, return a fresh random bit string of specified length.

2.2 Pseudorandom Functions and Message Authentication Codes

For the integers $n \geq \ell \geq 1$, a family of functions $F = \left\{F_K : \{0,1\}^n \to \{0,1\}^\ell\right\}$ is a pseudorandom function (PRF) if for any adversary $\mathcal{A}$, with ordinary number of queries and resources, the following advantage is small:

$$\mathbf{Adv}^{\mathsf{prf}}_{F}(\mathcal{A}) = \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{F_K(.)} \Rightarrow 1] - \Pr[\rho \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{\rho(.)} \Rightarrow 1]. \quad (1)$$

A pseudorandom permutation (PRP) is a kind of PRF containing one-to-one and onto functions with the same input and output length. A variable-input-length pseudorandom function (VILPRF) family $F = \left\{F_K : \{0,1\}^* \to \{0,1\}^\ell\right\}$ is an special PRF in which the input can be a string of arbitrary length.

A message authentication code (MAC) $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ consists of a randomized key generation algorithm $\mathcal{K}$, and two functions $\mathcal{T}_K(N,M) : \{0,1\}^w \times \{0,1\}^* \to \{0,1\}^\tau$ and $\mathcal{V}_K(N,M,T) : \{0,1\}^w \times \{0,1\}^* \times \{0,1\}^\tau \to \{\mathtt{true},\mathtt{false}\}$ for message tagging and tag verification, respectively. Unforgeability under chosen message attack (uf-cma) is the model to define the security of $\mathcal{MA}$. This scheme is a secure MAC if for any adversary $\mathcal{A}$, with ordinary number of queries and resources, the following advantage is small:

$$\mathbf{Adv}^{\mathsf{uf\text{-}cma}}_{\mathcal{MA}}(\mathcal{A}) = \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{T}_K(.,.),\mathcal{V}_K(.,.,.)} \text{ forges}]. \quad (2)$$

A forgery is successful when $\mathcal{A}$ makes a query to $\mathcal{V}_K$ with two conditions. Firstly, the inputs N, M, and T given to $\mathcal{V}_K$ have not been appeared before in any query-response pair of $\mathcal{T}_K$. Secondly, $\mathcal{V}_K$ does not return $\mathtt{false}$ on the query.

2.3 Authenticated Encryption

An authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a tuple of three algorithms. $\mathcal{K}$ is the key generation algorithm. The encryption function $\mathcal{E}(K,N,M) : \{0,1\}^k \times \{0,1\}^\omega \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^\tau$ receives the key, nonce, and message, and returns the ciphertext and tag (C,T). The decryption function $\mathcal{D}(K,N,C,T) : \{0,1\}^k \times \{0,1\}^\omega \times \{0,1\}^* \times \{0,1\}^\tau \to \{0,1\}^* \cup \{\perp\}$ performs the inverse functionality, and returns the plaintext. $\mathcal{D}$ may also returns $\perp$ in the case of an error. $\mathcal{E}_K(N,M)$ and $\mathcal{D}_K(N,C,T)$ are the same functions, considering the key as a parameter. The AEs referred in this paper are nonce-based. Therefore, an extra parameter N is given to $\mathcal{E}$ and $\mathcal{D}$ as the nonce.

$\mathcal{AE}$ should have two security properties, privacy and authenticity. Formally, a secure $\mathcal{AE}$ should provide the indistinguishability under chosen plaintext attack (ind-cpa) and maintain the integrity of the ciphertext (int-ctxt). No nonce value N should be repeated in the inputs of the encryption function $\mathcal{E}$. Otherwise, the privacy or authenticity may not be guaranteed. Note that a repeated nonce in the inputs of $\mathcal{D}$, or between the inputs of $\mathcal{E}$ and $\mathcal{D}$, is fine. Without loss of generality, we can assume that all adversaries are nonce-respecting, i.e., they never send same nonce value to $\mathcal{E}$. The privacy advantage of an adversary $\mathcal{A}$ is defined as follows.

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{AE}}(\mathcal{A}) = \Pr[K \leftarrow \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)} = 1] - \Pr[\mathcal{A}^{\$(\cdot,\cdot)} = 1]. \quad (3)$$

$\$(\cdot,\cdot)$ is an oracle which returns a fresh random string of length $|\mathcal{E}_K(\cdot,\cdot)|$. Note that $\$(\cdot,\cdot)$ is not queried twice on an exact input, as $\mathcal{A}$ is nonce-respecting.

The authenticity advantage of an adversary $\mathcal{A}$ is defined as follows.

$$\mathbf{Adv}^{\mathsf{int\text{-}ctxt}}_{\mathcal{AE}}(\mathcal{A}) = \Pr[K \leftarrow \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot),\mathcal{D}_K(\cdot,\cdot,\cdot)} \text{ forges}]. \quad (4)$$

A forgery is successful when $\mathcal{A}$ makes a query to $\mathcal{D}_K$ with two conditions. Firstly, the inputs N, C, and T given to $\mathcal{D}_K$ have not been appeared before in any query-response pair of $\mathcal{E}_K$. Secondly, $\mathcal{D}_K$ does not return $\perp$ on the query. If both $\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{AE}}(\mathcal{A})$ and $\mathbf{Adv}^{\mathsf{int\text{-}ctxt}}_{\mathcal{AE}}(\mathcal{A})$ are small respecting any adversary $\mathcal{A}$, with ordinary number of queries and resources, then $\mathcal{AE}$ is a secure authenticated encryption scheme.

3 Two-pass Lattice-based Authenticated Encryption

3.1 Using a CBC-MAC Variant

The first proposed authenticated encryption scheme, referred to as LAE1, is illustrated in Fig. 1. It is obtained by an encrypt-and-MAC (E&M) composition of a nonce-based symmetric encryption scheme and a message authentication code (MAC). The encryption part is built using SPRING in the CTR mode, in which the indices given to the SPRINGs has two segments. The first segment is the nonce N, which is fixed for all SPRINGs of the encryption part. The second segment is the block number encoded with a gray-code. The input and output of SPRING are n and ℓ bits, respectively. The length of the nonce is fixed to w bits. The SPRING functions of the encryption part use K_1 as the key. The authentication part is a variant of CBC-MAC [29]. The concatenation of the nonce and message are fed into a CBC-chain of SPRINGs. The SPRINGs in the authentication part use a different key K_2, in order to ensure a safe composition.

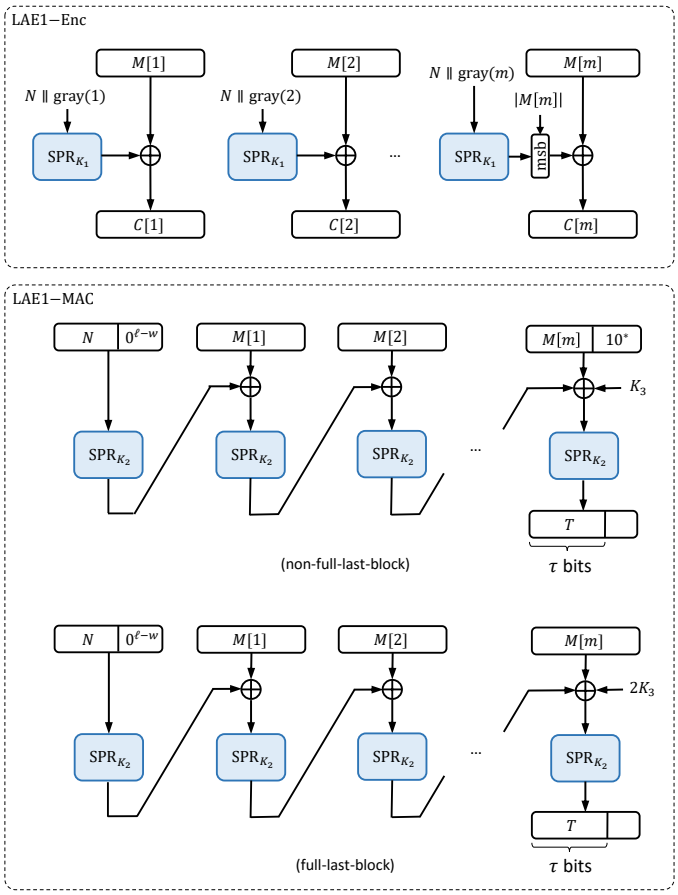

Fig. 1: The flow diagram of LAE1 authenticated encryption procedure; Top is the encryption part (LAE1-Enc), and bottom is the authentication part (LAE1-MAC), which is divided into full-last-block and non-full-last-block modes.

In the processing of the last block, another parameter is XORed with the input of SPRING, which depends on a third short key K_3. The multiplication of 2 by K_3, in the full-last-block case, is performed in $\mathrm{GF}(2^\ell)$ using at most one shift and one XOR. The encryption procedure of LAE1 is shown in Scheme 1. The decryption procedure is easily obtained by XORing the given ciphertext C with $X[i]$'s to obtain M^*, and computing the expected tag T^* from M^* in the forward direction. If T^* is the same as the given tag T, then the successfully decrypted message M^* is returned. Otherwise, only a $\perp$ is returned.

Design Rationale. The authentication part of LAE1 is a stand-alone MAC based on SPRING as the underlying primitive. It is similar to TMAC [19] with a few differences. TMAC uses a block cipher instead of a PRF. Also, TMAC does not have a parameter τ to adjust the tag length. The input and output lengths of the underlying function in TMAC (i.e., the block cipher) are the same, while the SPRING output is shorter than its input. There are some alternatives to be used as a base to construct a SPRING-based MAC. A structure similar to OMAC [20] may seem to be advantageous as it requires only one key for the authentication (LAE1 uses two keys K_2, K_3 for this purpose); however, it needs an extra invocation of SPRING which is relatively heavy. Moreover, the length of the extra key K_3 is

Scheme 1 (LAE1)

procedure AUTHENCRYPT(N, M)
  $(M[1], ..., M[m]) \xleftarrow{\ell} M$
  // LAE1-Enc
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $Z[i] \leftarrow \mathrm{SPR}_{K_1}(N \| \mathsf{gray}(i))$
    $C[i] \leftarrow M[i] \oplus Z[i]$
  $Z[m] \leftarrow \mathsf{msb}_{|M[m]|}\left[\mathrm{SPR}_{K_1}(N \| \mathsf{gray}(m))\right]$
  $C[m] \leftarrow M[m] \oplus Z[m]$
  $C \leftarrow C[1] \| ... \| C[m]$
  // LAE1-MAC
  $X[0] \leftarrow N \| 0^{\ell-w}$
  $Y[0] \leftarrow \mathrm{SPR}_{K_2}(X[0] \| 0^{n-\ell})$
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $X[i] \leftarrow M[i] \oplus Y[i-1]$
    $Y[i] \leftarrow \mathrm{SPR}_{K_2}(X[i] \| 0^{n-\ell})$
  if $|M[m]| = \ell$ **then**
    $X[m] \leftarrow M[m] \oplus X[m-1] \oplus 2K_3$
  else
    $X[m] \leftarrow \mathsf{pad}_\ell(M[m]) \oplus X[m-1] \oplus K_3$
  $Y[m] \leftarrow \mathrm{SPR}_{K_2}(X[m] \| 0^{n-\ell})$
  $T \leftarrow \mathsf{msb}_\tau(Y[m])$
  return (C, T)

n bits, which is much smaller than the size of the SPRING keys K_1,K_2 (see Section 5 for the efficiency results). Similar to [16], to achieve a better performance, the SPRING input in the encryption part is designed to be a gray-code counter, consisting of a fixed part N and the gray-code encoding of the block number.

A SPRING-based parallel MAC can be built using a structure similar to PMAC [30]. This makes the authenticated encryption scheme parallel (the encryption part is already parallel). However, the overall performance is not so different than LAE1, as the SPRING inputs are again non-gray-code. As mentioned before, the type of the composition used in LAE1 is encrypt-and-MAC (E&M). Instead of the encrypt-then-MAC (EtM) technique in which the ciphertext is given to the MAC algorithm, E&M allows parallel processing of the ciphertext and tag. To build an AE using the EtM generic composition, Bellare and Namprempre [31] claim that perfoming a MAC only on the ciphertext is sufficient. However, Namprempre et al. [32] show that their results cannot be applied directly to the nonce-based authenticated encryption. Thus, we cannot save one SPRING invocation by using the EtM technique.

Secure E&M generic compositions are proposed by Namprempre et al. [32]. LAE1 has some differences with the proposed constructions in [32]. Mainly, the authors of this paper insisted on the use of a single key, and derived subsequent keys via a PRF. However, LAE1 has three different keys for the sake of running time efficiency.

3.1.1 Security of LAE1

Theorem 1 and Theorem 2 show the security of LAE1.

Theorem 1 (Privacy of LAE1). Fix $n \geq \ell \geq 1$. Let $\mathcal{F} = \{F : \{0,1\}^n \to \{0,1\}^\ell\}$ be a family of functions. For any adversary $\mathcal{A}$ to attack the privacy of $LAE1[\mathcal{F}]$, who runs in time t and asks q oracle queries with a maximum length of m blocks for each query, there exists

an adversary $\mathcal{P}$ against the pseudorandomness of $\mathcal{F}$, and we have

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathit{LAE1[\mathcal{F}]}}(\mathcal{A}) \leq 2\mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}) + \frac{(4m^2+1)q^2}{2^{\ell+1}} + \frac{q^2}{2^{n+1}}. \tag{5}$$

Moreover, adversary P asks $q' = 2\sigma + q$ oracle queries, and runs in time $t' = 2t + \sigma t_F + \alpha n(\sigma + q)$, where t_F is the time to compute F, and α is a constant depending on the model of the computation.

To prove Theorem 1, the following lemmas are required.

Lemma 1 (Privacy of LAE1-Enc). Fix $n \geq \ell \geq 1$. Let $\mathcal{F} = \{F : \{0,1\}^n \to \{0,1\}^\ell\}$ be a family of functions. For any adversary $\mathcal{B}$ to attack the privacy of *LAE1-Enc*$[\mathcal{F}]$, who asks q queries with total message length of σ blocks, there exists an adversary $\mathcal{P}_1$ against the pseudorandomness of $\mathcal{F}$, and we have

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathit{LAE1\text{-}Enc[\mathcal{F}]}}(\mathcal{B}) \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}_1). \tag{6}$$

Moreover, adversary $\mathcal{P}_1$ asks $q' = \sigma q$ oracle queries, and runs in time $t' = t + \alpha n(\sigma + q)$, where α is a constant depending on the model of the computation.

The proof of Lemma 1 is relevant to the security proof of [33, Theorem 13].

Lemma 2 (Ideal LAE1-MAC is pseudorandom). Let $\mathcal{R} = \mathrm{Func}(n, \ell)$. For any adversary $\mathcal{A}$ asking q queries, each of which at most m blocks, we have

$$\mathbf{Adv}^{\mathsf{vilprf}}_{\mathit{LAE1\text{-}MAC[\mathcal{R}]}}(\mathcal{A}) \leq \frac{(4m^2+1)q^2}{2^{\ell+1}} + \frac{q^2}{2^{n+1}}. \tag{7}$$

Note that this result is independent of the computational resources of $\mathcal{A}$.

Proof of Lemma 2. Suppose $\mathcal{T}$ is the MAC part of the LAE1 scheme. We introduce *fFCBC* as a variant of *FCBC* [18], which uses pseudorandom functions. *fFCBC*$[F_1, F_2, F_3]$ is a CBC-MAC which calls F_1 on the intermediate blocks and calls either F_2 or F_3 on the last block, depending on being a full or partial block. By the definition of the VILPRF advantage, we have

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{vilprf}}_{\mathit{LAE1\text{-}MAC[\mathcal{R}]}}(\mathcal{A}) = \\
&\Pr[\rho \xleftarrow{\$} \mathrm{Func}(n,\ell), K_3 \xleftarrow{\$} \{0,1\}^n : \mathcal{A}^{\mathcal{T}[\rho,K_3](\cdot)} = 1] - \\
&\quad \Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{\mathit{fFCBC}[\rho_1,\rho_2,\rho_3](\cdot)} = 1] + \\
&\Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{\mathit{fFCBC}[\rho_1,\rho_2,\rho_3](\cdot)} = 1] - \\
&\quad \Pr[\rho \xleftarrow{\$} \mathrm{Func}(*,\ell) : \mathcal{A}^{\rho(\cdot)} = 1].
\end{aligned} \tag{8}$$

We now consider each of these two pairs of probabilities. The second pair is the information-theoretic security of *fFCBC*, which is analyzed in Appendix A.1 and bounded as follows:

$$\mathbf{Adv}^{\mathit{vilprf}}_{\mathit{fFCBC}[\rho_1,\rho_2,\rho_3]}(\mathcal{A}) \leq \frac{(4m^2+1)q^2}{2^{\ell+1}}. \tag{9}$$

Note that $\mathcal{T}[\rho, K_3](\cdot)$ is equivalent to *fFCBC*$[\rho(\cdot), \rho(\cdot \oplus K_3), \rho(\cdot \oplus 2K_3)]$. Thus, the first pair of probabilities can be written as

$$\begin{aligned}
&\Pr[\rho \xleftarrow{\$} \mathrm{Func}(n,\ell), K_3 \xleftarrow{\$} \{0,1\}^n : \mathcal{B}^{\rho(\cdot),\rho(\cdot\oplus K_3),}_{\rho(\cdot\oplus 2K_3)} = 1] - \\
&\Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{B}^{\rho_1(\cdot),\rho_2(\cdot),\rho_3(\cdot)} = 1],
\end{aligned} \tag{10}$$

where $\mathcal{B}$ simply simulates $\mathcal{A}$ to distinguish the two sets of oracles. Computing the advantage of such adversary is straightforward and it is bounded by $q^2/2^{n+1}$. □

Lemma 3 (LAE1-MAC is a VILPRF). Fix $n \geq \ell \geq 1$. Let $\mathcal{F} = \{F : \{0,1\}^n \to \{0,1\}^\ell\}$ be a family of functions. For any adversary $\mathcal{D}$ to attack the pseudorandomness of *LAE1-MAC*$[\mathcal{F}]$, there exists an adversary $\mathcal{P}_2$ against pseudorandomness of $\mathcal{F}$, and we have

$$\mathbf{Adv}^{\mathsf{vilprf}}_{\mathit{LAE1\text{-}MAC[\mathcal{F}]}}(\mathcal{D}) \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}_2) + \frac{(4m^2+1)q^2}{2^{\ell+1}} + \frac{q^2}{2^{n+1}}. \tag{11}$$

Moreover, adversary $\mathcal{P}_2$ asks $q' = (\sigma+q)q$ oracle queries, and runs in time $t' = t + \alpha n(\sigma + q)$, where α is a constant depending on the model of the computation.

Lemma 3 is the complexity-theoretic counterpart of Lemma 2, which can be proven in a standard way (for example, see [29, Section 3.2]).

Proof of Theorem 1. Suppose $\mathcal{E}$ and $\mathcal{T}$ are the Enc and MAC parts of LAE1, respectively. We derive two adversaries $\mathcal{B}$ and $\mathcal{D}$ from $\mathcal{A}$. Both simulate an ind-cpa game for $\mathcal{A}$. Meanwhile, $\mathcal{B}$ tries to break the privacy of LAE1-Enc and $\mathcal{D}$ attempts to distinguish LAE1-MAC from a random function. $\mathcal{B}$ forwards the $\mathcal{A}$'s request to its own oracle to obtain C. Then, generates a fresh random T and outputs (C, T). Furthermore, $\mathcal{D}$ generates a random key K_1. Then, $\mathcal{D}$ itself computes C and invokes its oracle to obtain T. Thus, we have

$$\begin{aligned}
&\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathit{LAE1[\mathcal{F}]}}(\mathcal{A}) = \\
&\quad = \Pr[\mathcal{A}^{\mathcal{E}(\cdot,\cdot),\mathcal{T}(\cdot,\cdot)} \Rightarrow 1] - \Pr[\mathcal{A}^{\$(\cdot,\cdot),\mathcal{T}(\cdot,\cdot)} \Rightarrow 1] + \\
&\qquad \Pr[\mathcal{A}^{\$(\cdot,\cdot),\mathcal{T}(\cdot,\cdot)} \Rightarrow 1] - \Pr[\mathcal{A}^{\$(\cdot,\cdot),\$(\cdot,\cdot)} \Rightarrow 1] \\
&\quad = \mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathit{LAE1\text{-}Enc[\mathcal{F}]}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{vilprf}}_{\mathit{LAE1\text{-}MAC[\mathcal{F}]}}(\mathcal{D}) \\
&\quad \leq 2\mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}) + \frac{(4m^2+1)q^2}{2^{\ell+1}} + \frac{q^2}{2^{n+1}},
\end{aligned} \tag{12}$$

where $\mathcal{P}$ can be either $\mathcal{P}_1$ or $\mathcal{P}_2$ whichever has more advantage. Additionally, the last inequality is provided by Lemma 1 and Lemma 3. □

Theorem 2 (Authenticity of LAE1). Fix $n \geq \ell \geq 1$. Let $\mathcal{F} = \{F : \{0,1\}^n \to \{0,1\}^\ell\}$ be a family of functions. For any adversary $\mathcal{A}$ to attack the authenticity of *LAE1*$[\mathcal{F}]$, who runs in time t and asks q_e encryption and q_d decryption queries, there exists an adversary $\mathcal{P}$ against the pseudorandomness of $\mathcal{F}$, and we have

$$\mathbf{Adv}^{\mathsf{int\text{-}ctxt}}_{\mathit{LAE1[\mathcal{F}]}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}) + \frac{(4m^2+1)q^2+2}{2^{\ell+1}} + \frac{q^2}{2^{n+1}} + \frac{q_d}{2^\tau}, \tag{13}$$

where $q = q_e + q_d$. Moreover, adversary $\mathcal{P}$ asks $q' = \sigma + q$ oracle queries, and runs in time $t' = t + \sigma t_F + \alpha n(\sigma + q)$,

Scheme 2 (LAE2)

procedure AUTHENCRYPT(N, M)
  $(M[1], ..., M[m]) \xleftarrow{\ell} M$
  // LAE2-Enc
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $Z[i] \leftarrow \mathrm{SPR}_{K_1}(N \| \mathsf{gray}(i))$
    $C[i] \leftarrow M[i] \oplus Z[i]$
  $Z[m] \leftarrow \mathsf{msb}_{|M[m]|}\left[\mathrm{SPR}_{K_1}(N \| \mathsf{gray}(m))\right]$
  $C[m] \leftarrow M[m] \oplus Z[m]$
  $C \leftarrow C[1] \| ... \| C[m]$
  // LAE2-MAC
  $Y[0] \leftarrow 0^n$
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $Y[i] \leftarrow \left[Y[i-1] \oplus (C[i] \| 0^{n-\ell})\right] \otimes K_2$
  $Y[m] \leftarrow \left[Y[m-1] \oplus (C[m] \| 0^{n-|C[m]|})\right] \otimes K_2$
  $Y[m+1] \leftarrow (Y[m] \oplus \mathsf{len}(C)) \otimes K_2$
  $Y[m+2] \leftarrow \mathsf{msb}_\ell(Y[m+1]) \oplus \mathrm{SPR}_{K_1}(N \| \mathsf{gray}(0))$
  $T \leftarrow \mathsf{msb}_\tau(Y[m+2])$
  return (C, T)

where t_F is the time to compute F, and α is a constant depending on the model of the computation.

The proof of Theorem 2 is presented in Appendix A.2.

3.2 Using a Carter-Wegman MAC

LAE1 calls SPRING twice per message block. Moreover, the performance results in Section 5 show that computing SPRING is the most time consuming part of the execution. The idea to design a more efficient scheme is to reduce the number of SPRING invocations in the authentication part. We have used the technique of Wegman and Carter [34] to build a MAC using a universal class of hash functions and the lattice-based PRF SPRING. In order to apply this technique, the ciphertext is given to a secret function from an XOR-universal class of hash functions, and the output is masked with the PRF output on a fresh and unique input.

Scheme 2 and Fig. 2 show the second proposed lattice-based AE, referred to as LAE2, using the Carter-Wegman method [34]. Multiplications in this scheme are performed in $\mathrm{GF}(2^n)$. The encryption part is the same as before in LAE1. The universal class of hash functions utilized in LAE2 is defined as follows:

$$H_a(C) = C[1]a^{m+1} + C[2]a^m + ... + C[m]a^2 + \mathsf{len}(C)a.$$

Here, $m = |C|_\ell$ and the polynomial $a \in \mathrm{GF}(2^n)$ is the function index. All the operations are performed in $\mathrm{GF}(2^n)$.

Using an XOR-universal hash function built by iterative multiplications of a secret polynomial $a \in \mathrm{GF}(2^n)$ is similar to the NIST-recommended GCM mode of operation [26]. However, GCM encrypts a constant block (zero block) to derive the second key for authentication, while LAE2 uses another key K_2 to save one SPRING invocation (see the more detailed discussion in Section 3.1 on a similar case). Moreover, there are many pad and msb functions involved because of asymmetry between the input and output length of SPRING, which should be handled carefully in the security proof.

Design Rationale. Although LAE2 is considered as a two-pass authenticated encryption scheme, the multiplications in $\mathrm{GF}(2^n)$ can be computed very efficiently. Thus, the authentication part is much more efficient than LAE1. Typical high-end processors have instruction set extensions to perform this operation in a few clock cycles (see Section 5 for more details on the ones used in our benchmarks). Moreover, when an old or constrained processor is required, or in the case of hardware implementations, this multiplication can also be performed efficiently using a moderate-size precomputed lookup table. Nevertheless, both LAE1 and LAE2 are not fully parallel, and their authentication part should be computed sequentially.

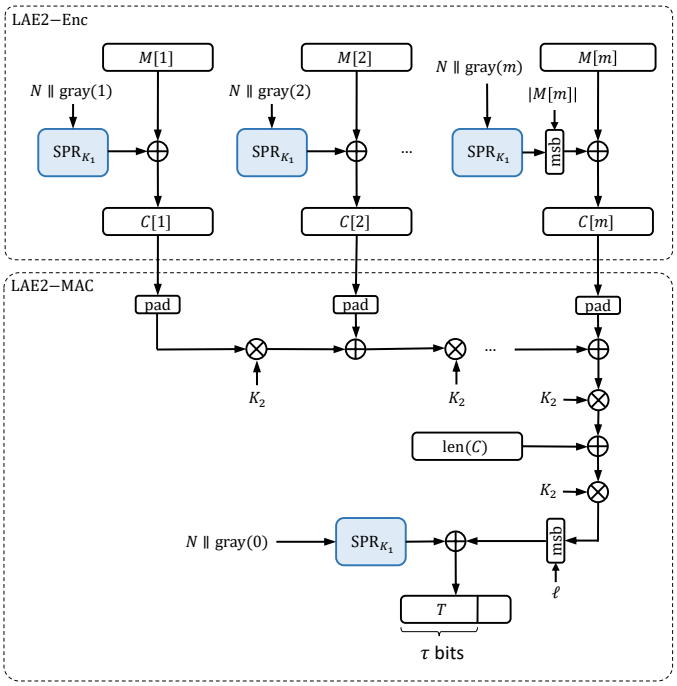

Fig. 2: The flow diagram of LAE2 authenticated encryption procedure. Top is the encryption part (LAE2-Enc), and bottom is the authentication part (LAE2-MAC).

It is worth mentioning that replacing the majority of SPRING functions with polynomial multiplications in $\mathrm{GF}(2^n)$ does not introduce any new security assumption in LAE2. The hardness of lattice problems is still the only supporting assumption. Thus, many SPRING invocations in the authentication part of LAE1 are reduced to only one in LAE2, without any loss in the security level.

3.2.1 Security of LAE2

Theorem 3 and Theorem 4 show the security of LAE2.

Theorem 3 (Privacy of LAE2). Fix $n \geq \ell \geq 1$. Let $\mathcal{F} = \{F : \{0,1\}^n \rightarrow \{0,1\}^\ell\}$ be a family of functions. For any adversary $\mathcal{A}$ to attack the privacy of $LAE2[\mathcal{F}]$, who runs in time t and asks q oracle queries, each of which has at most $m < 2^{n-w}$ blocks, there exists an adversary $\mathcal{P}$ against the pseudorandomness of $\mathcal{F}$, and we have

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{LAE2[\mathcal{F}]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}). \tag{14}$$

Moreover, adversary P asks $q' = \sigma + q$ oracle queries, and runs in time $t' = t + (\sigma+q)t_\otimes + \alpha n(\sigma+q)$, where $t_\otimes$ is the time to compute a $\mathrm{GF}(2^n)$ multiplication, and α is a constant depending on the model of the computation.

Game 1 (G_0 for LAE2)

procedure AUTHENCRYPT(N, M)
  $(M[1], ..., M[m]) \xleftarrow{\ell} M$
  for $i \leftarrow 1$ **to** $m-1$ **do**
    if $N \| \mathsf{gray}(i) \notin Dom(F)$ **then**
      $F\big(N \| \mathsf{gray}(i)\big) \xleftarrow{\$} \{0,1\}^{\ell}$
    $Z[i] \leftarrow F\big(N \| \mathsf{gray}(i)\big)$
    $C[i] \leftarrow M[i] \oplus Z[i]$
  if $N \| \mathsf{gray}(m) \notin Dom(F)$ **then**
    $F\big(N \| \mathsf{gray}(m)\big) \xleftarrow{\$} \{0,1\}^{\ell}$
  $Z[m] \leftarrow \mathsf{msb}_{|M[m]|}\Big[F\big(N \| \mathsf{gray}(m)\big)\Big]$
  $C[m] \leftarrow M[m] \oplus Z[m]$
  $C \leftarrow C[1] \| ... \| C[m]$
  $Y[0] \leftarrow 0^n$
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $Y[i] \leftarrow \Big[Y[i-1] \oplus \big(C[i] \| 0^{n-\ell}\big)\Big] \otimes K_2$
  $Y[m] \leftarrow \Big[Y[m-1] \oplus \big(C[m] \| 0^{n-|C[m]|}\big)\Big] \otimes K_2$
  $Y[m+1] \leftarrow \big(Y[m] \oplus \mathsf{len}(C)\big) \otimes K_2$
  if $N \| \mathsf{gray}(0) \notin Dom(F)$ **then**
    $F\big(N \| \mathsf{gray}(0)\big) \xleftarrow{\$} \{0,1\}^{\ell}$
  $Y[m+2] \leftarrow \mathsf{msb}_{\ell}\big(Y[m+1]\big) \oplus F\big(N \| \mathsf{gray}(0)\big)$
  $T \leftarrow \mathsf{msb}_{\tau}\big(Y[m+2]\big)$
  return (C, T)

procedure AUTHDECRYPT(N, C, T)
  $(C_1, ..., C_m) \xleftarrow{\ell} C$
  $Y[0] \leftarrow 0^n$
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $Y[i] \leftarrow \Big[Y[i-1] \oplus \big(C[i] \| 0^{n-\ell}\big)\Big] \otimes K_2$
  $Y[m] \leftarrow \Big[Y[m-1] \oplus \big(C[m] \| 0^{n-|C[m]|}\big)\Big] \otimes K_2$
  $Y[m+1] \leftarrow \big(Y[m] \oplus \mathsf{len}(C)\big) \otimes K_2$
  if $N \| \mathsf{gray}(0) \notin Dom(F)$ **then**
    $F\big(N \| \mathsf{gray}(0)\big) \xleftarrow{\$} \{0,1\}^{\ell}$
  $Y[m+2] \leftarrow \mathsf{msb}_{\ell}\big(Y[m+1]\big) \oplus F\big(N \| \mathsf{gray}(0)\big)$
  $T^* \leftarrow \mathsf{msb}_{\tau}\big(Y[m+2]\big)$
  if $T = T^*$ **then**
    wins $\leftarrow$ `true`; **return** `true`
  return $\perp$

Game 2 (G_1,G_2 for LAE2)

procedure AUTHENCRYPT(N, M)
  $(M[1], ..., M[m]) \xleftarrow{\ell} M$
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $Z[i] \xleftarrow{\$} \{0,1\}^{\ell}$
    if $N \| \mathsf{gray}(i) \in Dom(F)$ **then**
      $\mathrm{bad}_1 \leftarrow$ `true`; $\underline{Z[i] \leftarrow F\big(N \| \mathsf{gray}(i)\big)}$
    $F\big(N \| \mathsf{gray}(i)\big) \leftarrow Z[i]$
    $C[i] \leftarrow M[i] \oplus Z[i]$
  $Z[m] \xleftarrow{\$} \{0,1\}^{\ell}$
  if $N \| \mathsf{gray}(m) \in Dom(F)$ **then**
    $\mathrm{bad}_1 \leftarrow$ `true`; $\underline{Z[m] \leftarrow F\big(N \| \mathsf{gray}(m)\big)}$
  $F\big(N \| \mathsf{gray}(m)\big) \leftarrow Z[m]$
  $Z[m] \leftarrow \mathsf{msb}_{|M[m]|}\big[Z[m]\big]$
  $C[m] \leftarrow M[m] \oplus Z[m]$
  $C \leftarrow C[1] \| ... \| C[m]$
  $Y[0] \leftarrow 0^n$
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $Y[i] \leftarrow \Big[Y[i-1] \oplus \big(C[i] \| 0^{n-\ell}\big)\Big] \otimes K_2$
  $Y[m] \leftarrow \Big[Y[m-1] \oplus \big(C[m] \| 0^{n-|C[m]|}\big)\Big] \otimes K_2$
  $Y[m+1] \leftarrow \big(Y[m] \oplus \mathsf{len}(C)\big) \otimes K_2$
  if $N \| \mathsf{gray}(0) \notin Dom(F)$ **then**
    $F\big(N \| \mathsf{gray}(0)\big) \xleftarrow{\$} \{0,1\}^{\ell}$
  $Y[m+2] \leftarrow \mathsf{msb}_{\ell}\big(Y[m+1]\big) \oplus F\big(N \| \mathsf{gray}(0)\big)$
  $T \leftarrow \mathsf{msb}_{\tau}\big(Y[m+2]\big)$
  return (C, T)

procedure AUTHDECRYPT(N, C, T)
  $(C_1, ..., C_m) \xleftarrow{\ell} C$
  $Y[0] \leftarrow 0^n$
  for $i \leftarrow 1$ **to** $m-1$ **do**
    $Y[i] \leftarrow \Big[Y[i-1] \oplus \big(C[i] \| 0^{n-\ell}\big)\Big] \otimes K_2$
  $Y[m] \leftarrow \Big[Y[m-1] \oplus \big(C[m] \| 0^{n-|C[m]|}\big)\Big] \otimes K_2$
  $Y[m+1] \leftarrow \big(Y[m] \oplus \mathsf{len}(C)\big) \otimes K_2$
  if $N \| \mathsf{gray}(0) \notin Dom(F)$ **then**
    $F\big(N \| \mathsf{gray}(0)\big) \xleftarrow{\$} \{0,1\}^{\ell}$
  $Y[m+2] \leftarrow \mathsf{msb}_{\ell}\big(Y[m+1]\big) \oplus F\big(N \| \mathsf{gray}(0)\big)$
  $T^* \leftarrow \mathsf{msb}_{\tau}\big(Y[m+2]\big)$
  if $T = T^*$ **then**
    $\mathrm{bad}_2 \leftarrow$ `true`; $\underline{\mathrm{wins} \leftarrow \texttt{true}}$; $\underline{\textbf{return } \texttt{true}}$
  return $\perp$

The proof sketch of Theorem 3 is as follows. The problem in the information-theoretic setting, i.e., when $F \xleftarrow{\$} \mathrm{Func}(n, \ell)$, is simple. Both ciphertext C and tag T are XORed with the output of F. In addition, a nonce-respecting adversary $\mathcal{A}$ always queries with a fresh nonce N. Thus, the distribution of (C, T) is perfectly uniform. Transition from information-theoretic to complexity theoretic setting is also standard. □

Theorem 4 (Authenticity of LAE2). Fix $n \geq \ell \geq 1$. Let $\mathcal{F} = \{F : \{0,1\}^n \to \{0,1\}^{\ell}\}$ be a family of functions. For any adversary $\mathcal{A}$ to attack the authenticity of *LAE2*$[\mathcal{F}]$, who runs in time t and asks q_e encryption and q_d decryption queries, each of which with a total length of σ_e and σ_d blocks, there exists an adversary $\mathcal{P}$ against the pseudo-randomness of $\mathcal{F}$, and we have

$$\mathbf{Adv}^{\mathsf{int\text{-}ctxt}}_{LAE2[\mathcal{F}]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}) + \frac{q}{2^{\ell} - q2^{\ell-\tau}} + \frac{q_d}{2^{\tau}}, \quad (15)$$

where $q = q_e + q_d$. Moreover, adversary $\mathcal{P}$ asks $q' = \sigma_e q_e + q_d$ oracle queries and runs in time $t' = t + (\sigma + q)t_{\otimes} + \alpha n(\sigma + q)$, where $\sigma = \sigma_e + \sigma_d$, $t_{\otimes}$ is the time to compute a $\mathrm{GF}(2^n)$ multiplication, and α is a constant depending on the model of the computation.

Theorem 4 can be proven in a standard way from Lemma 4.

Lemma 4 (Authenticity of Ideal LAE2). Fix $n \geq \ell \geq 1$. Let $\mathcal{R} = \mathrm{Func}(n, \ell)$. For any adversary $\mathcal{A}$ asking q_e encryption and q_d decryption queries, we have

$$\mathbf{Adv}^{\mathsf{int\text{-}ctxt}}_{LAE2[\mathcal{R}]}(\mathcal{A}) \leq \frac{q}{2^{\ell} - q2^{\ell-\tau}} + \frac{q_d}{2^{\tau}}, \quad (16)$$

where $q = q_e + q_d$.

Proof of Lemma 4. We use the technique of sequences of games [35] to prove this result. Game 1 (G_0) simulates the environment of Lemma 4 for the adversary $\mathcal{A}$. There is an exception that the AUTHDECRYPT procedure in G_0 returns `true` instead of the message in the case that the forgery is

successful. However, this difference can be ignored because the following equation always holds.

$$\mathbf{Adv}^{\text{int-ctxt}}_{LAE2[\mathcal{R}]}(\mathcal{A}) = \Pr[G_0^{\mathcal{A}} \Rightarrow \text{wins}]. \quad (17)$$

Note that, without loss of generality, we can assume that the adversary $\mathcal{A}$ is deterministic. Therefore, the probability is taken only over the random function sampled from $\mathcal{R}$.

G_0 models the random function ρ using the lazy sampling technique. The games G_1 and G_2 are specified in Game 2. The underlined statements exist only in game G_1 and are removed in G_2. From the $\mathcal{A}$'s point of view, G_0 and G_1 are identical. Thus, the winning probability of $\mathcal{A}$ remains unchanged from G_0 to G_1. On the other hand, G_1 and G_2 are identical until bad. That is, as long as no bad flags (bad_1 or bad_2) are set to true, these two games are identical for $\mathcal{A}$. Thus, according to the fundamental lemma of game playing [36], we have

$$\Pr[G_1^{\mathcal{A}} \Rightarrow \text{wins}] - \Pr[G_2^{\mathcal{A}} \Rightarrow \text{wins}] \leq \Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_1 \text{ or } \text{bad}_2]. \quad (18)$$

However, as the AUTHDECRYPT procedure of G_2 always returns $\perp$, we have

$$\Pr[G_2^{\mathcal{A}} \Rightarrow \text{wins}] = 0. \quad (19)$$

Thus, we can conclude

$$\begin{aligned}\mathbf{Adv}^{\text{int-ctxt}}_{LAE2[\mathcal{R}]}(\mathcal{A}) &\leq \Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_1 \text{ or } \text{bad}_2] \\ &\leq \Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_1] + \Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_2].\end{aligned} \quad (20)$$

Note that bad_1 never occurs because $\mathcal{A}$ is nonce-respecting and never provides repeated nonces to the AUTHENCRYPT oracle. This is true even if the nonce value N, given to AUTHENCRYPT, has been repeatedly used in previous AUTHDECRYPT queries. We define $\Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_2 \text{ on } q_i]$ as the probability that bad_2 is set to true on the i-th query for the first time. As these probabilities are disjoint, we have

$$\Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_2] = \sum_{i=1}^{q} \Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_2 \text{ on } q_i]. \quad (21)$$

Now, we bound $\Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_2 \text{ on } q_i]$.

Let N_i, C_i, T_i, and T_i^* be the values appeared in the AUTHENCRYPT or AUTHDECRYPT procedure of the i-th query in G_2 (not all of them may be defined for each i). The probability of setting bad_2 is zero for the encryption queries because $\mathcal{A}$ is nonce-respecting. For the decryption queries, we have

$$\Pr[G_2^{\mathcal{A}} \text{ sets } \text{bad}_2 \text{ on } q_i] = \Pr[T_i = T_i^*]. \quad (22)$$

We claim that the event $T_i = T_i^*$ is independent of all the previous queries q_j ($j < i$) with $N_j \neq N_i$. If q_j is an encryption query, the returned C_j is uniformly random and independent, and T_j is deterministically calculated from C_j. If q_j is a decryption query, the returned $\perp$ only reveals that $T_j \neq T_j^*$, where T_j^* is independent of the event $T_i = T_i^*$.

Now, we assume that the adversary $\mathcal{A}$ uses the same nonce N for all the queries prior and include q_i, to maximize the probability of setting bad_2 on q_i. There are two cases in this setting:

Case 1: All the previous queries are AUTHDECRYPT, and are responded by a $\perp$. In this case, for $j < i$, it is revealed to $\mathcal{A}$ that $T_j \neq T_j^* = \mathsf{msb}_\tau\Big(H_{K_2}(C_j)\Big) \oplus N^*$, where $N^* = \mathsf{msb}_\tau\Big(\rho\left(N \| \,\mathsf{gray}(0)\right)\Big)$ and is common between all the queries. Thus, at most $2^{\ell-\tau}$ choices of K_2 become invalid after each query. On the other hand, only one of the choices makes $T_i = T_i^*$. Therefore, the following equation holds.

$$\Pr[T_i = T_i^*] = \frac{1}{2^\ell - (i-1)2^{\ell-\tau}}. \quad (23)$$

Case 2: Exactly one of the previous queries is AUTHENCRYPT. Assume that q_k ($1 \leq k < i$) is that encryption query. The other queries are all AUTHDECRYPT and are responded by a $\perp$. Thus, $\mathcal{A}$ has a tuple (M_k, C_k, T_k) for which $T_k = \mathsf{msb}_\tau\Big(H_{K_2}(C_k)\Big) \oplus N^*$ holds. Due to the fact that N^* is an independent and uniformly-random value (random function F evaluated on fresh nonce N), T_k has also this property. Hence, the pair (C_k, T_k) causes at most $2^{\ell-\tau}$ choices of K_2 become invalid, and similar to Case 1 equation 23 holds.

Moreover, another strategy is to perform exhaustive search on the correct tag with an advantage of $q_d/2^\tau$. Finally, we can conclude as follows.

$$\begin{aligned}\mathbf{Adv}^{\text{int-ctxt}}_{LAE2[\mathcal{R}]}(\mathcal{A}) &\leq \sum_{i=1}^{q} \frac{1}{2^\ell - (i-1)2^{\ell-\tau}} + \frac{q_d}{2^\tau} \\ &\leq \frac{q}{2^\ell - q2^{\ell-\tau}} + \frac{q_d}{2^\tau}. \ \square\end{aligned} \quad (24)$$

4 SINGLE-PASS LATTICE-BASED AUTHENTICATED ENCRYPTION

LAE1 and LAE2 perform the encryption and authentication processes separately, running the underlying primitive twice per message block. A single-pass AE performs these tasks in a single processing. To be more specific, it calls the primitive function or permutation once per input block, in addition to a few constant number of calls. Firstly, Julta [21] introduces the first secure single-pass authenticated encryption IAPM. Rogaway et al. [22] extend this idea to build OCB, which is a fast and well-featured AE [23, 14].

Scheme 3 is the third proposed AE scheme, referred to as LAE3. It is derived from a recent AE from Minematsu [25] called OTR. OTR is an extension to OCB, which uses a Feistel structure to replace each pair of block ciphers in OCB with a pair of PRFs. Figure 3 shows the structure and data flow of LAE3. In this scheme, similar to the last block of LAE1, a whitening value is XORed with the input of SPRING to obtain a tweakable PRF. The polynomial operations in this scheme are performed in $\text{GF}(2^\ell)$. Note that the multiplication of small constants, for instance in 4δ and $2L$, are easily computed by some shifts and XORs. The input of SPRING is truncated to ℓ bits in LAE3. The remaining least significant $(n-\ell)$ bits are padded with zero.

There are two differences between LAE3 and OTR. Firstly, the block cipher is replaced with a PRF. Although [25] introduces a PRF-capable version of OTR; however, there were technical issues about the integration of SPRING in this scheme. The proposed scheme does not use the OTR technique to obtain a tweakable PRF from SPRING. Instead, some whitening value is XORed with the input of SPRING. The other difference is that the input and output length

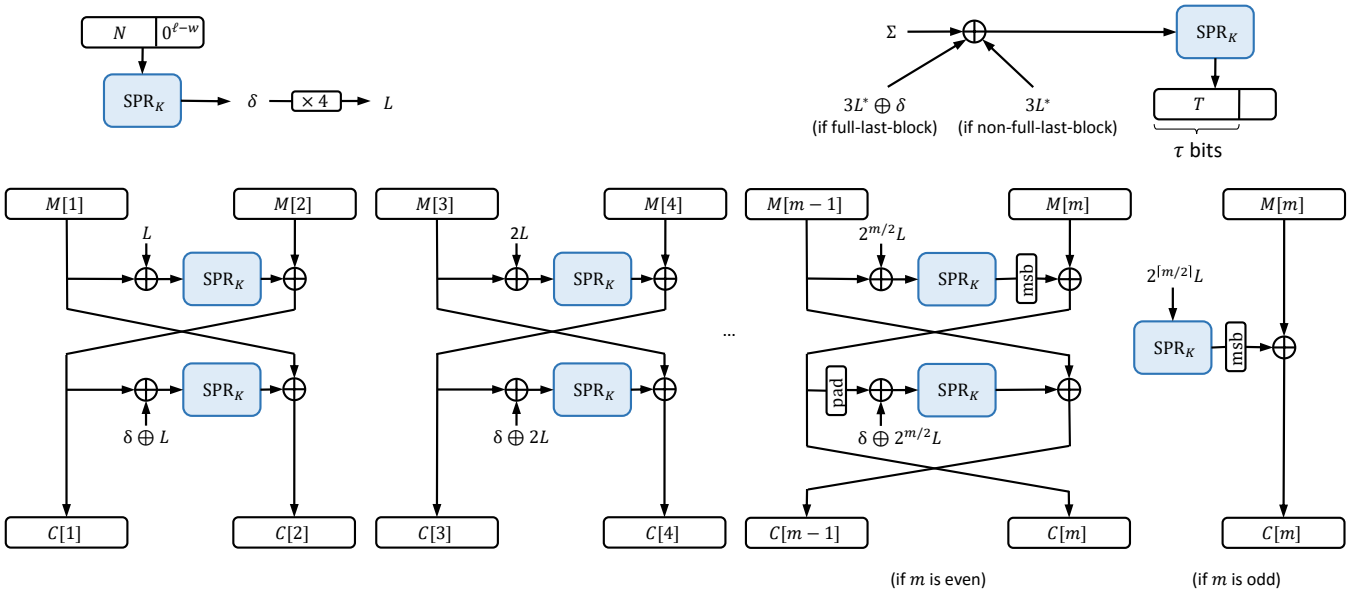

Fig. 3: The flow diagram of LAE3 authenticated encryption procedure. For the description of L^* and Σ, refer to Scheme 3.

Scheme 3 (LAE3)

procedure AUTHENCRYPT(N, M)
  $\Sigma \leftarrow 0^\ell$
  $\delta \leftarrow \mathrm{SPR}_K(N \,\|\, 0^{\ell-w})$
  $L \leftarrow 4\delta$
  $(M[1], ..., M[m]) \xleftarrow{\ell} M$
  for $i \leftarrow 1$ **to** $\lceil m/2 \rceil$ **do**
    $C[2i-1] \leftarrow \mathrm{SPR}_K\big(L \oplus M[2i-1]\big) \oplus M[2i]$
    $C[2i] \leftarrow \mathrm{SPR}_K\big(L \oplus \delta \oplus C[2i-1]\big) \oplus M[2i-1]$
    $\Sigma \leftarrow \Sigma \oplus M[2i]$
    $L \leftarrow 2L$
  if m is even **then**
    $L^* \leftarrow L \oplus \delta$
    $Z \leftarrow \mathrm{SPR}_K\big(L \oplus M[m-1]\big)$
    $C[m] \leftarrow \mathsf{msb}_{|M[m]|}(Z) \oplus M[m]$
    $C[m-1] \leftarrow \mathrm{SPR}_K\big(L^* \oplus \mathsf{pad}_\ell(C[m])\big) \oplus M[m-1]$
    $\Sigma \leftarrow \Sigma \oplus Z \oplus \mathsf{pad}_\ell(C[m])$
  if m is odd **then**
    $L^* \leftarrow L$
    $C[m] \leftarrow \mathsf{msb}_{|M[m]|}\big(\mathrm{SPR}_K(L^*)\big) \oplus M[m]$
    $\Sigma \leftarrow \Sigma \oplus M[m]$
  if $|M[m]| \neq n$ **then**
    $T \leftarrow \mathsf{msb}_\tau\Big(\mathrm{SPR}_K\big(3L^* \oplus \Sigma\big)\Big)$
  else
    $T \leftarrow \mathsf{msb}_\tau\Big(\mathrm{SPR}_K\big(3L^* \oplus \delta \oplus \Sigma\big)\Big)$
  $C \leftarrow C[1] \,\|...\|\, C[m]$
  return (C, T)

of SPRING are not equal. We use only ℓ bits of SPRING input (padding it with zeroes) to make the input and output length the same.

Design Rationale. There are few proposals for single-pass non-lattice-based authenticated encryption in the literature. The methods of IAPM [21] and OCB [22, 23, 14] cannot be followed in the case of SPRING. In these schemes, the plaintext is directly fed into the underlying primitive function and the output becomes a part of the ciphertext. PRFs are not useful in this setting because a PRF is not necessarily a bijection; hence, it has data loss. Only a pseudorandom permutation (e.g., a block cipher) can be used in such designs. Minematsu [25] introduces OTR as a PRF-capable single-pass AE. OTR utilizes the tweak-prepending technique to construct a tweakable PRF from a normal one. Unfortunately, this technique cannot be applied to SPRING. That is because SPRING has a fixed and short input length n, while tweak-prepending requires a variable-input-length PRF or one with a large-enough input length. Alternatively, the input whitening technique is used in LAE3. In this case, the outputs of some functions of an XOR-universal class is XORed with the input of SPRINGs to make a tweakable PRF. As a final note, LAE3 requires a fresh δ for each nonce N. Thus, we cannot save a SPRING invocation, similar to LAE1, by introducing a second key.

4.1 Security of LAE3

Theorem 5 and Theorem 6 show the security of LAE3.

Theorem 5 (Privacy of LAE3). Fix $n \geq \ell \geq 1$. Let $\mathcal{F} = \{F : \{0,1\}^n \rightarrow \{0,1\}^\ell\}$ be a family of functions. For any adversary $\mathcal{A}$ to attack the privacy of $LAE3[\mathcal{F}]$, who runs in time t and asks q encryption queries, with a maximum total length of σ blocks, there exists an adversary $\mathcal{P}$ against the pseudorandomness of $\mathcal{F}$, and we have

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{LAE3[\mathcal{F}]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}) + \frac{6(q+\sigma)^2}{2^\ell}. \qquad (25)$$

Moreover, adversary $\mathcal{P}$ asks $q' = \sigma + 2q$ oracle queries, and runs in time $t' = t + \alpha\ell(\sigma + q)$, where α is a constant depending on the model of the computation.

Theorem 6 (Authenticity of LAE3). Fix $n \geq \ell \geq 1$. Let $\mathcal{F} = \{F : \{0,1\}^n \rightarrow \{0,1\}^\ell\}$ be a family of functions. For any adversary $\mathcal{A}$ to attack the authenticity of $LAE3[\mathcal{F}]$, who runs in time t and asks q_e encryption and

q_d decryption queries, with a maximum total length of σ_e and σ_d blocks, respectively, there exists an adversary $\mathcal{P}$ against the pseudo-randomness of $\mathcal{F}$, and we have

$$\mathbf{Adv}^{\text{int-ctxt}}_{LAE3[\mathcal{F}]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{prf}}_{\mathcal{F}}(\mathcal{P}) + \frac{6(q_e + q_d + \sigma_e + \sigma_d)^2}{2^{\ell}} + \frac{q_d}{2^{\tau}}. \quad (26)$$

Moreover, adversary $\mathcal{P}$ asks $q' = q_e + q_d + \sigma_e + \sigma_d$ number of oracle queries, and runs in time $t' = t + \alpha\ell(\sigma_e + \sigma_d + 2q_e + 2q_d)$, where α is a constant depending on the model of the computation.

The proof of Theorem 5 and Theorem 6 are similar to the privacy and authenticity proofs of OTR [25, Theorems 1 and 2]. The main difference is that the associated data is not involved in the proof because LAE3 does not support it.

5 Comparison and Performance Results

In this section, the efficiency and performance results of the proposed schemes are reported. These schemes are implemented with moderate optimizations. The target of the implementations are high-end processors because single-instruction multiple-data (SIMD) instructions are utilized to speed up polynomial operations of the lattice-based AEs. The source code of the implementations are integrated into the OpenSSL framework to be benchmarked along with the optimized implementation of AES-128-GCM and AES-256-GCM in OpenSSL. GCM [26] is an efficient and widely-used authenticated encryption mode of operation. AES-128-GCM and AES-256-GCM are two instantiations of GCM using 128-bit and 256-bit AES, respectively. The implemented lattice-based schemes are parameterized with $n = 128$, $\ell = 127$, and $w = 96$ to claim 128-bit security in the pre- and post-quantum setting. Consequently, AES-256-GCM is also included in the comparison which is 128-bit secure in the post-quantum setting. The three proposed schemes LAE1, LAE2, and LAE3 are implemented based on the SPRING implementation provided by its designers [16]. The implementation of the SPRING function in LAE3 and in the authentication part of LAE1 cannot make use of the optimizations introduced in [16, Section 3] regarding the gray-code input.

The proposed schemes are implemented in C++ and integrated with the benchmark of OpenSSL version 1.0.1i. The OpenSSL benchmark is modified to feed more variable sizes of input, and use the "rdtsc" instruction to count the processor clocks. The instruction set of the Intel Carry-Less Multiplication (CLMUL) is used to multiply polynomials in $\mathrm{GF}(2^{128})$. The benchmarking process is executed one CPU core. The source code is compiled using GCC version 4.8.2. Two compiler options "-O3" and "-march=native" are specified in order to configure the compiler to generate optimized codes according to the targeted processors. Thus, the source code is recompiled on each targeted machine. The Intel SSE2 instruction set is utilized indirectly by using vector processing features of GCC. OpenSSL, as well as the integrated new schemes, were built into a 64-bit binary executable. The benchmark is run on a 64-bit Linux machine with kernel version 3.13.30. Two Intel CPUs are used in the experiments. The first one is Intel Core i3-2120 running at 3.30GHz with Sandy Bridge microarchitecture, and the second one is Intel Core i7-4770 running at 3.4GHz with Haswell microarchitecture.

TABLE 1: The percentage increase or decrease of the encryption clock cycles of the proposed schemes LAE1, LAE2, and LAE3 in comparison to AES-256-GCM on the Sandy Bridge microarchitecture.

Scheme	Sample input lengths (byte)				
	16	**40**	**64**	**128**	**1500**
LAE1	450%	205%	186%	135%	87%
LAE2	74%	12%	-3%	-19%	-34%
LAE3	367%	146%	119%	72%	28%
AES-128-GCM	-23%	-22%	-25%	-27%	-29%
AES-256-GCM	0%	0%	0%	0%	0%

TABLE 2: The percentage increase or decrease of the encryption clock cycles of the proposed schemes LAE1, LAE2, and LAE3 in comparison to AES-256-GCM on the Haswell microarchitecture.

Scheme	Sample input lengths (byte)				
	16	**40**	**64**	**128**	**1500**
LAE1	470%	222%	206%	156%	114%
LAE2	80%	24%	9%	-2%	-15%
LAE3	374%	151%	123%	76%	36%
AES-128-GCM	-23%	-27%	-25%	-28%	-26%
AES-256-GCM	0%	0%	0%	0%	0%

Figure 4 shows the processor clock cycles on Intel Sandy Bridge and Haswell microarchitectures, running the encryption procedure of LAE1, LAE2, and LAE3. The lengths chosen for the plaintexts are powers of 2 and 10 bytes in order to cover both complete and incomplete last blocks, as well as common IP packet sizes 40, 576, 1300, and 1500 bytes. Note that OpenSSL is configured not to use AES-NI instruction set of Intel CPUs for AES-128-GCM and AES-256-GCM.

Although LAE3 is single-pass, it is not more efficient than the two-pass LAE2. That is due to the optimization techniques applied on the lattice-based PRF SPRING. SPRING is very efficient if it runs multiple times given the values of a gray-code counter. In this environment, the aggregated computation of each SPRING is reduced significantly. In the case of LAE1, only the encryption part enjoys the optimization for the gray-code input. As an improvement, the authentication part of LAE2 utilizes $\mathrm{GF}(2^{128})$ multiplications instead of a chain of SPRINGs. There is only one SPRING invocation in the authentication part, which is also in the sequence of the encryption gray-code counter. Therefore, SPRING computation is highly optimized in LAE2. The authentication pass is removed in LAE3, though, the SPRING's inputs are no longer from a gray-code counter, resulting to a less efficient scheme than LAE2. All the three proposed schemes are computationally heavy for very short plaintexts (e.g., less than 40 bytes). Only LAE2 becomes good for larger inputs. This phenomena is common in symmetric encryption schemes, however, it is a more considerable issue in the case of the proposed

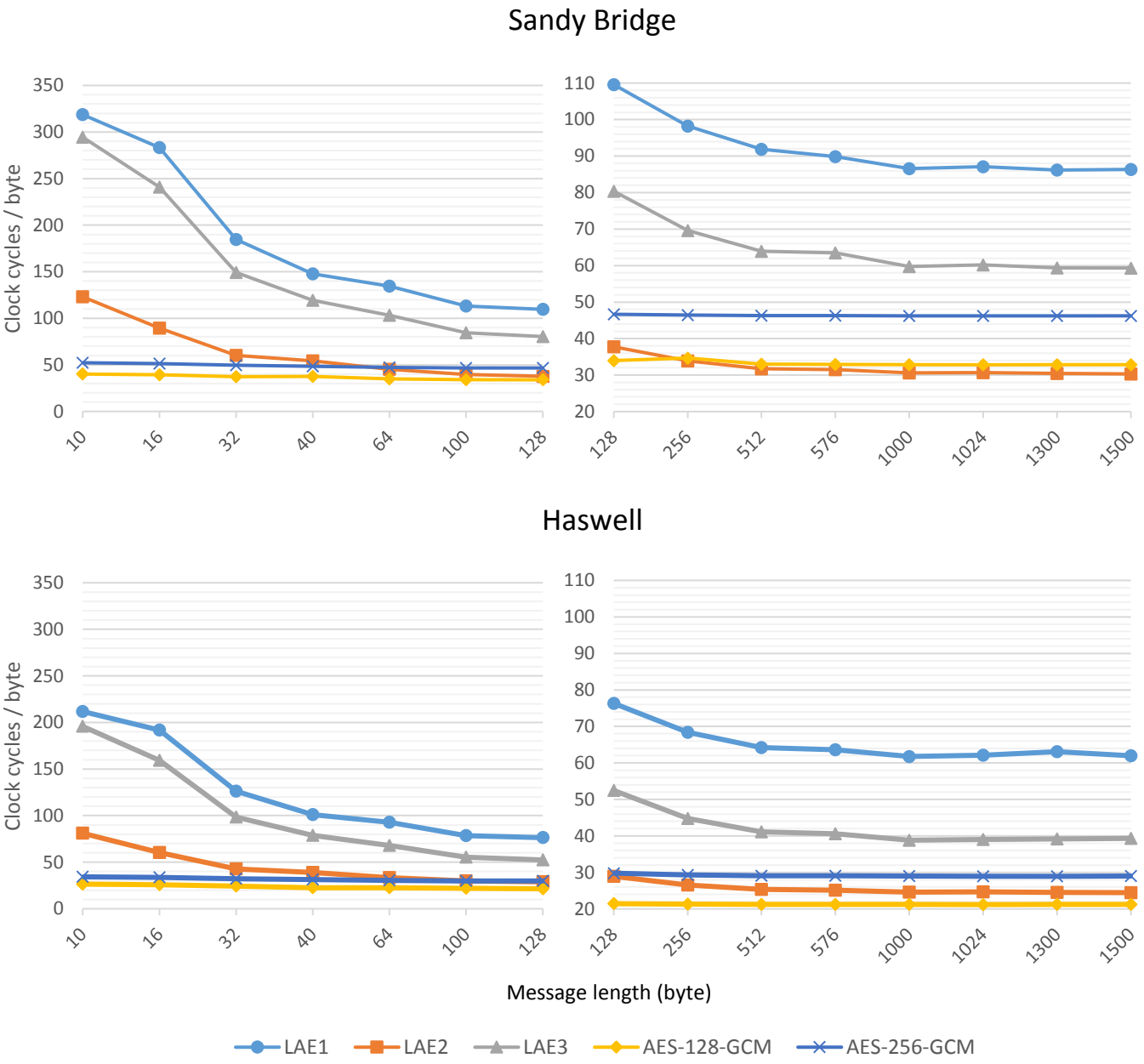

Fig. 4: Performance results of the proposed schemes LAE1, LAE2, and LAE3 (encryption procedure), compared with AES-128-GCM and AES-256-GCM. The implementations are executed on one core of Intel Core i3-2120 with Sandy Bridge microarchitecture, and on one core of Intel Core i7-4770 with Haswell microarchitecture.

lattice-based schemes because the saving in the aggregated computation of SPRINGs is substantial.

Table 1 and Table 2 present the percentage increase or decrease of the clock cycles of LAE1, LAE2, and LAE3 in comparison to the clock cycles of AES-256-GCM, for the Sandy Bridge and Haswell microarchitectures, respectively. LAE2 becomes faster than AES-256-GCM from the messages of length 64 bytes and 128 bytes for Sandy Bridge and Haswell microarchitectures, respectively. It is noted that, on the Sandy Bridge processor the performance of LAE2 advances above AES-128-GCM for 512-byte and longer messages. For a message of length 1500 bytes, LAE2 encryption performs 8% faster than AES-128-GCM. Table 3 also shows the comparison of key sizes between the proposed schemes and the two references AES-128-GCM and AES-256-GCM. Like most lattice-based cryptographic schemes, the key sizes of the proposed AEs are very large. These sizes are for the key before applying the fast Fourier transform (FFT).

6 Conclusion

In this paper, we have proposed three authenticated encryption (AE) schemes LAE1, LAE2, and LAE3, which enjoy a security proof based on hard lattice problems. These AE schemes have an important property that they are based on the hardness of a mathematical problem (a lattice basic problem). That is in contrast to the previous practical provably-secure AEs based on the heuristic hardness of a cryptographic primitive, such as a block cipher.

TABLE 3: Key sizes of the proposed schemes LAE1, LAE2, and LAE3 in comparison with AES-128-GCM and AES-256-GCM.

Scheme	Key Size (bits)	
LAE1	98431	$(2 \times 49152 + 127)$
LAE2	49280	$(49152 + 128)$
LAE3	49152	
AES-128-GCM	128	
AES-256-GCM	256	

Moreover, inheriting the conjectured quantum-resistance of lattice problems, the proposed schemes are secure against quantum attacks. In addition, we have analyzed and proved the exact security of these schemes in the practice-oriented provable security paradigm. This approach is not common in lattice-based cryptography; however, mitigates many security risks that provably-secure schemes have in practice. The implementation results on the Intel Sandy Bridge and Haswell microarchitectures show that LAE2 is efficient enough to be used in practice and to compete widely-used AEs. For instance, LAE2 becomes faster than AES-256-GCM on Sandy Bridge to encrypt messages of length 64 bytes or longer. Particularly, for a 1500-byte message, this scheme is 34% faster than AES-256-GCM. The main comparison is performed with AES-256-GCM because this scheme and LAE2 are both 128-bit secure in the post-quantum setting.

ACKNOWLEDGMENTS

The authors would like to thank Banerjee et al. [16] for providing the source code of their SPRING implementation. They also thank Mohammad Razeghi for his help in the implementation of the proposed schemes.

REFERENCES

[1] D. J. Bernstein, "Introduction to post-quantum cryptography," in *Post-Quantum Cryptography*, D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds. Springer Berlin Heidelberg, 2009, pp. 1–14.

[2] C. Peikert, "A Decade of Lattice Cryptography," ePrint IACR, Tech. Rep. 939, 2015.

[3] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*. IEEE, 1994, pp. 124–134.

[4] J. Proos and C. Zalka, "Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves," *Quantum Info. Comput.*, vol. 3, no. 4, pp. 317–344, Jul. 2003.

[5] R. Lindner and C. Peikert, "Better Key Sizes (and Attacks) for LWE-Based Encryption," in *Topics in Cryptology – CT-RSA 2011*, ser. Lecture Notes in Computer Science, A. Kiayias, Ed. Springer, 2011, no. 6558, pp. 319–339.

[6] D. Micciancio and C. Peikert, "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller," in *Advances in Cryptology – EUROCRYPT 2012*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds. Springer, 2012, no. 7237, pp. 700–718.

[7] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice Signatures and Bimodal Gaussians," in *Advances in Cryptology – CRYPTO 2013*, ser. Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds. Springer, 2013, no. 8042, pp. 40–56.

[8] N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. Huss, "On the Design of Hardware Building Blocks for Modern Lattice-Based Encryption Schemes," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, ser. Lecture Notes in Computer Science, E. Prouff and P. Schaumont, Eds. Springer, 2012, no. 7428, pp. 512–529.

[9] T. Oder, T. Pöppelmann, and T. Güneysu, "Beyond ECDSA and RSA: Lattice-based Digital Signatures on Constrained Devices," in *Proceedings of the 51st Annual Design Automation Conference*, ser. DAC '14. New York, NY, USA: ACM, 2014, pp. 110:1–110:6.

[10] A. Boorghany and R. Jalili, "Implementation and Comparison of Lattice-based Identification Protocols on Smart Cards and Microcontrollers," ePrint IACR, Tech. Rep. 078, 2014.

[11] A. Boorghany, S. Bayat Sarmadi, and R. Jalili, "On Constrained Implementation of Lattice-based Cryptographic Primitives and Schemes on Smart Cards," *IACR Cryptology ePrint Archive*, vol. Report 2014/514, 2014.

[12] R. Azarderakhsh, Z. Liu, H. Seo, and H. Kim, "NEON PQCryto: Fast and Parallel Ring-LWE Encryption on ARM NEON Architecture," ePrint IACR, Tech. Rep. 1081, 2015.

[13] "CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness," http://competitions.cr.yp.to/caesar-call.html, 2013.

[14] T. Krovetz and P. Rogaway, "The Software Performance of Authenticated-Encryption Modes," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science, A. Joux, Ed. Springer, Jan. 2011, no. 6733, pp. 306–327.

[15] A. Banerjee, C. Peikert, and A. Rosen, "Pseudorandom Functions and Lattices," in *Advances in Cryptology – EUROCRYPT 2012*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds. Springer, 2012, no. 7237, pp. 719–737.

[16] A. Banerjee, H. Brenner, G. Leurent, C. Peikert, and A. Rosen, "SPRING: Fast Pseudorandom Functions from Rounded Ring Products," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science, C. Cid and C. Rechberger, Eds. Springer Berlin Heidelberg, Mar. 2014, no. 8540, pp. 38–57.

[17] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: ACM, 1996, pp. 212–219.

[18] J. Black and P. Rogaway, "CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions," *Journal of Cryptology*, vol. 18, no. 2, pp. 111–131, Apr. 2005.

[19] K. Kurosawa and T. Iwata, "TMAC: Two-Key CBC MAC," in *Topics in Cryptology — CT-RSA 2003*, ser. Lecture Notes in Computer Science, M. Joye, Ed. Springer Berlin Heidelberg, 2003, no. 2612, pp. 33–49.

[20] T. Iwata and K. Kurosawa, "OMAC: One-Key CBC MAC," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science, T. Johansson, Ed. Springer Berlin Heidelberg, 2003, no. 2887, pp. 129–153.

[21] C. S. Jutla, "Encryption Modes with Almost Free Message Integrity," in *Advances in Cryptology — EUROCRYPT 2001*, ser. Lecture Notes in Computer Science, B. Pfitzmann, Ed. Springer Berlin Heidelberg, Jan. 2001, no. 2045, pp. 529–544.

[22] P. Rogaway, M. Bellare, and J. Black, "OCB: A Block-cipher Mode of Operation for Efficient Authenticated Encryption," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 3, pp. 365–403, Aug. 2003, oCB1.

[23] P. Rogaway, "Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC," in *Advances in Cryptology - ASIACRYPT 2004*, ser. Lecture Notes in Computer Science, P. J. Lee, Ed. Springer, Jan. 2004, no. 3329, pp. 16–31, oCB2.

[24] V. D. Gligor and P. Donescu, "Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science, M. Matsui, Ed. Springer Berlin Heidelberg, Apr. 2001, no. 2355, pp. 92–108.

[25] K. Minematsu, "Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions," in *Advances in Cryptology – EUROCRYPT 2014*, ser. Lecture Notes in Computer Science, P. Q. Nguyen and E. Oswald, Eds. Springer Berlin Heidelberg, Jan. 2014, no. 8441, pp. 275–292.

[26] D. A. McGrew and J. Viega, "The Security and Performance of the Galois/Counter Mode (GCM) of Operation," in *Progress in Cryptology - INDOCRYPT 2004*, ser. Lecture Notes in Computer Science, A. Canteaut and K. Viswanathan, Eds. Springer, Jan. 2005, no. 3348, pp. 343–355.

[27] P. Rogaway, "Authenticated-encryption with Associated-data," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 98–107.

[28] P. Rogaway and T. Shrimpton, "A Provable-Security Treatment of the Key-Wrap Problem," in *Advances in Cryptology - EUROCRYPT 2006*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed. Springer Berlin Heidelberg, May 2006, no. 4004, pp. 373–390.

[29] M. Bellare, J. Kilian, and P. Rogaway, "The Security of the Cipher Block Chaining Message Authentication Code," *Journal of Computer and System Sciences*, vol. 61, no. 3, pp. 362–399, Dec. 2000.

[30] J. Black and P. Rogaway, "A Block-Cipher Mode of Operation for Parallelizable Message Authentication," in *Advances in Cryptology — EUROCRYPT 2002*, ser. Lecture Notes in Computer Science, L. R. Knudsen, Ed. Springer Berlin Heidelberg, 2002, no. 2332, pp. 384–397.

[31] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm," *Journal of Cryptology*, vol. 21, no. 4, pp. 469–491, Oct. 2008.

[32] C. Namprempre, P. Rogaway, and T. Shrimpton, "Reconsidering Generic Composition," in *Advances in Cryptology – EUROCRYPT 2014*, ser. Lecture Notes in Computer Science, P. Q. Nguyen and E. Oswald, Eds. Springer Berlin Heidelberg, May 2014, no. 8441, pp. 257–274.

[33] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," in , *38th Annual Symposium on Foundations of Computer Science, 1997. Proceedings*, Oct. 1997, pp. 394–403.

[34] M. N. Wegman and J. L. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, vol. 22, no. 3, pp. 265–279, Jun. 1981.

[35] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs," ePrint IACR, Tech. Rep. 332, 2004.

[36] M. Bellare and P. Rogaway, "The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs," in *Advances in Cryptology - EUROCRYPT 2006*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed. Springer Berlin Heidelberg, May 2006, no. 4004, pp. 409–426.

Ahmad Boorghany received the B.Sc. degree in software engineering and the M.Sc. degree in information technology, in 2010 and 2012, respectively, both from the Sharif University of Technology, Tehran, Iran. He is now a Ph.D. candidate in computer engineering, in the Department of Computer Engineering, Sharif University of Technology. His research interests are lattice-based and modern cryptography, provable security, and building efficient cryptographic schemes and protocols. He is a member of the International Association for Cryptologic Research (IACR).

Siavash Bayat-Sarmadi received the B.Sc. degree from the University of Tehran, Iran, in 2000, the M.Sc. degree from Sharif University of Technology, Tehran, Iran, in 2002, and the PhD degree from the University of Waterloo in 2007, all in computer engineering (hardware). He was with Advanced Micro Devices, Inc. for about 6 years. Since September 2013, he has been a faculty member in the Department of Computer Engineering, Sharif University of Technology. He has served on the executive committees of several conferences. His research interests include hardware security and trust, cryptographic computations, and secure, efficient and dependable computing and architectures. He is a member of the IEEE.

Rasool Jalili received his B.S. degree in Computer Science from Ferdowsi University of Mashhad in 1985, and M.S. degree in Computer Engineering from Sharif University of Technology in 1989. He received his Ph.D. in Computer Science from The University of Sydney, Australia, in 1995. He then joined the Department of Computer Engineering, Sharif University of Technology in 1995. He has published more than 140 papers in international journals and conference proceedings. He is now an associate professor, doing research in the areas of computer dependability and security, access control, distributed systems, and database systems in his Data and Network Security Laboratory (DNSL).

APPENDIX A
OMITTED PROOFS

A.1 Security of *fFCBC*

In this section, the information-theoretic security of *fFCBC* is presented. At first, suppose *fCBC* is the plain CBC mode, instantiated with a PRF function. The collision probability of *fCBC* is defined as follows.

$$\mathrm{Col}_{n,\ell}(m,m') = \max_{M\neq M'} \Big\{ \Pr\Big[\rho \xleftarrow{\$} Func(n,\ell) : fCBC_\rho(M) = fCBC_\rho(M')\Big]\Big\}, \quad (27)$$

where the messages M, M' are chosen from $\{0,1\}^{m\ell}$ and $\{0,1\}^{m'\ell}$, respectively.

Lemma 5 (Collision resistance of fCBC). Fix $m, m' \geq 1$. The following bound holds for the *fCBC* collision probability.

$$\mathrm{Col}_{n,\ell}(m,m') \leq \frac{(m+m')^2+1}{2^\ell} \quad (28)$$

The proof of Lemma 5. Fix two messages M, M', where $|M| = m\ell$, $|M'| = m'\ell$, and $M \neq M'$. Moreover, assume that the first k blocks of M and M' are equal (k may be zero). Now, Game 3 presents the computation of two *fCBC* functions. The collision is occurred when $Y_m = Y'_{m'}$.

Consider the case that the bad flag is never set to `true`. Then, all the Y_i and Y'_i variables in lines 13 and 19, are set to uniform and independent values. Thus, if $k < m$ and $k < m'$, or if either $k = m$ or $k = m'$ (but not both, because $M \neq M'$), then Y_m and $Y'_{m'}$ are uniformly distributed and independent. Thus, we have

$$\Pr[Y_m = Y'_{m'} \mid \mathrm{bad} = \texttt{false}] = \frac{1}{2^\ell}, \quad (29)$$

where the probability is taken over the random function ρ. Moreover, the following equations hold.

$$\begin{aligned}\mathrm{Col}_{n,\ell}(m,m') = \Pr[Y_m = Y'_{m'}] = \\ \Pr[Y_m = Y'_{m'} \wedge \mathrm{bad} = \texttt{false}] + \\ \Pr[Y_m = Y'_{m'} \wedge \mathrm{bad} = \texttt{true}],\end{aligned} \quad (30)$$

$$\begin{aligned}\Pr[Y_m = Y'_{m'} \wedge \mathrm{bad} = \texttt{false}] \leq \\ \frac{\Pr[Y_m = Y'_{m'} \wedge \mathrm{bad} = \texttt{false}]}{\Pr[\mathrm{bad} = \texttt{false}]} = \\ \Pr[Y_m = Y'_{m'} \mid \mathrm{bad} = \texttt{false}] = \frac{1}{2^\ell},\end{aligned} \quad (31)$$

$$\Pr[Y_m = Y'_{m'} \wedge \mathrm{bad} = \texttt{true}] \leq \Pr[\mathrm{bad} = \texttt{true}]. \quad (32)$$

Thus, we have

$$\mathrm{Col}_{n,\ell}(m,m') \leq \Pr[\mathrm{bad} = \texttt{true}] + \frac{1}{2^\ell}. \quad (33)$$

Now, to bound the probability of setting the bad flag, note that it is set to true in lines 5, 11, and 17. Just before these lines X_i or X'_i is assigned, and if it is in the domain of ρ, the bad flag is set. For the case that the bad flag has not been yet set to true, the value assigned to X_i or X'_i is uniformly-random and independent. That is because the value of Y_{i-1} or Y'_{i-1} is uniformly-random and independent, and it is XORed with the message to form X_i or X'_i. If there are $i-1$ values in the domain of ρ, the probability of a collision for the first time for a new random value is $(i-1)/2^\ell$. Thus, the following equation holds.

$$\Pr[\mathrm{bad} = \texttt{true}] \leq \sum_{i-1}^{m+m'} \frac{i-1}{2^\ell} \leq \frac{(m+m')^2}{2^\ell}. \quad (34)$$

And finally we have

$$\mathrm{Col}_{n,\ell}(m,m') \leq \frac{(m+m')^2+1}{2^\ell}. \ \square \quad (35)$$

Theorem 7 (Security of fFCBC). For any adversary $\mathcal{A}$ who asks q oracle queries, each of which with a maximum length of m blocks, we have

$$\begin{aligned}\Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{fFCBC[\rho_1,\rho_2,\rho_3](\cdot)} \Rightarrow 1] - \\ \Pr[\rho \xleftarrow{\$} \mathrm{Func}(*,\ell) : \mathcal{A}^{\rho(\cdot)} \Rightarrow 1] \leq \frac{(4m^2+1)q^2}{2^{\ell+1}}.\end{aligned} \quad (36)$$

Proof of Theorem 7. Lets define Col as the event of occurring a collision in the outputs of ρ_2 or in the outputs of ρ_3 (but not between them). Thus, we can break the left side of the theorem equation to

$$\begin{aligned}\Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{fFCBC[\rho_1,\rho_2,\rho_3](\cdot)} \Rightarrow 1 \mid \mathrm{Col}]\Pr[\mathrm{Col}] + \\ \Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{fFCBC[\rho_1,\rho_2,\rho_3](\cdot)} \Rightarrow 1 \mid \overline{\mathrm{Col}}]\Pr[\overline{\mathrm{Col}}] - \\ \Pr[\rho \xleftarrow{\$} \mathrm{Func}(*,\ell) : \mathcal{A}^{\rho(\cdot)} \Rightarrow 1].\end{aligned} \quad (37)$$

Using the following simple equations

$$\Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{fFCBC[\rho_1,\rho_2,\rho_3](\cdot)} \Rightarrow 1 \mid \mathrm{Col}] \leq 1, \quad (38)$$

$$\Pr[\overline{\mathrm{Col}}] \leq 1, \quad (39)$$

we can bound the left-hand side of the theorem equation as follows.

$$\begin{aligned}\Pr[\mathrm{Col}] + \\ \Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{fFCBC[\rho_1,\rho_2,\rho_3](\cdot)} \Rightarrow 1 \mid \overline{\mathrm{Col}}] - \\ \Pr[\rho \xleftarrow{\$} \mathrm{Func}(*,\ell) : \mathcal{A}^{\rho(\cdot)} \Rightarrow 1].\end{aligned} \quad (40)$$

Game 3 (The game of *fCBC* collision)

```
1: bad ← false; Dom(ρ) ← φ
2: for i ← 1 to k do
3:     if i = 1 then X_i ← X'_i ← M_1
4:     else X_i ← X'_i ← Y_{i-1} ⊕ M_i
5:     if X_i ∈ Dom(ρ) then bad ← true
6:     else ρ(X_i) ←$ {0,1}^ℓ
7:     Y_i ← Y'_i ← ρ(X_i)
8: for i ← k+1 to m do
9:     if i = 1 then X_i ← M_1
10:    else X_i ← Y_{i-1} ⊕ M_i
11:    if X_i ∈ Dom(ρ) then bad ← true
12:    else ρ(X_i) ←$ {0,1}^ℓ
13:    Y_i ← ρ(X_i)
14: for i ← k+1 to m' do
15:    if i = 1 then X'_i ← M'_1
16:    else X'_i ← Y'_{i-1} ⊕ M'_i
17:    if X'_i ∈ Dom(ρ) then bad ← true
18:    else ρ(X'_i) ←$ {0,1}^ℓ
19:    Y'_i ← ρ(X'_i)
```

If there is no collision, it is apparent that

$$\Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{f^{FCBC[\rho_1,\rho_2,\rho_3](\cdot)}} \Rightarrow 1 \mid \overline{\mathrm{Col}}] = \Pr[\rho \xleftarrow{\$} \mathrm{Func}(*,\ell) : \mathcal{A}^{\rho(\cdot)} \Rightarrow 1], \tag{41}$$

and therefore, we have

$$\Pr[\rho_1,\rho_2,\rho_3 \xleftarrow{\$} \mathrm{Func}(n,\ell) : \mathcal{A}^{f^{FCBC[\rho_1,\rho_2,\rho_3](\cdot)}} \Rightarrow 1] - \Pr[\rho \xleftarrow{\$} \mathrm{Func}(*,\ell) : \mathcal{A}^{\rho(\cdot)} \Rightarrow 1] \leq \Pr[\overline{\mathrm{Col}}]. \tag{42}$$

Moreover, using the union bound,

$$\Pr[\mathrm{Col}] \leq \Pr[\mathrm{PadCol}] + \Pr[\mathrm{UnpadCol}], \tag{43}$$

where $\mathrm{UnpadCol}$ is the event of a collision in the outputs of ρ_3 for unpadded messages, and PadCol is the event of a collision in the outputs of ρ_2 for padded messages.

Now, we can break the probability of PadCol as follows.

$$\Pr[\mathrm{PadCol}] \leq \sum_{i=1}^{q_{\mathrm{pad}}} \Pr[\mathrm{PadCol}_i], \tag{44}$$

where $\Pr[\mathrm{PadCol}_i]$ is the probability that the collision is occurred for the first time after the i-th padded query, and q_{pad} is the total number of padded queries. Note that PadCol_i's are disjoint. Because there is no collision in the first $i-1$ queries, the response of these queries, sent to the adversary, are uniformly random and independent. Thus, any adaptive strategy of $\mathcal{A}$ can be changed to a nonadaptive strategy without loos of advantage.

Using the result of Lemma 5, we have

$$\Pr[\mathrm{PadCol}_i] \leq (i-1)\mathrm{Col}_{n,\ell}(m,m), \tag{45}$$

and the final bound on the probability of PadCol is

$$\Pr[\mathrm{PadCol}] \leq \sum_{i=1}^{q_{\mathrm{pad}}} (i-1)\mathrm{Col}_{n,\ell}(m,m) \leq \frac{(4m^2+1)q_{\mathrm{pad}}^2}{2^{\ell+1}}. \tag{46}$$

The same bound is also applied for $\mathrm{UnpadCol}$. Finally, the theorem result is obtained:

$$\begin{aligned}\Pr[\mathrm{Col}] &\leq \frac{(4m^2+1)q_{\mathrm{pad}}^2}{2^{\ell+1}} + \frac{(4m^2+1)q_{\mathrm{unpad}}^2}{2^{\ell+1}} \\ &\leq \frac{(4m^2+1)q^2}{2^{\ell+1}}. \ \square\end{aligned} \tag{47}$$

A.2 Authenticity Proof of LAE1

Proof of Theorem 2. Based on the pseudorandomness of LAE1-MAC (Lemma 3), it is standard to prove the following equation about the unforgeability of LAE1-MAC under chosen message attack:

$$\mathbf{Adv}^{\mathsf{uf\text{-}cma}}_{\mathit{LAE1\text{-}MAC[\mathcal{F}]}}(\mathcal{B}) \leq \mathbf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{P}) + \frac{(4m^2+1)q^2+2}{2^{\ell+1}} + \frac{q^2}{2^{n+1}}. \tag{48}$$

Now, adversary $\mathcal{A}$ can be utilized to build the adversary $\mathcal{B}$. This adversary acts as follows. For any encryption queries $\mathcal{A}$ makes, $\mathcal{B}$ uses its oracle to obtain T. Then, it generates a random key K_1 to compute C. Moreover, upon a decryption query from $\mathcal{A}$, it decrypts C to obtain M^*, and sends (N, M^*, T) to its forgery (decryption) oracle. As a result, we simply have

$$\mathbf{Adv}^{\mathsf{int\text{-}ctxt}}_{\mathit{LAE1[\mathcal{F}]}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{uf\text{-}cma}}_{\mathit{LAE1\text{-}MAC[\mathcal{F}]}}(\mathcal{B}), \tag{49}$$

and the theorem is proved. $\square$