LAEM
(Lightweight Authentication Encryption Mode)

Submitters: Han Sui, Wenling Wu, Lei Zhang*, Danxia Zhang

Email: zhanglei@iscas.ac.cn

Telephone: 86-10-62661708

Organization: TCA, Institute of Software, Chinese Academy of Sciences

Address: South Fourth Street, Zhong Guan Cun, Beijing, China, 100190

March 25, 2019

Contents

Chapter 1

Introduction

We propose a lightweight authenticated encryption mode, which is named LAEM. As an instance, we adopt the lightweight block cipher SIMON[2] with 128-bit block size and 128/192/256-bit key size as the underlying block cipher. The final algorithm of authenticated encryption with associated data is denoted as LAEM-SIMON, and it ensures both the confidentiality of the plaintexts and the integrity of the ciphertexts.

The recommended parameter sets for LAEM-SIMON are listed in Table 1.1. Each parameter is an integer number of bytes. The block size is 16 bytes (128-bit). The nonce size is 16 bytes (128-bit). The key size can be 16/24/32 bytes (128/192/256-bit). The primary recommended parameter member is LAEM with SIMON 128/128 as the underlying block cipher.

	underlying cipher	key size	block size	nonce size
	E_K	k	n	n
primary recommend	SIMON 128/128	128	128	128
	SIMON 128/192	192	128	128
	SIMON 128/256	256	128	128

Table 1.1: Recommended parameters of LAEM-SIMON

In the encryption/authentication procedure, the scheme accepts a 128-bit nonce, a 128/192/256-bit master key K, a message M, an associated data A, and outputs the ciphertext C. In the decryption/verification procedure, it accepts a 128-bit nonce, a 128/192/256-bit key K, a ciphertext C, an associated data A, and returns the decrypted message M if the verification succeeds or $\perp$ otherwise.

LAEM has both online-encryption and online-decryption, and it is also parallelizable. The confidentiality and integrity of the scheme is provable secure, assuming the underlying block cipher is a strong PRP under the OAE2 measure. The scheme is nonce-respect, which means the nonce should be unique and not repeated under the same key.

Chapter 2

Specification

2.1 Preliminaries

2.1.1 Symbols

At first, we list some symbols and notations which will be used.

Symbols	Descriptions
0^b	the string of b successive “0”
1^b	the string of b successive “1”
$\{0,1\}^b$	the set containing all b-bit strings
$\{0,1\}^*$	the set containing all strings
$\|A\|$	the bit length of string A
$A[i]$	the ith bit of A
$A[i,\ldots,j]$	the bits of $A[i]A[i+1]\cdots A[j]$, $i<j$
$A\|\|B$	the concatenation of strings A and B
$[i]_b$	the binary representation of i as b-bit string
$a \twoheadleftarrow A$	select an element a from set A uniformly at random

2.1.2 Linear Mix function ρ

For $A, B \in \{0,1\}^n$, let $A \cdot B$, or simply AB, denote the multiplication of A and B in Galois field $GF(2^n)$, and let $A \oplus B$ denote bitwise xor of A and B. A linear mix function ρ is defined as $\rho(x, y) = (2x \oplus y, 3x \oplus y)$. It takes two n-bit strings x and y as input and gives two n-bit strings x' and y' as output. Its inverse function ρ^{-1} takes x and y' as input and gives x' and y as output, which can be represented as $\rho^{-1}(x, y') = (x \oplus y', 3x \oplus y')$. Here we use field multiplication by 2 and 3, which can be represented as simple shift and xor operations.

The linear mix function ρ and its inverse ρ^{-1} are illustrated in Fig. 2.1

2.1.3 Segmented-AE schemes

A *segmented-AE scheme* is a triplet $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with a *nonce space* $\mathcal{N}$, an *associated-data space* $\mathcal{H}$ and a *state space* $\mathcal{S}$. Let $\mathcal{M} = \{0,1\}^*$ and $\mathcal{C} = \{0,1\}^*$ be *message space* and *ciphertext space* respectively, and the *key space* $\mathcal{K}$ is a

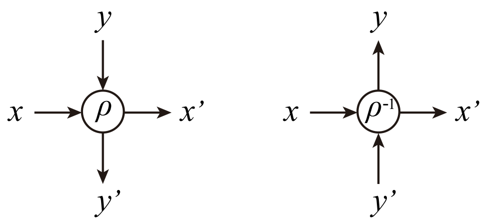

Figure 2.1: The linear mix function ρ and its inverse function ρ^{-1}

nonempty set. Let encryption $\mathcal{E} = (\mathcal{E}.\mathrm{init}, \mathcal{E}.\mathrm{next}, \mathcal{E}.\mathrm{last})$ and decryption $\mathcal{D} = (\mathcal{D}.\mathrm{init}, \mathcal{D}.\mathrm{next}, \mathcal{D}.\mathrm{last})$, and then the components of $\mathcal{E}$ and $\mathcal{D}$ are defined as follows:

$$\begin{array}{ll} \mathcal{E}.\mathrm{init} : \mathcal{K} \times \mathcal{N} \to \mathcal{S} & \mathcal{D}.\mathrm{init} : \mathcal{K} \times \mathcal{N} \to \mathcal{S} \\ \mathcal{E}.\mathrm{next} : \mathcal{S} \times \mathcal{H} \times \mathcal{M} \to \mathcal{C} \times \mathcal{S} & \mathcal{D}.\mathrm{next} : \mathcal{S} \times \mathcal{H} \times \mathcal{C} \to (\mathcal{M} \times \mathcal{S}) \cup \{\perp\} \\ \mathcal{E}.\mathrm{last} : \mathcal{S} \times \mathcal{H} \times \mathcal{M} \to \mathcal{C} & \mathcal{D}.\mathrm{last} : \mathcal{S} \times \mathcal{H} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\} \end{array}$$

For a segmented-AE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, if there exists a constant τ such that $|C_i| = |M_i| + \tau$ for every $i < l$, where l is the number of segments in $\mathcal{M}$, we call τ the *segment-expansion* of Π.

If the state space of Π is finite and there is a constant ω such that $\mathcal{E}$.next ($\mathcal{D}$.next) and $\mathcal{E}$.last ($\mathcal{D}$.last) use at most ω bits of working memory, we say that Π has *online-encryption* (*online-decryption*). The scheme Π is *online* if it has both *online-encryption* and *online-decryption*.

2.2 Specification of LAEM

The inputs to LAEM are a variable-length plaintext M, a variable-length associated data A, a fixed-length nonce N, and a fixed-length key K. The length of M and A is unlimited. The total number of segments in messages is at most 2^{64}. The nonce N should never be used repeatedly in different encryptions.

2.2.1 AEAD

Let E_K be a block cipher with key $K \in \mathcal{K}$, and ρ be the linear mix function defined in subsection 1.1.2. The authenticated encryption with associated data procedure of LAEM is illustrated in Fig. 2.2.

For simplicity, the encryption and decryption procedures can be described simply as follows.

$$\begin{array}{ll} \mathcal{E}.\mathrm{init} : \mathcal{K} \times \mathcal{N} \to \mathcal{S} & \mathcal{D}.\mathrm{init} : \mathcal{K} \times \mathcal{N} \to \mathcal{S} \\ \mathcal{E}.\mathrm{data} : \mathcal{S} \times \mathcal{H} \to \mathcal{S} & \mathcal{D}.\mathrm{data} : \mathcal{S} \times \mathcal{H} \to \mathcal{S} \\ \mathcal{E}.\mathrm{next} : \mathcal{S} \times \mathcal{M} \to \mathcal{C} \times \mathcal{S} & \mathcal{D}.\mathrm{next} : \mathcal{S} \times \mathcal{C} \to (\mathcal{M} \times \mathcal{S}) \cup \{\perp\} \\ \mathcal{E}.\mathrm{last} : \mathcal{S} \times \mathcal{M} \to \mathcal{C} & \mathcal{D}.\mathrm{last} : \mathcal{S} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\} \end{array}$$

where the state update functions of associated data and message are denoted as $\mathcal{E}$.data/$\mathcal{D}$.data and $\mathcal{E}$.next/$\mathcal{D}$.next respectively.

The n-bit nonce N is used to generate the initial state S_0^*. The associated data A is divided into n-bit strings $(A_1, \ldots, A_a)$, where $|A_i| = n$ for $1 \leq i \leq a{-}1$,

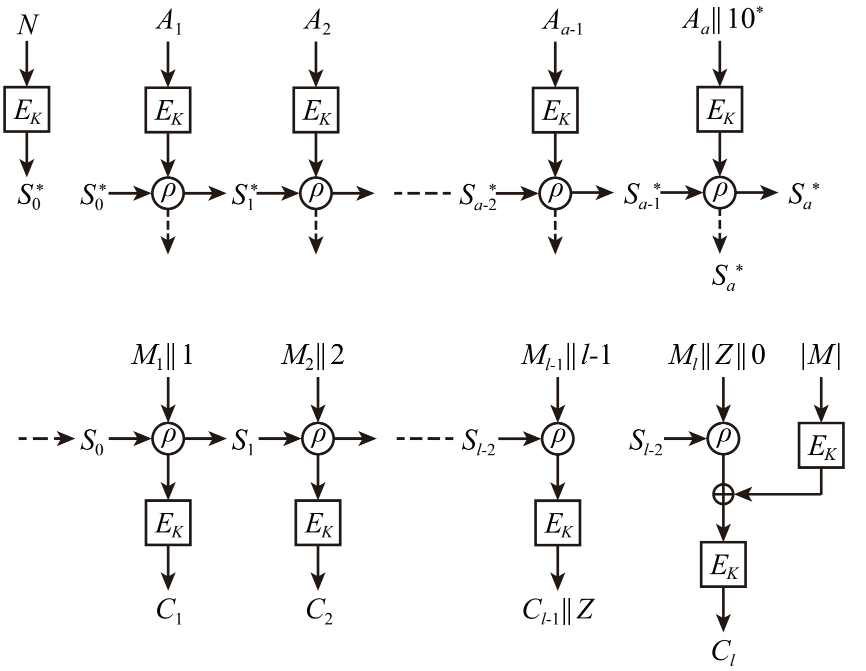

Figure 2.2: The encryption procedure of LAEM

$0 < |A_a| \leq n$. For the last string A_a, if $|A_a| = n$ we choose the output x' of ρ as S_a^* and if $|A_a| < n$ we first pad it with one bit "1" followed by $s \geq 0$ bits "0" to obtain an n-bit string and then choose the output y' of ρ as S_a^*. This state is also used as the initial state to process the message, namely $S_0 = S_a^*$. The message M is divided into m-bit segments $(M_1, \ldots, M_l)$, where $|M_i| = m$ for $1 \leq i \leq l-1$, $0 < |M_l| \leq m$. Then the encryption procedure contains the following four steps:

1. $\mathcal{E}$.init: generate an initial state with the nonce.

2. $\mathcal{E}$.data: process the associated data and update the state with (S_{i-1}^*, A_i). This step is called iteratively until $i = a$.

3. $\mathcal{E}$.next: generate a ciphertext C_i and update the state with $(S_{i-1}, M_i || [i]_{n-m})$, where $|C_i| = |M_i| + \tau$. This step is called iteratively until $i = l-2$.

4. $\mathcal{E}$.last: generate ciphertext (C_{l-1}, C_l) with (S_{l-2}, M_{l-1}, M_l), where $|C_{l-1}| + |C_l| = |M_{l-1}| + |M_l| + 2\tau$.

In the scheme, $[i]_{n-m}$ denotes the $(n-m)$-bit representation of a counter i, which is used for authenticity. Since a valid M_i' obtained in decryption should be equal to the concatenation of m-bit M_i and $(n-m)$-bit counter i, we can take the condition of $M_i'[m+1, \ldots, n] = [i]_{n-m}$ as verification.

Notice that the step $\mathcal{E}$.last deals with two segments, and the overall segment-expansion is 2τ. To solve this problem, we take $M^* = M_{l-1} \| M_l$ as a segment with variable length, where $m < |M^*| \leq 2m$. The segment-expansion of M^*

is 2τ, while the other segment-expansions still be τ. Specifically, we define (τ,ω)-expanding segmented-AE schemes for which $|C_i| = |M_i| + \tau$ for $i < l$, and $|C_l| = |M_l| + \omega$. Hence, LAEM is a $(\tau, 2\tau)$-expanding segmented-AE scheme. In later security analysis, we will discuss whether the variable-length expansion is necessary for building blockcipher-based OAE2-scheme.

The encryption and decryption procedures of LAEM are also described as the following pseudo-code in Fig. 2.3.

```
$\mathcal{E}(K,N,A,M)$
parse $A$ as $(A_1,\dots,A_a)$ with $|A_i| = n$
    for $i = 1,\dots,a-1$ and $0 < |A_a| \le n$
parse $M$ as $(M_1,\dots,M_l)$ with $|M_i| = m$
    for $i = 1,\dots,l-1$ and $0 < |M_l| \le m$
$S_0^* \leftarrow E_K(N)$
for $i = 1,...,a-1$ do
    $Y_i \leftarrow E_K(A_i)$
    $S_i^* \leftarrow 2 \cdot S_{i-1}^* \oplus Y_i$
$Y_a \leftarrow E_K(A_a \| 10^{n-|A_a|-1})$
if $|A_a| = n$ then $Y_a \leftarrow E_K(A_a)$
    $S_a^* \leftarrow 2 \cdot S_{a-1}^* \oplus Y_a$
    else $Y_a \leftarrow E_K(A_a \| 10^{n-|A_a|-1})$
    $S_a^* \leftarrow 3 \cdot S_{a-1}^* \oplus Y_a$
$S_0 \leftarrow S_a^*$
for $i = 1,...,l-2$ do
    $X_i \leftarrow 3 \cdot S_{i-1} \oplus (M_i \| [i]_{n-m})$
    $S_i \leftarrow 2 \cdot S_{i-1} \oplus (M_i \| [i]_{n-m})$
    $C_i \leftarrow E_K(X_i)$
$X_{l-1} \leftarrow 3 \cdot S_{l-2} \oplus (M_{l-1} \| [l-1]_{n-m})$
$C^* \leftarrow E_K(X_{l-1})$
$C_{l-1} \leftarrow C^*[1,\dots,n-m+|M_l|]$
$Z \leftarrow C^*[n-m+|M_l|+1,\dots,n]$
$M^* \leftarrow M_l \| Z \| [0]_{n-m}$
$X_l \leftarrow 3 \cdot S_{l-2} \oplus M^* \oplus E_K(|M|)$
$C_l \leftarrow E_K(X_l)$
return $(C_1,\dots,C_{l-1},C_l)$
```

```
$\mathcal{D}(K,N,A,C)$
parse $A$ as $(A_1,\dots,A_a)$ with $|A_i| = n$
    for $i = 1,\dots,a-1$ and $0 < |A_a| \le n$
parse $C$ as $(C_1,\dots,C_l)$ with $|C_i| = n$
    for $i = 1,\dots,l-2,l$ and $|C_{l-1}| \le n$
$S_0^* \leftarrow E_K(N)$
for $i = 1,...,a-1$ do
    $Y_i \leftarrow E_K(A_i)$
    $S_i^* \leftarrow 2 \cdot S_{i-1}^* \oplus Y_i$
$Y_a \leftarrow E_K(A_a \| 10^{n-|A_a|-1})$
if $|A_a| = n$ then $Y_a \leftarrow E_K(A_a)$
    $S_a^* \leftarrow 2 \cdot S_{a-1}^* \oplus Y_a$
    else $Y_a \leftarrow E_K(A_a \| 10^{n-|A_a|-1})$
    $S_a^* \leftarrow 3 \cdot S_{a-1}^* \oplus Y_a$
$S_0 \leftarrow S_a^*$
for $i = 1,...,l-2$ do
    $X_i \leftarrow E_K^{-1}(C_i)$
    $M_i' \leftarrow 3 \cdot S_{i-1} \oplus X_i$
    if $M_i'[m+1,\dots,n] \ne [i]_{n-m}$ then return $\perp$
        else $M_i \leftarrow M_i'[1,\dots,m]$
    $S_i \leftarrow 2 \cdot S_{i-1} \oplus M_i'$
$X_l \leftarrow E_K^{-1}(C_l)$
$M_l' \leftarrow 3 \cdot S_{l-2} \oplus X_l \oplus E_K(|C| - l \cdot (n-m))$
if $M_l'[m+1,\dots,n] \ne [0]_{n-m}$ then return $\perp$
    else $M_l \leftarrow M_l'[1,\dots,|C_{l-1}| - (n-m)]$
    $Z \leftarrow M_l'[|C_{l-1}| - (n-m) + 1,\dots,m]$
$X_{l-1} \leftarrow E_K^{-1}(C_{l-1} \| Z)$
$M_{l-1}' \leftarrow 3 \cdot S_{l-2} \oplus X_{l-1}$
if $M_{l-1}'[m+1,\dots,n] \ne [l-1]_{n-m}$ then return $\perp$
    else $M_{l-1} \leftarrow M_{l-1}'[1,\dots,m]$
return $(M_1,\dots,M_{l-1},M_l)$
```

Figure 2.3: The pseudo-code of LAEM

2.2.2 Short message

For extremely short message whose length is no more than m-bit, special operation should be made to the Step $\mathcal{E}$.last. For this case, the encryption procedure is illustrated in Fig. 2.4.

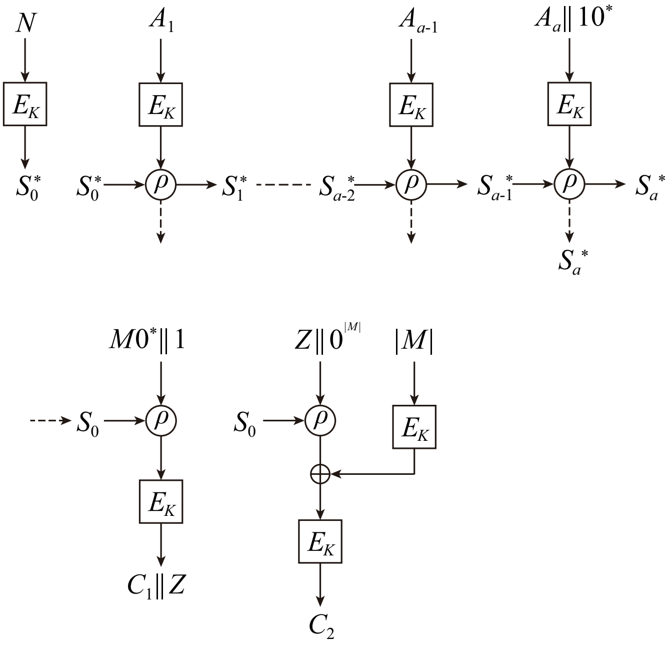

Figure 2.4: The encryption procedure of LAEM for short message with $|M| \leq m$

In this case, Steps of $\mathcal{E}$.init and $\mathcal{E}$.data remain unchanged, and the Step of $\mathcal{E}$.next will be omitted. For the Step of $\mathcal{E}$.last, the message is first padded with $(m - |M|)$-bit "0" to get an m-bit string $M0^*$. Then generate ciphertext C_1 with $(S_0, M0^*||[1]_{n-m})$, where $|C_1| = |M|$. At last, generate ciphertext C_2 with $(S_0, Z||0^{|M|}, [|M|]_n)$. Note that the procedure is similar with section 1.2.1. Details of the Step $\mathcal{E}$.last for short message with $|M| \leq m$ can also be found in the pseudo-code in Fig. 2.5

```
$S_0 \leftarrow S_a^*$

$X_1 \leftarrow 3 \cdot S_{i-1} \oplus (M\|0^{n-|M|-1}1)$
$C^* \leftarrow E_K(X_1)$
$C_1 \leftarrow C^*[1, |M|]$
$Z \leftarrow C^*[|M| + 1, n]$
$M^* \leftarrow Z\|0^{|M|}$
$X_2 \leftarrow 3 \cdot S_0 \oplus M^* \oplus E_K(|M|)$
$C_2 \leftarrow E_K(X_2)$
**return** $(C_1, C_2)$
```

```
$S_0 \leftarrow S_a^*$

$X_2 \leftarrow E_K^{-1}(C_2)$
$M_2' \leftarrow 3 \cdot S_0 \oplus X_2 \oplus E_K(|C| - n))$
**if** $M_2'[2n - |C| + 1, n] \neq 0^{|C|-n}$ **then return** $\perp$
    **else** $Z \leftarrow M_2'[1, 2n - |C|]$
$X_1 \leftarrow E_K^{-1}(C_1\|Z)$
$M' \leftarrow 3 \cdot S_0 \oplus X_1$
**if** $M'[|C| - n + 1, n] \neq 0^{2n-|C|-1}1$ **then return** $\perp$
    **else** $M \leftarrow M'[1, |C| - n]$
**return** $M$
```

Figure 2.5: The pseudo-code of $\mathcal{E}$.last for short message with $|M| \leq m$

2.2.3 The underlying block cipher

Considering the requirements of lightweight AEAD algorithm, we choose the lightweight block cipher SIMON[2] as the underlying block cipher E_K of LAEM.

SIMON is a family of lightweight block ciphers designed to be extremely small in hardware. It supports a variety of block and key sizes. According to the definition in [2], the SIMON block cipher with an n-bit word (and hence a $2n$-bit block) and an m-word (mn-bit) key is denoted SIMON $2n/mn$, where n is required to be 16, 24, 32, 48, or 64. In order to satisfy the requirements of key length and nonce length of AEAD algorithm, we choose SIMON with 128-bit block size and 128/192/256-bit key sizes as the underlying block cipher. Specifically, parameters of the three versions are listed in Table 2.1.

E_K	block size $2n$	key size mn	word size n	key words m	const seq	rounds
SIMON 128/128	128	128	64	2	z_2	68
SIMON 128/192	128	192	64	3	z_3	69
SIMON 128/256	128	256	64	4	z_4	72

Table 2.1: Parameters of the underlying block cipher

Each instance of SIMON uses the familiar Feistel structure. For SIMON $2n$, the round function is defined by

$$R_k(x, y) = (y \oplus f(x) \oplus k, x),$$

$$f(x) = (S^1 x \ \& \ S^8 x) \oplus S^2 x$$

where S^j denotes left circular shift by j bits, and & denotes bitwise AND. The n-bit round key k is generated by key schedules.

The SIMON key schedules employ five sequences of 1-bit round constants denoted as $z_0, \ldots z_4$. Each of these sequences is defined in terms of one of the following period 31 sequences:

$$u = u_0 u_1 u_2 \ldots = 1111101000100101011000011100110 \ldots,$$

$$v = v_0 v_1 v_2 \ldots \ = 1000111011111001001100001011010 \ldots,$$

$$w = u_0 w_1 w_2 \ldots = 1000010010110011111000110111010 \ldots,$$

For the three versions of SIMON used as our underlying block cipher, the constant sequences z_2, z_3, and z_4 have period 62 and are formed by computing the bitwise XOR of the period 2 sequence $t = t_0 t_1 t_2 \ldots = 01010101 \ldots$ with u, v, and w, respectively.

Let $c = 2^n - 4 = \texttt{0xff} \cdots \texttt{fc}$. For SIMON $2n$ with m key words $(k_{m-1}, \ldots, k_0)$ and constant sequence z_j, round keys are generated by

$$k_{i+m} = \begin{cases} c \oplus (z_j)_i \oplus k_i \oplus (I \oplus S^{-1}) S^{-3} k_{i+1}, & \texttt{if } m = 2 \\ c \oplus (z_j)_i \oplus k_i \oplus (I \oplus S^{-1}) S^{-3} k_{i+2}, & \texttt{if } m = 3 \\ c \oplus (z_j)_i \oplus k_i \oplus (I \oplus S^{-1})(S^{-3} k_{i+1} \oplus k_{i+1}), & \texttt{if } m = 4 \end{cases}$$

for $0 \leq i < T - m$, where T is the number of rounds. Note that key words k_0 to k_{m-1} are used as the first m round keys; they are loaded into the shift registers with k_0 on the right and k_{m-1} on the left.

For the complete specification of SIMON one can refer to [2] for more details.

2.2.4 LAEM-SIMON and its security claims

Based on the description of LAEM and underlying block cipher in the former sections, here we give the final specification of lightweight authenticated encryption with associated data algorithm LAEM-SIMON.

For the linear mix function $\rho(x, y) = (2x \oplus y, 3x \oplus y)$, the multiplication is defined in Galois field GF(2^{128}) with primitive polynomial $p(x) = x^{128} + x^7 + x^2 + x + 1$.

The algorithm has 4 parameters: key length, block length, segment length, nonce length. Each parameter is an integer number of bytes. The key length can be 16/24/32 bytes (128/192/256-bit). The block length is 16 bytes (128-bit). The segment length is 8 bytes (64-bit). The nonce length is 16 bytes (128-bit). The recommended parameter sets for LAEM-SIMON are as follows.

underlying cipher	key size	block size	segment size	nonce size
E_K	k	n	m	n
SIMON 128/128	128	128	64	128
SIMON 128/192	192	128	64	128
SIMON 128/256	256	128	64	128

Table 2.2: Recommended parameters of LAEM-SIMON

For the AEAD algorithm LAEM-SIMON, the security claim of confidentiality is expected to be the length of key. For the security claim of integrity, in the scheme a valid M_i' obtained in decryption should be equal to the concatenation of m-bit M_i and $(n-m)$-bit counter i, and hence we can use $[i]_{n-m}$ as verification. Therefore, for a message M which is divided into m-bit segments $(M_1, \ldots, M_l)$, where $|M_i| = m$ for $1 \leq i \leq l-1$, $0 < |M_l| \leq m$, the tag length should be $(n-m)l = 64l$-bit. On the other hand, for short message with $|M| \leq m$, the tag length should be n-bit.

Chapter 3

Design Rationale

3.1 AE Mode

First of all, we explain the design strategy of the authenticated encryption mode of LAEM. Considering the advantages of block cipher, such as low cost, high efficiency, mature analysis, and widespread applications, we try to build a blockcipher-based OAE2 scheme.

A general model of OAE2 scheme is described in Figure 3.1. In the model, a blockcipher-based basic operation maps (S_{i-1}, M_i) to (S_i, C_i). The operation consists of a linear mix function and an underlying block cipher E_k. First, the linear mix function maps a pair of input (S_{i-1}, M_i) to (S_i, M_i'). Then, M_i' is encrypted by the block cipher E_k to get the ciphertext C_i and the intermediate state is updated as S_i.

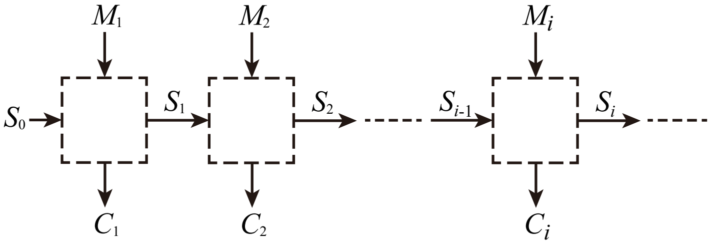

Figure 3.1: A general model of OAE2 scheme

In order to reduce the implementation cost, we build the basic operation with only one call of the underlying block cipher and one call of a linear mix function. For the linear mix function, we denote it as $\rho(x, y) = (ax \oplus by, cx \oplus dy)$. The input of underlying block cipher is formalized as $cS_i \oplus dM_i$, or $ca^{i-1}S_0 \oplus \sigma_{j=1}^{i-1} a^{i-j-1} bcM_j \oplus dM_i$, where the multiple of S_0 is a mask. To avoid collisions in the inputs of underlying block cipher, we try to generate distinct masks. The powering-up construction proposed by Rogaway[20] gives us an efficient way to produce many distinct masks as $2^\alpha 3^\beta 7^\gamma L$ from one secret value $L = E_K(0)$. Based on this construction, we choose $\rho(x, y) = (2x \oplus y, 3x \oplus y)$ with respect to a polynomial which makes the values of mask δ distinct from each other, where $\delta = 2^\alpha 3^\beta S_0$ for varying indices of α and β. The polynomial should satisfy two requirements: (1) it needs to be primitive; (2) $\log_2 3$ is "huge". For the state size $n = 128$, the polynomial $p(x) = x^{128} + x^7 + x^2 + x + 1$ satisfies the requirements, and makes the values of $2^\alpha 3^\beta$ all distinct and not equal to 1 for any $\alpha \in [-2^{115}, 2^{115}]$ and any $\beta \in [-2^{10}, 2^{10}]$. Note that for other state size n, we can still find other suitable polynomials satisfying these requirements.

3.2 AE with Associated Data

The problem of handling associated data with AE scheme was formally proposed by Rogaway [19]. It is mainly aiming for the requirements of binding some cleartext datas, such as an IP address, to a ciphertext. In an authenticated encryption with associated data scheme (denoted as AEAD), the associated data is normally treated as a header for purpose of routing or message parsing (CCM[11], GCM[12], SpongeWrap[3], and ALE[6]). Usually, the scheme generates an intermediate state based on a nonce and an associated data, and then the state will be used in encrypting plaintext (or decrypting ciphertext).

In the definition of OAE2, the associated data is partitioned and provided with each plaintext (or cipheretxt) segment. The number of associated data segments must equal to the number of the corresponding message segments. Moreover, for the situation of encrypting a long message with a short associated data, such as a message with an IP address, partitioning and padding the associated data into segments as long as message segments results in unnecessary redundant. In particular, when the associated data needs to be verified before releasing messages, an OAE2 scheme has to store all messages before the last associated data segment is verified.

When encrypting long messages with long associated data, an OAE2 scheme partitions both the message and the associated data into segments, and processes with a message segment and an associated data segment each time calling the basic operation. To keep the scheme online, which means that each calling uses at most ω bits of working memory, the segment size of message and associated data should be small enough to satisfy the online memory requirement.

This seems unreasonable compared to the way of processing associated data in the initial procedure before receiving messages. Therefore, we choose to take associated data as a header of message and process it before the message input.

3.3 The Underlying Block Cipher

According to the security evaluation of LAEM, the confidentiality and integrity of the scheme is provable secure, assuming the underlying block cipher is a strong PRP under the OAE2 measure. Moreover, based on the design requirements, the algorithm is supposed to perform efficiently in constrained environments. Especially, components with significant third-party analysis are favorable. Therefore, instead of designing a new lightweight block cipher hastily, we adopt the extremely lightweight block cipher SIMON [2] as the underlying block cipher E_K of LAEM for instance.

The SIMON families of lightweight block ciphers were designed by researchers from National Security Agency of the USA and published formally in 2013 [2]. SIMON is designed to be extremely small in hardware, and also flexible across a range of platforms. The encryption procedure of SIMON only uses basic operations such as XOR, bitwise AND and rotation. Hence, compared to other lightweight block ciphers, it has very competitive performance, for instance, compactness in hardware and small memory footprint.

SIMON supports a variety of 10 versions with block size varying from 32-bit to 128-bit and key size varying from 64-bit to 256-bit. However, according to the submission requirements about key length and nonce length, together with the security claims of our scheme, we only choose SIMON with 128-bit block size and 128/192/256-bit key sizes as the underlying block cipher.

Since its publication, SIMON has attracted a lot of analytic attention because of its very simple and novel design, and lack of security analysis in the design document. Through years of security evaluation in the public, a large number of cryptanalytic results have been reported in the literature. The attacks have covered most of the known cryptanalytic techniques against block cipher, including differential attack, linear attack, impossible differential attack, zero-correlation linear attack, integral attack and so on. According to the results of third-party analysis up to now, there are no feasible attacks on full round SIMON, and it surely maintains enough security margin. Therefore, it totally fulfills the security requirements of underlying block cipher used in LAEM.

Chapter 4

Features

In this section, we list some main features of LAEM with respect to security and performance aspects. This scheme has many useful features, especially for processing network packets, constraint environment applications and so on.

Online. An on-line authenticated encryption algorithm can provide real-time encryption/decryption without occupying recourse for buffer. This can be very attractive for some practical application envioronments, such as constrained devices. LAEM allows both online encryption and online decryption. Each segment of decrypted plaintext, as well as encrypted ciphertext, can be released directly after computation, since the verification cost is negligible. Therefore, LAEM can offer online authenticated encryption with quite small working memory, and will not suffer from the problem of releasing unverified plaintext (RUP).

Provable Secure. LAEM is an OAE2-secure scheme when the underlying block cipher is a strong PRP under the OAE2b measure. We extend OAE2 measure to adapt our variable-length segment-expansion OAE2 scheme which calls the underlying component only one time in both $\mathcal{E}$.next and $\mathcal{E}$.last.

Fully Parallelizable. LAEM calls the underlying block cipher once per m-bit plaintext segment and it is fully parallelizable in both encryption and decryption procedures. Therefore, LAEM is suitable not only for resource constrained environments, but also for high-performance parallel implementations on modern general purpose CPUs and dedicated hardwares.

Maximum Message Length. The message length is irrelevant to the security bound of LAEM, which allows magnitude data to be processed per key. Considering the version of LAEM with 128-bit block size and 64-bit segment length, messages can be partitioned up to 2^{63} segments. Therefore, the amount of data processed per key is up to 2^{69} bits.

Flexible Parameter Selection. LAEM is a blockcipher-based OAE2 scheme. The segment size m determines the trade-off between security level and computation efficiency. Users can choose appropriate value of segment size m according to the implementation requirements. When m is close to the block size n, LAEM achieves high efficiency, nearly rate-1. When m is close to $n/2$, LAEM achieves high integrity security level in the cost of being nearly rate-2. According to the results of security analysis, the integrity security with $m < n/2$ will be not higher than that of a scheme with $m = n/2$. Therefore, we recommend the value of m to be chosen from $[n/2, n]$.

For example, when $m = n/2$ the integrity of LAEM against generic attacks is upper bounded by $2^{n/2}$, and the processing efficiency is rate-2, which means the scheme requires twice block cipher calls per n-bit data. When $m = 2n/3$, the integrity against generic attacks is upper bounded by $2^{n/3}$, and the processing efficiency is rate-1.5, which means the scheme requires one n-bit block cipher call per $2n/3$-bit data.

Extendibility of Underlying Components. LAEM is mainly built based on an underlying block cipher and a linear mix function ρ. By choosing different fields and corresponding polynomials $p(x)$, we can build LAEM type schemes with different segment lengths and block sizes. Taking an n-bit block cipher as the underlying component and an irreducible polynomial $p(x)$ in Galois field $\mathrm{GF}(2^n)$ which satisfies the requirements discussed in Section 3.2, we can build a LAEM type scheme flexibly. The scheme will use about n-bit working memory in both encryption and decryption, where n is between 32 and 512, determined by the underlying block cipher. These make LAEM scheme adjustable in different environments.

Considering an environment with small working memory, such as 64 bits, neither a conventional AE scheme wrapped in CHAIN/STREAM protocol or an OAE1 scheme with intermediate tags is online for the limitation of memory. Taking a lightweight block cipher as the underlying block cipher, LAEM can still maintain online encryption and online decryption in such a small working memory environment.

Lightweight. LAEM uses only one block cipher with a single key. Moreover, the linear mix function is built with simple addition and multiplication in a Galois field. Therefore, the scheme needs only a little bit more area costs than the underlying lightweight block cipher.

One-Pass. Only one pass through the plaintext is required to provide both integrity and confidentiality. This allows on-the-fly encryption without storing previous ciphertext blocks.

One-Key. Only one single key is required to process both plaintext and associated data, which achieves the minimum key number.

Chapter 5

Implementation Evaluation

5.1 Hardware Performances

The design of LAEM aims at lightweight authenticated encryption scheme, and tries to provide confidentiality and integrity protections for constraint environment applications. The scheme is one-key, one-pass, and fully parallelizable. Moreover, the underlying block cipher adopted is extremely compact in hardware.

In the LAEM scheme, for each segment of message and associated data only around one basic operation needs to be called. For the basic operation, except the underlying block cipher E_k, a linear mix function ρ is also applied. The linear mix function is defined as $\rho(x, y) = (2x \oplus y, 3x \oplus y)$, where the multiplication is defined in field $GF(2^{128})$ with primitive polynomial $p(x) = x^{128} + x^7 + x^2 + x + 1$. By choosing appropriate and simple multiplication parameters, all of the multiplications can be implemented by a small amount of shift and XOR operations, which cost little in hardware. Moreover, the message segments are padded with counter, which can be implemented very efficiently in hardware. Therefore, the main hardware implementation cost will be dominated by the underlying block cipher.

For the instance underlying block cipher, we adopt the extremely compact lightweight block cipher families SIMON with 128-bit block size and 128/192/256-bit key sizes. According to the hardware performances evaluated by the designers, three key sizes versions of SIMON 128 costs 1234/1508/1782 GE in the area-minimizing implementations respectively (quoted from Table 6.1 in [2]).

5.2 Software Performances

Similar to the analysis of hardware performances, the scheme LAEM can be implemented efficiently in various software platforms. Since LAEM satisfies advantageous features such as on-line, one-pass, and one-key, the scheme achieves minimum calls of underlying block cipher and low memory occupations. Moreover, SIMON is also designed to be flexible across a range of software platforms. According to the software performances evaluated by the designers, three key sizes versions of SIMON 128 can achieve 333/335/353 cycles/byte respectively in high-throughput implementations on 8-bit microcontrollers (quoted from Table

7.1 in [2]). Moreover, by utilizing the high-speed SSE instructions, they can also be implemented efficiently on 32/64-bit processors, which achieve about 7.5/7.7/8.0 cycles/byte respectively (quoted from Table A.1 in [2]).

Chapter 6

Security Evaluation

6.1 Provable Security

6.1.1 Definition of security of OAE2 schemes

We use OAE2b measure to quantify the advantage that an adversary gets in attacking an OAE2 scheme [20]. OAE2b is a string-oriented formulation, and employs more realistic accounting of the adversary's actual resource expenditure than OAE2a and OAE2c.

Notice that LAEM is not a constant-segment-expansion OAE2 scheme, and LAEM's corresponding ideal model and OAE2b measure will be little different from the ones in [15]. We denote (τ, ω)-expanding segmented-AE scheme Π such that $|C_i| = |M_i| + \tau$ for $i < l$, and $|C_l| = |M_l| + \omega$.

Let $\mathrm{Inj}(\tau)$ denote the set of all τ-expanding injective functions $f : \{0,1\}^* \to \{0,1\}^*$, where $|f(x)| = |x| + \tau$. Define $\mathrm{IdealOAE}(\tau, \omega)$ as follows:

```
for $m \in \mathbb{Z}^+, N \in \{0,1\}^*, \boldsymbol{M} \in (\{0,1\}^*)^{m-1}$ do
    $f_{N,\boldsymbol{M},0} \twoheadleftarrow \mathrm{Inj}(\tau)$
    $f_{N,\boldsymbol{M},1} \twoheadleftarrow \mathrm{Inj}(\omega)$
for $m \in \mathbb{Z}^+, \boldsymbol{M} \in (\{0,1\}^*)^m, \sigma \in \{0,1\}$ do
    $F(N, M, \sigma) \leftarrow (f_{N,\Lambda,0}(M_1), f_{N,M_1,0}(M_2), \ldots, f_{N,M_1,\ldots,M_{m-2},0}(M_{m-1}),$
        $f_{N,M_1,\ldots,M_{m-1},\sigma}(M_m))$
ret $F$
```

Figure 6.1 defines games $\mathrm{Real2B}_\Pi$ and $\mathrm{Ideal2B}_\Pi$ for a (τ, ω)-expanding segmented AE scheme Π. Given an adversary $\mathcal{A}$ with oracles Enc and Dec determined by these games, we define the adversary's distinguishing advantage as follows:

$$\mathbf{Adv}_{\Pi}^{\mathrm{oae2b}}(\mathcal{A}) = |\Pr[\mathcal{A}^{\mathrm{Real2B}_\Pi} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathrm{Ideal2B}_\Pi} \Rightarrow 1]|$$

```
Real2B$_\Pi$
proc initialize
$I, J \leftarrow 0$
$K \twoheadleftarrow \mathcal{K}$

proc Enc.init($N$)
if $N \notin \mathcal{N}$ then ret $\perp$
$I \leftarrow I + 1$
$S^{(I)} \leftarrow \mathcal{E}.\text{init}(K, N)$
ret $I$

proc Enc/Dec.data($i, A$)
if $i \notin [1..I]$ or $S^{(i)} = \perp$ then ret $\perp$
$S^{(i)} \leftarrow \mathcal{E}.\text{data}(S^{(i)}, A)$

proc Enc.next($i, M$)
if $i \notin [1..I]$ or $S^{(i)} = \perp$ then ret $\perp$
$(C, S^{(i)}) \leftarrow \mathcal{E}.\text{next}(S^{(i)}, M)$
ret $C$

proc Enc.last($i, M$)
if $i \notin [1..I]$ or $S^{(i)} = \perp$ then ret $\perp$
$C \leftarrow \mathcal{E}.\text{last}(S^{(i)}, M)$
$S \leftarrow \perp$
ret $C$

proc Dec.init($j, C$)
if $N \neq \mathcal{N}$ then ret $\perp$
$I \leftarrow I + 1$
$S^{(I)} \leftarrow \mathcal{E}.\text{init}(K, N)$
ret $I$

proc Dec.next($j, C$)
if $j \notin [1..J]$ or $S'^{(j)} = \perp$ then ret $\perp$
$(M, S'^{(j)}) \leftarrow \mathcal{D}.\text{next}(S'^{(j)}, C)$
ret $M$

proc Dec.last($j, C$)
if $i \notin [1..J]$ or $S'^{(j)} = \perp$ then ret $\perp$
$M \leftarrow \mathcal{D}.\text{last}(S'^{(j)}, C)$
$S'^{(j)} \leftarrow \perp$
ret $M$
```

```
Ideal2B$_\Pi$
proc initialize
$I, J \leftarrow 0$
$F \twoheadleftarrow \text{IdealOAE}(\tau, 2\tau)$

proc Enc.init($N$)
if $N \notin \mathcal{N}$ then ret $\perp$
$I \leftarrow I + 1$
$N^{(I)} \leftarrow N; M \leftarrow \epsilon$
ret $I$

proc Enc/Dec.data($i, A$)
if $i \notin [1..I]$ or $A^{(i)} = \perp$ then ret $\perp$
$A^{(i)} \leftarrow A^{(i)} \| A$

proc Enc.next($i, M$)
if $i \notin [1..I]$ or $M^{(i)} = \perp$ then ret $\perp$
$M^{(i)} \leftarrow M^{(i)} \| M; \lambda \leftarrow \lceil |M^{(i)}|/m \rceil$
$C \leftarrow F(N^{(i)}, A^{(i)}, M^{(i)}, 0)$
ret $C_\lambda$

proc Enc.last($i, M$)
if $i \notin [1..I]$ or $S^{(i)} = \perp$ then ret $\perp$
$M^{(i)} \leftarrow M^{(i)} \| M; \lambda \leftarrow \lceil |M^{(i)}|/m \rceil$
$C \leftarrow F(N^{(i)}, A^{(i)}, M^{(i)}, 1)$
$M^{(i)} \leftarrow \perp$
ret $C_\lambda$
proc Dec.init($j, C$)

if $N \neq \mathcal{N}$ then ret $\perp$
$I \leftarrow I + 1$
$N^{(I)} \leftarrow N; C \leftarrow \epsilon$
ret $I$

proc Dec.next($j, C$)
if $j \notin [1..J]$ or $C^{(j)} = \perp$ then ret $\perp$
$C^{(j)} \leftarrow C^{(j)} \| C; \lambda \leftarrow \lceil |C^{(j)}|/m \rceil$
if $\exists M$ s.t. $F(N^{(j)}, A^{(j)}, M, 0) = C^{(j)}$
    then ret $M_\lambda$
    else $C^{(j)} \leftarrow \perp$
ret $\perp$

proc Dec.last($j, C$)
if $j \notin [1..J]$ or $C^{(j)} = \perp$ then ret $\perp$
$C^{(j)} \leftarrow C^{(j)} \| C; \lambda \leftarrow \lceil |C^{(j)}|/m \rceil$
if $\exists M$ s.t. $F(N^{(j)}, A^{(j)}, M, 1) = C^{(j)}$
    then ret $M_\lambda$
    else $C^{(j)} \leftarrow \perp$
ret $\perp$
```

Figure 6.1: OAE2b measure

6.1.2 Main Results

We now prove the OAE2 security of LAEM.

Theorem 1. *Let* Π *be LAEM with the underlying blockcipher* E*. For any* *adversary $\mathcal{A}$ making at most q forward queries of total segment length no more than σ and at most q_v backward queries of total segment length no more than σ_v, with associated-data of total block length no more than α, we have*

$$\begin{aligned}\mathbf{Adv}^{\text{oae2b}}_{\Pi_E}(t, t_v, q, q_v, \sigma, \sigma_v, \alpha) \quad \leq \quad & \mathbf{Adv}^{\text{sprp}}_{E}(\sigma + \sigma_v + \alpha + 2q + 2q_v) \\ & + \frac{3\sigma^2 + 2\sigma_v^2 + 6\sigma\sigma_v + \alpha^2}{2^{n+1}}.\end{aligned}$$

Notice that for each input of Enc.last or Dec.last, we take the input as 2 segments in counting total segment length.

Proof. Our analysis is derived by game-playing argument. To make the analysis concise, we assume that an adversary always makes valid queries, which means answers of the encryption (decryption) oracle will not be used to query the decryption (encryption) oracle; and the adversary will not make queries with repeating nonce or query a segment after a message ended with an "end" flag. The situations with $\perp$ as output does not happen.

Consider games G_0-G_5 as shown in Figures 6.2-6.7 in Appendix. G_0 corresponds to the real LAEM. G_1 is identical to G_0, except that the underlying block cipher E_K is replaced by a permutation P randomly chosen in $\text{Perm}(n)$. The advantage of distinguishing G_0 and G_1 is bound by the SPRP security of E_K,

$$|\Pr[\mathcal{A}^{G_0} \Rightarrow 1] - \Pr[\mathcal{A}^{G_1} \Rightarrow 1]| \leq \mathbf{Adv}^{\text{sprp}}_{E}(t + t_v, \sigma + \sigma_v + \alpha + 2q + 2q_v).$$

In game G_2, a mapping $F : \{0,1\}^n \to \{0,1\}^n$ perfectly simulates P. When called with a fresh input X, F randomly chooses a string Y from $\{0,1\}^n$ and maps X into Y; and when X has been queried before, F finds the corresponding image Y and return. Then, $\Pr[\mathcal{A}^{G_1} \Rightarrow 1] = \Pr[\mathcal{A}^{G_2} \Rightarrow 1]$.

Game G_3 is identical to G_2, except that when the flag *bad* sets `true`, G_3 will choose a new value for T (a combination of a state and a counter i). The two games are identical-until-*bad*,

$$|\Pr[\mathcal{A}^{G_2} \Rightarrow 1] - \Pr[\mathcal{A}^{G_3} \Rightarrow 1]| \leq \Pr[\mathcal{A}^{G_2} \,\text{sets}\, bad].$$

We estimate the probability of "*bad* sets `ture`" by counting the increasing size of $\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2$ and $\text{Dom}F(\cdot)$, and bound the probability by $(2\sigma^2 + \sigma_v^2 + 4\sigma\sigma_v + 4q\sigma + 2q_v\sigma + 2\alpha\sigma - 2\sigma - 2q_v\sigma_v - \sigma_v + \alpha^2 + 2q_v^2)/2^{n+1}$.

Notice that the input of F has two parameters, a segment M and a modified state T. When T is fixed, F can be seen as an injective function about M. We try to replace $F(S \oplus (M \| [0]_{n-m}))$ by a family of injective functions $G_T(M)$, where $T \in \{0,1\}^n$ and $M \in \{0,1\}^m$. There are two special situations:

1. $T_1 = T_2$ for $T_1, T_2 \in \{0,1\}^n$, implying to $G_{T_1}(\cdot) = G_{T_2}(\cdot)$;

2. $T_1 \oplus (M_1 \| [0]_{n-m}) = T_2 \oplus (M_2 \| [0]_{n-m})$ for $(T_1, M_1), (T_2, M_2) \in \mathcal{Q}_1|_{(T,M)}$. Because $G_T(\cdot)$ is built according to P, $T_1 \oplus (M_1 \| [0]_{n-m}) = T_2 \oplus (M_2 \| [0]_{n-m})$ implies $G_{T_1}(M_1) = G_{T_2}(M_2)$. When this situation happens in queries, the adversary will know $G_{T_1}(M) = G_{T_2}(M \oplus \delta)$ for $\delta = M_1 \oplus M_2$ and all $M \in \{0,1\}^m$.

We avoid two situations by choosing suitable values of T, as we do in game G_3. Therefore, elements in $\{G_T(\cdot)\}_{T\in\mathcal{T}_1|_T}$ are independent of each other. By replacing $F(\cdot)$ by $G_{\cdot}(\cdot)$, we get the encryption part of game G_4 as show in Figure 6.5.

In the decryption of game G_3, the adversary can only get an answer not equal to false when the inequality in line 427 is false in Dec.next of game G_3, or the inequalities in lines 439 and 446 are false in Dec.last. We see that the probability of setting false inequality depends on the eligible set of M and the eligible set of C. By the method Rogaway and Shrimpton[21] used in the analysis of the indistinguishability of DAE and PRI, we simulate the Dec.next of game G_3 by choosing randomly from the set $\{0,1\}^n - \{C : (T_{j-1}^{(i)}, T_j^{(i)}, \cdot, \cdot, C) \in \mathcal{Q}_2\}$, and build the Dec.next of game G_2. Similar for the Dec.last. With the perfect simulation of G, game G_4 coincides with G_3. Thus $\Pr[\mathcal{A}^{G_3} \Rightarrow 1] = \Pr[\mathcal{A}^{G_4} \Rightarrow 1]$.

In game G_4, we have a family $\{G_T(\cdot)\}_{T\in\mathcal{Q}_1|_T}$ of independent injective functions with different values of T randomly chosen from $\{0,1\}^n$. Notice that T is a mark for an injective function $G_T(\cdot)$ that $G_T(\cdot) \neq G_{T'}(\cdot)$ for any $T' \in \mathcal{Q}_1|_T$ with $T \neq T'$. It will make no difference to change the subscript $T_{j-1}^{(i)}$ into $(N^{(i)}, A^{(i)}, M_1^{(i)}, \ldots, M_{j-1}^{(i)})$ for every $G_{T_j^{(i)}}(M_j^{(i)})$. For Enc.last, we build a new injective funtion G' with G for a valid query (T_1, T_2, M_1, M_2):

$$G'_{T_1,T_2}(M_1, M_2) = (G_{T_1}(M_1)[1, n-m+|M_2|], G_{T_2}(M_2\|G_{T_1}(M_1)[n-m+|M_2|+1, n])).$$

We change $(T_{j-1}^{(i)}, T_j^{(i)})$ into $(N^{(i)}, A^{(i)}, M_1^{(i)}, \ldots, M_{j-1}^{(i)})$ as we do to $G_{T_j^{(i)}}(M_j^{(i)})$ for every $G'_{T_{j-1}^{(i)},T_j^{(i)}}(M_j^{(i)}, M_{j+1}^{(i)})$.

By adding one bit flag to distinguish G and G', we get injective functions like $G^*_{N^{(i)},A^{(i)},M_1^{(i)},\ldots,M_{j-1}^{(i)},0}(M_j^{(i)})$ and $G^*_{N^{(i)},A^{(i)},M_1^{(i)},\ldots,M_{j-1}^{(i)},1}(M_j^{(i)}, M_{j+1}^{(i)})$. The only difference between $G^*_{N,A,M,0}$ (or $G^*_{N,A,M,1}$) and $H_{N,A,M,0} \twoheadleftarrow \text{Inj}(n-m)$ (or $H_{N,A,M,1} \twoheadleftarrow \text{Inj}(2(n-m))$) is that G^* always outputs different images, while H with different subscript values may output same images. The probability of distinguishing G^* and H by an adversary is bound by $(\sigma+\sigma_v)(\sigma+\sigma_v-1)/2^{n+1}$. We build game G_5 with H, thus

$$|\Pr[\mathcal{A}^{G_4} \Rightarrow 1] = \Pr[\mathcal{A}^{G_5} \Rightarrow 1]| \leq \frac{(\sigma+\sigma_v)(\sigma+\sigma_v-1)}{2^{n+1}}.$$

Notice that G_5 coincides with Ideal2B. We can get the adversary's distinguishing advantage by summing all these up.

$$\begin{aligned}
\mathbf{Adv}_{\Pi_E}^{\text{oae2b}}(t, t_v, q, q_v, \sigma, \sigma_v, \alpha) &\leq |\Pr[\mathcal{A}^{G_0} \Rightarrow 1] - \Pr[\mathcal{A}^{G_5} \Rightarrow 1]| \\
&\leq \mathbf{Adv}_E^{\text{sprp}}(\sigma + \sigma_v + \alpha + 2q + 2q_v) \\
&\quad + \frac{1}{2^{n+1}}(3\sigma^2 + 2\sigma_v^2 + 6\sigma\sigma_v + 4q\sigma + 2q_v\sigma \\
&\quad -2q_v\sigma_v + 2\alpha\sigma - 3\sigma - 2\sigma_v + \alpha^2 + 2q_v^2) \\
&\leq \mathbf{Adv}_E^{\text{sprp}}(\sigma + \sigma_v + \alpha + 2q + 2q_v) \\
&\quad + \frac{3\sigma^2 + 2\sigma_v^2 + 6\sigma\sigma_v + \alpha^2}{2^{n+1}}.
\end{aligned}$$

□

To compare with other schemes, we give the OAE2c security of LAEM which separately defines privacy and authenticity requirements as follows.

Theorem 2. *Let Π be LAEM with the underlying blockcipher E. For any adversary $\mathcal{A}$ making at most q forward queries of total segment length no more than σ,*

$$\mathbf{Adv}_{\Pi_E}^{\mathrm{oae2c-priv}}(t,q,\sigma) \leq \mathbf{Adv}_{E}^{\mathrm{sprp}}(t,\sigma+q) + \frac{(\sigma+q)^2}{2^n}.$$

Theorem 3. *Let Π be LAEM with the underlying blockcipher E. For any adversary $\mathcal{A}$ making at most q forward queries of total segment length no more than σ, and at most q_v backward queries of total segment length no more than σ_v,*

$$\begin{aligned}\mathbf{Adv}_{\Pi_E}^{\mathrm{oae2c-auth}}(t,t_v,q,q_v,\sigma,\sigma_v) \quad\leq\quad & \mathbf{Adv}_{E}^{\mathrm{sprp}}(t+t_v,\sigma+\sigma_v+2q+2q_v) \\ & + \frac{(\sigma+\sigma_v+q+q_v)^2}{2^n} + \frac{2q_v}{2^{n-m}}.\end{aligned}$$

6.2 The Underlying Block Cipher

Since the publication of Simon lightweight block cipher family in 2013, it has attracted many researchers' attention and a large number of security analysis results have been reported in the literature. The attacks including differential attack, linear attack, impossible differential attack, zero-correlation linear attack, integral attack and so on. In this section, we summarize the main attack results of Simon 128, which is the underlying block cipher used in LAEM.

6.2.1 Differential Attack

Differential attack is the most widely used method to evaluate the security of block ciphers. Therefore, differential attack on Simon was studied at the earliest and the publications were richest. The main results include differential characteristics of round-reduced Simon variants searched by various automatic techniques, such as threshold search [4], MILP [24], and SAT/SMT [17]. Moreover, based on these differential characteristics, Wang *et al.* [27] proposed a dynamic key-guessing technique which largely reduced the number of key guesses. Finally, they presented differential attacks on 50-round Simon 128/128, 51-round Simon 128/192, and 51-round Simon 128/256 respectively. These are the best differential attacks so far, which means three versions of Simon 128 with full rounds are all secure against differential attack.

6.2.2 Linear Attack

Linear attack is one of the most important cryptanalytic techniques and has showed great power against some block ciphers. It also provides the best attack results on Simon up to now. In [10], Chen and Wang presented improved linear attacks on all reduced versions of Simon with dynamic key-guessing technique. Specifically, the linear attacks can reach up to 49-round Simon 128/128, 51-round Simon 128/192, and 53-round Simon 128/256 respectively. These are the best cryptanalysis results on SIMON 128 under secret-key setting. Therefore, full round SIMON 128 maintains enough security margin against known attacks.

6.2.3 Impossible Differential Attack

Impossible differential attack is one of the most powerful cryptanalytic techniques against block cipher. Its main idea is to construct differentials with probability zero in order to eliminate the wrong key candidates leading to such impossible differentials. Usually, a miss-in-the-middle approach is used to construct the impossible differential distinguisher.

There have been several impossible differential attacks on various variants of SIMON. The main results about SIMON 128 contains an impossible differential attack on 22-round for all three key sizes presented in [1]. Then Boura *et al.* [8] further improved the results up to 27-round SIMON 128/128, 28-round SIMON 128/192, and 30-round SIMON 128/256.

6.2.4 Zero-correlation Linear Attack

Zero-correlation linear attack is one of the recent cryptanalytic techniques proposed by Bogdanov and Rijmen in [7]. Its main idea is to construct linear approximation with correlation zero. For the version of SIMON with 128-bit block size, the best zero-correlation linear attack was reported in [23]. First, they presented 19-round zero-correlation linear approximations of SIMON 128, and then based on the distinguisher they provided zero-correlation linear attacks on 32-round SIMON 128/192 and 34-round SIMON 128/256, respectively.

6.2.5 Integral Attack

Integral attack is a traditional cryptanalytic technique which exploits a distinguisher whose outputs have the zero-sum property with respect to a set of chosen inputs. Recently, integral attack has developed significantly since Todo [25] proposed a generalized integral property, called division property, at EUROCRYPT 2015. It becomes a powerful cryptanalytic technique against bit-based block cipher, or cipher with low algebraic degree round function.

In [25], based on the division property, Todo presented integral distinguishers of SIMON for each state size by viewing the round function as an Sbox of algebraic degree two. Their results shows that SIMON 128 has 13-round integral distinguisher. Later, Todo *et al.* further introduced bit-based division property in [26], which can treat each bit of SIMON independently. However, bit-based division property was only applicable to SIMON 32 since the time and memory complexity is computationally impractical for larger block size. Therefore, they introduced a new technique called *lazy propagation* which evaluated only a part of all propagations. This technique can evaluate the number of rounds that bit-based division property cannot find integral distinguishers. As a result, they theoretically proved that SIMON 128 do not have 29-round integral distinguishers.

6.2.6 MITM Attack

Meet-in-the-middle attack is a powerful cryptanalytic technique against block cipher, especially for cipher with simple key schedules. Moreover, in recent years meet-in-the-middle attack have developed a lot and many new techniques, such as sieve-in-the-middle[9], match box[13] etc, have been proposed.

By exploiting the weaknesses of the linear key schedules of Simon, a match box meet-in-the-middle attack on round-reduced Simon was presented in [22]. For the versions of Simon 128/192 and Simon 128/256, the match box meet-in-the-middle attack can both achieve 25-round with very low data complexity. For the version of Simon 128/128, a basic meet-in-the-middle attack is more effective which can achieve 19-round. All of these analysis results are far away from full-round.

6.2.7 Known-key Attack

Besides the security evaluation of Simon under classical secret single-key model, the resistance of Simon against known-key attacks has also been studied. Known-key attacks (also called known-key distinguishers) were introduced by Knudsen and Rijmen at ASIACRYPT 2007 [16] and have been applied to PRSEENT to get a full round known-key attack in [5]. Unlike the setting of classical secret-key model, the adversary in the known-key model knows the randomly chosen key. With the knowledge of the key, the adversary is supposed to find a non-random property that an ideal cipher should not have.

In [14], Hao *et al.* proposed known-key attacks on various versions of round-reduced Simon. Specifically, for the underlying block cipher Simon with 128-bit block used in LAEM, the known-key attack can achieve up to 63-round. Hence, even in the known-key model, full round Simon 128 still has enough security margin.

Bibliography

[1] Javad Alizadeh, Hoda A. Alkhzaimi, Mohammad Reza Aref, Nasour Bagheri, Praveen Gauravaram, Abhishek Kumar, Martin M. Lauridsen, and Somitra Kumar Sanadhya. Cryptanalysis of SIMON Variants with Connections. RFIDSec 2014, LNCS 8651, pp. 90–107, 2014.

[2] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404 (2013). `http://eprint.iacr.org/2013/404`

[3] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: single-pass authenticated encryption and other applications. In *International Workshop on Selected Areas in Cryptography*, pp. 320–337. Springer, 2011.

[4] Alex Biryukov, Arnab Roy, and Vesselin Velichkov. Differential Analysis of Block Ciphers SIMON and SPECK. FSE 2014, LNCS 8540, pp. 546–570, 2015.

[5] Blondeau C., Peyrin T., Wang L.: Known-key Distinguisher on Full PRESENT. CRYPTO 2015, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 455–474.

[6] Andrey Bogdanov, Florian Mendel, Francesco Regazzoni, Vincent Rijmen, and Elmar Tischhauser. Ale: Aes-based lightweight authenticated encryption. In *International Workshop on Fast Software Encryption*, pages 447–466. Springer, 2013.

[7] Bogdanov, A., Rijmen, V.: Linear hulls with correlation zero and linear cryptanalysis of block ciphers. Des. Codes Cryptogr. (2014) 70:369–383.

[8] Christina Boura, Maria Naya-Plasencia and Valentin Suder. Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and SIMON. ASIACRYPT 2014, PART I, LNCS 8873, pp. 179–199, 2014.

[9] Canteaut, A., Naya-Plasencia, M., Vayssiere, B.: Sieve-in-the-middle: improved MITM attacks. CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 222–240.

[10] Huaifeng Chen and Xiaoyun Wang. Improved Linear Hull Attack on Round-Reduced SIMON with Dynamic Key-Guessing Techniques. FSE 2016, LNCS 9783, pp. 428C449, 2016.

[11] Morris Dworkin. Special publication 800-38c: Recommendation for block cipher modes of operation: the ccm mode for authentication and confidentiality. *National Institute of Standards and Technology, US Department of Commerce*, 2005.

[12] Morris Dworkin. Special publication 800-38d: Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. *National Institute of Standards and Technology, US Department of Commerce*, 2007.

[13] Fuhr, T., Minaud, B.: Match box meet-in-the-middle attack against KATAN. FSE 2014, LNCS 8540, pp. 61–81, 2015.

[14] Yonglin Hao, Willi Meier. Truncated differential based known-key attacks on round-reduced Simon. Des. Codes Cryptogr. (2017) 83:467–492.

[15] Viet Tung Hoang, Reza Reyhanitabar, Phillip Rogaway, and Damian Vizár. Online authenticated-encryption and its nonce-reuse misuse-resistance. In *Annual Cryptology Conference*, pages 493–517. Springer, 2015.

[16] Knudsen L.R., Rijmen V.: Known-key distinguishers for some block ciphers. ASIACRYPT 2007. Lecture Notes in Computer Science, vol. 4833, pp. 315-324.

[17] Stefan Kolbl, Gregor Leander, and Tyge Tiessen. Observations on the SIMON Block Cipher Family. CRYPTO 2015, Part I, LNCS 9215, pp. 161–185, 2015.

[18] David McGrew and John Viega. The galois/counter mode of operation (gcm). *Submission to NIST. http://csrc. nist. gov/CryptoToolkit/modes/proposedmodes/gcm/gcm-spec. pdf*, 2004.

[19] Phillip Rogaway. Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 98–107. ACM, 2002.

[20] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes ocb and pmac. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 16–31. Springer, 2004.

[21] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. EUROCRYPT 2006, pp. 373–390, 2006.

[22] Ling Song, Lei Hu, Bingke Ma, and Danping Shi. Match Box Meet-in-the-Middle Attacks on the Simon Family of Block Ciphers. LightSec 2014, LNCS 8898, pp. 140–151, 2015.

[23] Ling Sun, Kai Fu, and Meiqin Wang. Improved Zero-Correlation Cryptanalysis on Simon. Inscrypt 2015, LNCS 9589, pp. 125–143, 2016.

[24] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers. ASIACRYPT 2014, PART I, LNCS 8873, pp. 158–178, 2014.

[25] Yosuke Todo. Structural Evaluation by Generalized Integral Property. EUROCRYPT 2015, Part I, LNCS 9056, pp. 287–314, 2015.

[26] Yosuke Todo and Masakatu Morii. Bit-Based Division Property and Application to SIMON Family. FSE 2016, LNCS 9783, pp. 357–377, 2016.

[27] Ning WANG, Xiaoyun WANG, Keting JIA, Jingyuan ZHAO. Differential attacks on reduced SIMON versions with dynamic key-guessing techniques. SCIENCE CHINA Information Sciences, 2018, Vol. 61 098103:1–098103:3.

[28] Doug Whiting, Russ Housley, and Neils Ferguson. Submission to nist: Counter with cbcmac (ccm)–aes mode of operation. *Computer Security Division, Computer Security Resource Center (NIST), available at: http://csrc. nist. gov/groups/ST/toolkit/BCM/documents/proposedmodes/ccm/ccm. pdf*, 2002.

Appendix: Figures

Game G_0

```
proc initialize
$K \twoheadleftarrow \mathcal{K}$

proc Enc.init($N^{(i)}$)
$S_0^{*(i)} \leftarrow E_K(N^{(i)})$

proc Enc.data($A_j^{(i)}$)
$Y_j^{(i)} \leftarrow E_K(A_j^{(i)})$
$S_j^{*(i)} \leftarrow 2 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$

proc Enc.data*($A_j^{(i)}$)
if $|A_j^{(i)}| = n$ then $Y_j^{(i)} \leftarrow E_K(A_j^{(i)})$
    $S_j^{*(i)} \leftarrow 2 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$
    else $Y_j^{(i)} \leftarrow E_K(A_j^{(i)} \| 10^{n-|A_j^{(i)}|-1})$
    $S_j^{*(i)} \leftarrow 3 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$
$S_0^{(i)} \leftarrow S_j^{*(i)}$

proc Enc.next($M_j^{(i)}$)
$C_j^{(i)} \leftarrow E_K(3 \cdot S_{j-1}^{(i)} \oplus (M_j^{(i)} \| [j]_{n-m}))$
$S_j^{(i)} \leftarrow 2 \cdot S_{j-1}^{(i)} \oplus (M_j^{(i)} \| [j]_{n-m})$
ret $C_j^{(i)}$

proc Enc.last($M_j^{(i)}, M_{j+1}^{(i)}$)
if $|M_{j+1}^{(i)}| > m$ then ret $\perp$
$C^{*(i)} \leftarrow E_K(3 \cdot S_{j-1}^{(i)} \oplus (M_j^{(i)} \| [j]_{n-m}))$
$C_j^{(i)} \leftarrow C^{*(i)}[1, n - m + |M_{j+1}^{(i)}|]$
$Z^{(i)} \leftarrow C^{*(i)}[n - m + |M_{j+1}^{(i)}|, n]$
$M^{*(i)} \leftarrow M_{j+1}^{(i)} \| Z^{(i)} \| [0]_{n-m}$
$C_{j+1}^{(i)} \leftarrow E_K(3 \cdot S_{j+1}^{(i)} \oplus M^{*(i)} \oplus E_K(|M|))$
ret $(C_j^{(i)}, C_{j+1}^{(i)})$
```

```
proc Dec.init($N^{(i)}$)
$S_0^{*(i)} \leftarrow E_K(N^{(i)})$

proc Dec.data($A_j^{(i)}$)
$Y_j^{(i)} \leftarrow E_K(A_j^{(i)})$
$S_j^{*(i)} \leftarrow 2 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$

proc Dec.data*($A_j^{(i)}$)
if $|A_j^{(i)}| = n$ then $Y_j^{(i)} \leftarrow E_K(A_j^{(i)})$
    $S_j^{*(i)} \leftarrow 2 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$
    else $Y_j^{(i)} \leftarrow E_K(A_j^{(i)} \| 10^{n-|A_j^{(i)}|-1})$
    $S_j^{*(i)} \leftarrow 3 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$
$S_0^{(i)} \leftarrow S_j^{*(i)}$

proc Dec.next($C_j^{(i)}$)
$M_j'^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus E_K^{-1}(C_j^{(i)})$
if $M_j'^{(i)}[m+1, n] \neq [i]_{n-m}$ then ret false
    else $M_j^{(i)} \leftarrow M_j'^{(i)}[1, m]$
ret $M_j^{(i)}$

proc Dec.last($C_j^{(i)}, C_{j+1}^{(i)}$)
if $|C_{j+1}^{(i)}| > m$ then ret $\perp$
$M'^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus E_K^{-1}(C_{j+1}^{(i)})$
        $\oplus E_K(|C| - (i+1) \cdot (n-m))$
if $M_{j+1}'^{(i)}[m+1, n] \neq [0]_{n-m}$ then ret false
    else $M_{j+1}^{(i)} \leftarrow M_{j+1}'^{(i)}[1, |C_j^{(i)}| - n + m]$
    $Z^{(i)} \leftarrow M_{j+1}'^{(i)}[|C_j^{(i)}| - n + m + 1, m]$
$X_j^{(i)} \leftarrow E_K^{-1}(C_j^{(i)} \| Z^{(i)})$
$M_j'^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus X_j^{(i)}$
if $M_j'^{(i)}[m+1, n] \neq [i]_{n-m}$ then ret false
    else $M_j^{(i)} \leftarrow M_j'^{(i)}[1, m]$
    ret $(M_j^{(i)}, M_{j+1}^{(i)})$
```

Figure 6.2: Game G_0 in the proof of Theorem 1.

```
                                                     Game $G_2$ $\boxed{\text{Game } G_3}$
proc initialize
$F(x) \leftarrow \mathsf{undef}$ for all $x$
$\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2 \leftarrow \emptyset$

proc Enc.init($N^{(i)}$)
if $F(N^{(i)}) = \mathsf{undef}$ then
    $F(N^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Dom}F(\cdot)$
$S_0^{*(i)} \leftarrow F(N^{(i)}); M^{(i)} \leftarrow \epsilon$

proc Enc.data($A_j^{(i)}$)
if $F(A_j^{(i)}) = \mathsf{undef}$ then
    $F(A_j^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Dom}F(\cdot)$
$Y_j^{(i)} \leftarrow F(A_j^{(i)})$
$S_j^{*(i)} \leftarrow 2 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$

proc Enc.data*($A_j^{(i)}$)
if $|A_j^{(i)}| = n$ then $A_j^{*(i)} \leftarrow A_j^{(i)}$
    else $A_j^{*(i)} \leftarrow A_j^{(i)} \| 10^{n-|A_j^{(i)}|-1}$
if $F(A_j^{*(i)}) = \mathsf{undef}$ then
    $F(A_j^{*(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Dom}F(\cdot)$
$Y_j^{(i)} \leftarrow F(A_j^{*(i)})$
if $|A_j^{(i)}| = n$ then $S_j^{*(i)} \leftarrow 2 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$
    $S_j^{*(i)} \leftarrow 3 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$
$S_0^{(i)} \leftarrow S_j^{*(i)}$
if $S_0^{(i)} \in \mathcal{Q}_0$ then
    $bad \leftarrow \mathsf{true}$ $\boxed{S_0^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_0}$
$\mathcal{Q}_0 \leftarrow \mathcal{Q}_0 \cup \{S_0^{(i)}\}$

proc Enc.next($M_j^{(i)}$)
$T_{j-1}^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus [j]_n$
if $T_{j-1}^{(i)} \in \mathcal{Q}_1|_T$ or $T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m}) \in \mathrm{Dom}F(\cdot)$ then
    $bad \leftarrow \mathsf{true}$ $\boxed{T_{j-1}^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T - ((M_j^{(i)} \| [0]_{n-m}) \oplus \mathrm{Dom}F(\cdot))}$
if $F(T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m})) = \mathsf{undef}$ then
    $F(T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m})) \twoheadleftarrow \{0,1\}^n - \mathrm{Ran}F(\cdot)$
$C_j^{(i)} \leftarrow F(T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m})); S_j^{(i)} \leftarrow 2 \cdot S_{j-1}^{(i)} \oplus (M_j^{(i)} \| [j]_{n-m})$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(T_{j-1}^{(i)}, M_j^{(i)}, C_j^{(i)})\}$
ret $C_j^{(i)}$

proc Enc.last($M_j^{(i)}, M_{j+1}^{(i)}$)
if $F(|M^{(i)}|) = \mathsf{undef}$ then
    $F(|M^{(i)}|) \twoheadleftarrow \{0,1\}^n - \mathrm{Dom}F(\cdot)$
$T_{j-1}^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus [j]_n; T_j^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus F(|M^{(i)}|)$
if $T_{j-1}^{(i)} \in \mathcal{Q}_1|_T$ or $T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m}) \in \mathrm{Dom}F(\cdot)$ then
    $bad \leftarrow \mathsf{true}$ $\boxed{T_{j-1}^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T - ((M_j^{(i)} \| [0]_{n-m}) \oplus \mathrm{Dom}F(\cdot))}$
if $F(T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m})) = \mathsf{undef}$ then
    $F(T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m})) \twoheadleftarrow \{0,1\}^n - \mathrm{Ran}F(\cdot)$
$C^{*(i)} \leftarrow F(T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m}))$
$C_j^{(i)} \leftarrow C^{*(i)}[1, n - m + |M_{j+1}^{(i)}|]; Z^{(i)} \leftarrow C^{*(i)}[n - m + |M_{j+1}^{(i)}| + 1, n]$
$M^{*(i)} \leftarrow M_{j+1}^{(i)} \| Z^{(i)} \| [0]_{n-m}$
if $T_j^{(i)} \oplus M^{*(i)} \in \mathrm{Dom}F(\cdot)$ then
    $bad \leftarrow \mathsf{true}$ $\boxed{T_j^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T - (M^{*(i)} \oplus \mathrm{Dom}F(\cdot))}$
if $F(T_j^{(i)} \oplus M^{*(i)}) = \mathsf{undef}$ then
    $F(T_j^{(i)} \oplus M^{*(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Ran}F(\cdot)$
$C_{j+1}^{(i)} \leftarrow F(T_j^{(i)} \oplus M^{*(i)})$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(T_{j-1}^{(i)}, M_j^{(i)}, C^{*(i)})\}$
$\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{(T_{j-1}^{(i)}, T_j^{(i)}, M_j^{(i)}, M_{j+1}^{(i)}, C_j^{(i)})\}$
ret $(C_j^{(i)}, C_{j+1}^{(i)})$
```

Figure 6.3: Encryption of Games G_2 and G_3 in the proof of Theorem 1. Game G_3 contains the corresponding boxed statements, but game G_2 doesn't.

```
                                                    Game $G_2$ $\boxed{\text{Game } G_3}$
proc Dec.init($N^{(i)}$)
if $F(N^{(i)}) = \mathsf{undef}$ then
    $F(N^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Ran}F(\cdot)$
$S_0^{*(i)} \leftarrow F(N^{(i)}); C^{(i)} \leftarrow \epsilon$

proc Dec.data($A_j^{(i)}$)
if $F(A_j^{(i)}) = \mathsf{undef}$ then
    $F(A_j^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Dom}F(\cdot)$
$Y_j^{(i)} \leftarrow F(A_j^{(i)})$
$S_j^{*(i)} \leftarrow 2 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$

proc Dec.data*($A_j^{(i)}$)
if $|A_j^{(i)}| = n$ then $A_j^{*(i)} \leftarrow A_j^{(i)}$
    else $A_j^{*(i)} \leftarrow A_j^{(i)} \| 10^{n-|A_j^{(i)}|-1}$
if $F(A_j^{*(i)}) = \mathsf{undef}$ then
    $F(A_j^{*(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Dom}F(\cdot)$
$Y_j^{(i)} \leftarrow F(A_j^{*(i)})$
if $|A_j^{(i)}| = n$ then $S_j^{*(i)} \leftarrow 2 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$
    $S_j^{*(i)} \leftarrow 3 \cdot S_{j-1}^{*(i)} \oplus Y_j^{(i)}$
$S_0^{(i)} \leftarrow S_j^{*(i)}$
if $S_0^{(i)} \in \mathcal{Q}_0$ then
    $bad \leftarrow \mathsf{true}$ $\boxed{S_0^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_0}$
$\mathcal{Q}_0 \leftarrow \mathcal{Q}_0 \cup \{S_0^{(i)}\}$

proc Dec.next($C_j^{(i)}$)
$T_{j-1}^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus [j]_n$
if $T_{j-1}^{(i)} \in \mathcal{Q}_1|_T$ then
    $bad \leftarrow \mathsf{true}$ $\boxed{T_{j-1}^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T}$
if $F^{-1}(C_j^{(i)}) = \mathsf{undef}$ then
    $F^{-1}(C_j^{(i)}) \twoheadleftarrow \{0,1\}^m - \mathrm{Dom}F(\cdot)$
if $(F^{-1}(C_j^{(i)}) \oplus T_{j-1}^{(i)})[m+1,n] \neq [0]_{n-m}$ then ret false
    else $M_j^{(i)} \leftarrow (F^{-1}(C_j^{(i)}) \oplus T_{j-1}^{(i)})[1,m]$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(T_{j-1}^{(i)}, M_j^{(i)}, C_j^{(i)})\}$
ret $M_j^{(i)}$

proc Dec.last($C_j^{(i)}, C_{j+1}^{(i)}$)
if $F(|C^{(i)}| - (i+1) \cdot (n-m)) = \mathsf{undef}$ then
    $F(|C^{(i)}| - (i+1) \cdot (n-m)) \twoheadleftarrow \{0,1\}^n - \mathrm{Ran}F(\cdot)$
$T_{j-1}^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus [j]_n; T_j^{(i)} \leftarrow 3 \cdot S_{j-1}^{(i)} \oplus F(|C^{(i)}| - (i+1) \cdot (n-m))$
if $T_j^{(i)} \in \mathcal{Q}_2|_T$ then
    $bad \leftarrow \mathsf{true}$ $\boxed{T_j^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_2|_T}$
if $F^{-1}(C_{j+1}^{(i)}) = \mathsf{undef}$ then
    $F^{-1}(C_{j+1}^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Dom}F(\cdot)$
if $(F^{-1}(C_{j+1}^{(i)}) \oplus T_j^{(i)})[m+1,n] \neq [0]_{n-m}$ then ret false
    else $M_{j+1}^{\prime(i)} \leftarrow (F^{-1}(C_{j+1}^{(i)}) \oplus T_j^{(i)})$
$M^{(i)} \leftarrow M_{j+1}^{\prime(i)}[1, |C_j^{(i)}| - n + m]; Z^{(i)} \leftarrow M_{j+1}^{\prime(i)}[|C_j^{(i)}| - n + m + 1, n]$
if $T_j^{(i)} \in \mathcal{Q}_1|_T$ then
    $bad \leftarrow \mathsf{true}$ $\boxed{T_{j-1}^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T}$
if $F^{-1}(C_j^{(i)} \| Z^{(i)}) = \mathsf{undef}$ then
    $F^{-1}(C_j^{(i)} \| Z^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Dom}F(\cdot)$
if $(F^{-1}(C_j^{(i)} \| Z^{(i)}) \oplus T_{j-1}^{(i)})[m+1,n] \neq [0]_{n-m}$ then ret false
    else $M_j^{\prime(i)} \leftarrow (F^{-1}(C_j^{(i)} \| Z^{(i)}) \oplus T_{j-1}^{(i)})[1,m]$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(T_{j-1}^{(i)}, M_j^{(i)}, C_j^{(i)} \| Z^{(i)})\}$
$\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{(T_{j-1}^{(i)}, T_j^{(i)}, M_j^{(i)}, M_{j+1}^{(i)}, C_{j+1}^{(i)})\}$
ret $(M_j^{(i)}, M_{j+1}^{(i)})$
```

Figure 6.4: Decryption of Games G_2 and G_3 in the proof of Theorem 1. Game G_3 contains the corresponding boxed statements, but game G_2 doesn't.

```
                                        Game $G_4$
proc initialize
$G_x(y) \leftarrow$ undef for all $x, y$
$\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{P} \leftarrow \emptyset$

proc Enc.init($N^{(i)}$)
proc Enc.data($A_j^{(i)}$)
proc Enc.data*($A_j^{(i)}$)

proc Enc.next($M_j^{(i)}$)
$T_{j-1}^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T - ((M_j^{(i)}\|[0]_{n-m}) \oplus \mathcal{P})$
$G_{T_{j-1}^{(i)}}(M_{j-1}^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Ran}G(\cdot)$
$C_j^{(i)} \leftarrow G_{T_{j-1}^{(i)}}(M_{j-1}^{(i)})$
$\mathcal{P} \leftarrow \mathcal{P} \cup \{T_{j-1}^{(i)} \oplus (M_j^{(i)}\|[0]_{n-m})\}$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(T_{j-1}^{(i)}, M_j^{(i)}, C_j^{(i)})\}$
ret $C_j^{(i)}$

proc Enc.last($M_j^{(i)}, M_{j+1}^{(i)}$)
$T_{j-1}^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T - ((M_j^{(i)}\|[0]_{n-m}) \oplus \mathcal{P})$
$G_{T_{j-1}^{(i)}}(M_j^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Ran}G(\cdot)$
$C^{*(i)} \leftarrow G_{T_{j-1}^{(i)}}(M_j^{(i)})$
$C_j^{(i)} \leftarrow C^{*(i)}[1, n-m+|M_{j+1}^{(i)}|]$
$Z^{(i)} \leftarrow C^{*(i)}[n-m+|M_{j+1}^{(i)}|+1, n]$
$\mathcal{P} \leftarrow \mathcal{P} \cup \{T_{j-1}^{(i)} \oplus (M_j^{(i)}\|[0]_{n-m})\}$
$T_j^{(i)} \twoheadleftarrow \{0,1\}^n - ((M_{j+1}^{(i)}\|Z^{(i)}\|[0]_{n-m}) \oplus \mathcal{P})$
$G_{T_j^{(i)}}(M_{j+1}^{(i)}\|Z^{(i)}) \twoheadleftarrow \{0,1\}^n - \mathrm{Ran}G(\cdot)$
$C_{j+1}^{(i)} \leftarrow G_{T_j^{(i)}}(M_{j+1}^{(i)}\|Z^{(i)})$
$\mathcal{P} \leftarrow \mathcal{P} \cup \{T_j^{(i)} \oplus (M_{j+1}^{(i)}\|Z^{(i)}\|[0]_{n-m})\}$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(T_{j-1}^{(i)}, M_j^{(i)}, C^{*(i)})\}$
$\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{(T_{j-1}^{(i)}, T_j^{(i)}, M_j^{(i)}, M_{j+1}^{(i)}, C_j^{(i)})\}$
ret $(C_j^{(i)}, C_{j+1}^{(i)})$
```

Figure 6.5: Encryption of Games G_4 in the proof of Theorem 1

```
Game $G_4$
proc Dec.init($N^{(i)}$)
proc Dec.data($A_j^{(i)}$)
proc Dec.data*($A_j^{(i)}$)

proc Dec.next($C_j^{(i)}$)
$T_{j-1}^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T$
$\widetilde{\mathcal{M}} \leftarrow \{0,1\}^m - \{M : (T_{j-1}^{(i)}, M, \cdot) \in \mathcal{Q}_1\}$
$\widetilde{\mathcal{C}} \leftarrow \{0,1\}^n - \{C : (T_{j-1}^{(i)}, \cdot, C) \in \mathcal{Q}_1\}$
$\theta \twoheadleftarrow [0..|\widetilde{\mathcal{C}}|]$
if $\theta \leq |\widetilde{\mathcal{M}}|$ then
    $M_j^{(i)} \leftarrow$ the $\theta^{th}$ string of $\widetilde{\mathcal{M}}$
    $G_{T_{j-1}^{(i)}}(M_j^{(i)}) \leftarrow C_j^{(i)}$; ret $M_j^{(i)}$
    $\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(T_{j-1}^{(i)}, M_j^{(i)}, C_j^{(i)})\}$
    else ret false

proc Dec.last($C_j^{(i)}, C_{j+1}^{(i)}$)
$T_{j-1}^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_1|_T; T_j^{(i)} \twoheadleftarrow \{0,1\}^n - \mathcal{Q}_2|_T$
$\widetilde{\mathcal{M}} \leftarrow \{0,1\}^m - \{M : (T_{j-1}^{(i)}, T_j^{(i)}, \cdot, M, \cdot) \in \mathcal{Q}_2\}$
$\widetilde{\mathcal{C}} \leftarrow \{0,1\}^n - \{C : (T_{j-1}^{(i)}, T_j^{(i)}, \cdot, \cdot, C) \in \mathcal{Q}_2\}$
$\theta \twoheadleftarrow [0..|\widetilde{\mathcal{C}}|]$
if $\theta \leq |\widetilde{\mathcal{M}}|$ then
    $M_{j+1}'^{(i)} \leftarrow$ the $\theta^{th}$ string of $\widetilde{\mathcal{M}}$;
    $G_{T_j^{(i)}}(M_{j+1}'^{(i)}) \leftarrow C_{j+1}^{(i)}$; ret $M_j^{(i)}$
$M_{j+1}^{(i)} \leftarrow M_{j+1}'^{(i)}[1, |C_j^{(i)}| - n + m]$
$Z^{(i)} \leftarrow M_{j+1}'^{(i)}[n - m + |C_j^{(i)}| + 1, m]$
    else ret false
$\widetilde{\mathcal{M}} \leftarrow \{0,1\}^m - \{M : (T_{j-1}^{(i)}, M, \cdot) \in \mathcal{Q}_1\}$
$\widetilde{\mathcal{C}} \leftarrow \{0,1\}^n - \{C : (T_{j-1}^{(i)}, \cdot, C) \in \mathcal{Q}_1\}$
$\theta \twoheadleftarrow [0..|\widetilde{\mathcal{C}}|]$
if $\theta \leq |\widetilde{\mathcal{M}}|$ then
    $M_j^{(i)} \leftarrow$ the $\theta^{th}$ string of $\widetilde{\mathcal{M}}$
    $G_{T_{j-1}^{(i)}}(M_j^{(i)}) \leftarrow C_j^{(i)} \| Z^{(i)}$; ret $(M_j^{(i)}, M_{j+1}^{(i)})$
    $\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(T_{j-1}^{(i)}, M_j^{(i)}, C_j^{(i)} \| Z^{(i)})\}$
    $\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{(T_{j-1}^{(i)}, T_j^{(i)}, M_j^{(i)}, M_{j+1}^{(i)}, C_j^{(i)})\}$
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{T_{j-1}^{(i)} \oplus (M_j^{(i)} \| [0]_{n-m}), T_j^{(i)} \oplus (M_{j+1}^{(i)} \| Z^{(i)} \| [0]_{n-m})\}$
    else ret false
```

Figure 6.6: Decryption of Game G_4 in the proof of Theorem 1

Game G_5

```
proc initialize
$H \leftarrow \mathrm{IdealOAE}(\tau, 2\tau)$

proc Enc.init($N^{(i)}$)
$M^{(i)} \leftarrow \epsilon$

proc Enc.data/data*($A_j^{(i)}$)
$A^{(i)} \leftarrow A^{(i)} \| A_j^{(i)}$

proc Enc.next($M_j^{(i)}$)
$C^{(i)} \leftarrow H_{N^{(i)},A^{(i)},M^{(i)},0}(M_j^{(i)})$
$M^{(i)} \leftarrow M^{(i)} \| M_j^{(i)}$
ret $C_j^{(i)}$

proc Enc.last($M_j^{(i)}, M_{j+1}^{(i)}$)
if $|M_{j+1}^{(i)}| > m$ then ret $\perp$
$(C_j^{(i)}, C_{j+1}^{(i)}) \leftarrow H_{N^{(i)},A^{(i)},M^{(i)},1}(M_j^{(i)}, M_{j+1}^{(i)})$
ret $(C_j^{(i)}, C_{j+1}^{(i)})$
```

```
proc Dec.init($N^{(i)}$)
$C^{(i)} \leftarrow \epsilon$

proc Dec.data/data*($A_j^{(i)}$)
$A^{(i)} \leftarrow A^{(i)} \| A_j^{(i)}$

proc Dec.next($C_j^{(i)}$)
if $\exists M_j^{(i)}$ s.t. $H_{N^{(i)},A^{(i)},M^{(i)},0}(M_j^{(i)}) = C^{(i)}$
    then ret $M_j^{(i)}$; $M^{(i)} \leftarrow M^{(i)} \| M_j^{(i)}$
    else ret flase

proc Dec.last($C_j^{(i)}, C_{j+1}^{(i)}$)
if $|C_{j+1}^{(i)}| > m$ then ret $\perp$
if $\exists(M^{(i)}, M_{j+1}^{(i)})$ s.t. $(C_j^{(i)}, C_{j+1}^{(i)})$
        $= H_{N^{(i)},A^{(i)},M^{(i)},1}(M^{(i)}, M_{j+1}^{(i)})$
    then ret $(M_j^{(i)}, M_{j+1}^{(i)})$
    else ret false
```

Figure 6.7: Game G_5 in the proof of Theorem 1.