SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Efficient beyond-birthday-bound secure authenticated encryption modes

Ping ZHANG[1,2], Honggang HU[1,2*] & Peng WANG[3]

[1]*University of Science and Technology of China, Hefei* 230027*, China;*
[2]*Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, Hefei* 230027*, China;*
[3]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100049*, China*

Appendix A Preliminaries

Notations. Let $\{0,1\}^*$ denote the set containing all finite bit strings (including the empty string). Let n be an integer, and $(\{0,1\}^n)^+$ be the set of all strings whose lengths are positive multiples of n bits. If X is a set, then $x \xleftarrow{\$} X$ is a value randomly chosen from X, and $|X|$ stands for the number of elements in X. For a finite string x, $|x|$ stands for its length. For two finite strings x and y, let $x\|y$ or xy denote the concatenation of them. Given a finite string $x \in \{0,1\}^*$ with $|x| \geqslant n$, let $(x)_n$ be the most significant (leftmost) n bits of x. For positive integers n and m such that $m \leqslant 2^n - 1$, let $[m]_n$ be the n-bit binary representation of m.

Finite field. Given a basis, the finite field $GF(2^n)$ can be viewed as the set $\{0,1\}^n$. For an n-bit string $a = a_{n-1} \cdots a_1 a_0 \in \{0,1\}^n$, we can define a polynomial $a(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ with binary coefficients. Hence, any integer between 0 and $2^n - 1$ can also be viewed as a polynomial with binary coefficients of degree at most $n-1$. For example, 2 corresponds to x, 3 corresponds to $x+1$, and 7 corresponds to $x^2 + x + 1$. The addition in $GF(2^n)$ is the addition of polynomials over $GF(2)$. We denote this operation by bitwise XOR, such as $a \oplus b$, where $a, b \in GF(2^n)$. To define multiplication in $GF(2^n)$, we need an irreducible polynomial $f(x)$ of degree n over $GF(2)$. For $n = 256$, $f(x) = x^{256} + x^{10} + x^5 + x^2 + 1$. The multiplication of two elements $A, B \in GF(2^n)$ is defined as the corresponding polynomial multiplication over $GF(2)$ reduced modulo $f(x)$, that is $A(x)B(x)\ mod\ f(x)$.

Random function and pseudorandom function (PRF). Let $Func(m,n)$ be the set of all functions from $\{0,1\}^m$ to $\{0,1\}^n$. If $m = n$, we write $Func(n)$ for simplicity. Let $R \xleftarrow{\$} Func(m,n)$ be a function randomly chosen from $Func(m,n)$.

An adversary is a probabilistic algorithm with access to certain oracles for the cryptographic scheme. Let $\mathcal{A}^O \Rightarrow 1$ be the event that an adversary $\mathcal{A}$ outputs 1 after interacting with the oracle O. Suppose that $F : \mathcal{K}_f \times \{0,1\}^m \to \{0,1\}^n$ is a keyed-function. Let $\mathcal{A}$ be a PRF-adversary with access to an encryption oracle. Then the PRF-advantage of $\mathcal{A}$ attacking F is defined as

$$Adv_F^{prf}(\mathcal{A}) = |Pr[K \xleftarrow{\$} \mathcal{K}_f : \mathcal{A}^{F_K(\cdot)} \Rightarrow 1] - Pr[R \xleftarrow{\$} Func(m,n) : \mathcal{A}^{R(\cdot)} \Rightarrow 1]|.$$

The probabilities are taken over the random coins used by the oracles and also over internal coins of $\mathcal{A}$, if any. If the PRF-advantage $Adv_F^{prf}(\mathcal{A})$ is negligible, F_K is a secure pseudorandom function (PRF).

For $t, q, l, \sigma > 0$, let $Adv(t,q,l,\sigma) = max_{\mathcal{A}} Adv(\mathcal{A})$ be the maximum advantage of all adversaries, where t is the running time, q is the number of oracle queries, l is the maximum block-length, and σ is the totally number of blocks in all queries (query complexity).

AE syntax. An IV-based AEAD scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of an encryption algorithm $\mathcal{E} : \mathcal{K} \times \mathcal{IV} \times \mathcal{AD} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$ and a decryption algorithm $\mathcal{D} : \mathcal{K} \times \mathcal{IV} \times \mathcal{AD} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}$, i.e.,

$$C||T \leftarrow \mathcal{E}_K(IV, A, M),$$
$$M/\perp \leftarrow \mathcal{D}_K(IV, A, C, T),$$

where $K \in \mathcal{K}$ is a key, $IV \in \mathcal{IV}$ is an initial value, $\mathcal{IV} = \{0,1\}^n$, $A \in \mathcal{AD}$ is an associated data, $\mathcal{AD} \subseteq \{0,1\}^*$, $M \in \mathcal{M}$ is a plaintext, $\mathcal{M} \subseteq \{0,1\}^*$, $C \in \mathcal{C}$ is a ciphertext, $\mathcal{C} \subseteq \{0,1\}^*$, $|C| = |M|$, and $T \in \mathcal{T}$ is a tag, $\mathcal{T} \subseteq \{0,1\}^*$. $\mathcal{D}_K(IV, A, C, T) = M$ if and only if $\mathcal{E}_K(IV, A, M) = (C, T)$. The symbol $\perp$ indicates the failure of the decryption oracle. A secure AE scheme returns $\perp$ if it receives an error (C, T) pair. If there is no associated data, A can be omitted.

* Corresponding author (email: hghu2005@ustc.edu.cn)

Privacy of AE. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IV-based AEAD scheme. Let $\mathcal{A}$ be an adversary, which has access to the encryption oracle. The adversary $\mathcal{A}$ queries (IV^i, A^i, M^i) to the encryption oracle and receives (C^i, T^i), where $1 \leqslant i \leqslant q$. Then the PRIV-advantage of $\mathcal{A}$ against $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined as

$$Adv_{\Pi}^{priv}(\mathcal{A}) = |Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot,\cdot)} \Rightarrow 1] - Pr[\mathcal{A}^{\$(\cdot,\cdot,\cdot)} \Rightarrow 1]|,$$

where $\$(\cdot,\cdot,\cdot)$ denotes the oracle that takes (IV, A, M) as inputs and returns a random string of length $|C| + |T|$. The goal of $\mathcal{A}$ is to distinguish the encryption oracle $\mathcal{E}_K(\cdot,\cdot,\cdot)$ from the random oracle $\$(\cdot,\cdot,\cdot)$. This privacy notion is also called confidentiality, whose goal is to ensure that any 1 bit of the plaintext is not obtained from the ciphertext for the adversary.

Authenticity of AE. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IV-based AEAD scheme. Let $K \in \mathcal{K}$. Let $\mathcal{A}$ be an adversary, which has access to the encryption oracle $\mathcal{E}_K(\cdot,\cdot,\cdot)$ and the decryption oracle $D_K(\cdot,\cdot,\cdot,\cdot)$. Firstly, the adversary $\mathcal{A}$ has access to the encryption oracle $\mathcal{E}_K(\cdot,\cdot,\cdot)$ and receives $(C^i, T^i) = \mathcal{E}_K(IV^i, A^i, M^i)$, where $1 \leqslant i \leqslant q$. Then $\mathcal{A}$ makes a forgery attempt $(IV', A', C', T') \notin \{(IV^i, A^i, C^i, T^i)\}_{i=1}^{q}$. The forgery attempt succeeds if $\mathcal{D}_K(IV', A', C', T') \neq \perp$. Then the AUTH-advantage of $\mathcal{A}$ against $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined as

$$Adv_{\Pi}^{auth}(\mathcal{A}) = Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot,\cdot),\mathcal{D}_K(\cdot,\cdot,\cdot,\cdot)} \ forges].$$

This authenticity notion is also called integrity of ciphertext (INT-CTXT), whose goal is to prevent the adversary from tampering or forging an attempt that can be correctly decrypted.

Appendix B Universal hash function family

This section reviews the definition of universal hash function family [1, 2].

Definition 1 ((ϵ, δ)-A(X)U hash function family)**.** Let $n \geqslant 1$ be an integer. Let $\mathcal{H} = \{H : \mathcal{K}_h \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n\}$ be a set of keyed hash functions. $\mathcal{H}$ is called a family of (ϵ, δ)-almost-(XOR)-universal ((ϵ, δ)-A(X)U) hash functions if the following two conditions hold:

1. For any two distinct pairs (x, y), $(x', y') \in \{0,1\}^* \times \{0,1\}^*$ and any element $z \in \{0,1\}^n$,

$$Pr[L \xleftarrow{\$} \mathcal{K}_h : H_L(x, y) \oplus H_L(x', y') = z] \leqslant \epsilon.$$

2. For any element $(x, y) \in \{0,1\}^* \times \{0,1\}^*$ and any element $z \in \{0,1\}^n$,

$$Pr[L \xleftarrow{\$} \mathcal{K}_h : H_L(x, y) = z] \leqslant \delta.$$

If $\delta = 2^{-n}$, $(\epsilon, 2^{-n})$-A(X)U is also called as ϵ uniform A(X)U. An ϵ uniform A(X)U-hash function is instantiated into GHASH [5, 8, 9], which is described in Algorithm B1. In the following, all multiplications (denoted by "$\cdot$") are finite-field multiplications over $GF(2^n)$.

Algorithm B1 The hash function $H_L(A, M)$

Input: a key L, an associated data A, and a plaintext M
Output: a hash value H
1: $A^* = A||0^{n-|A| \ mod \ n}$, $M^* = M||0^{n-|M| \ mod \ n}$
2: $X \leftarrow A^*||M^*||[|A|]_{n/2}||[|M|]_{n/2}$
3: $X_1 \| \cdots \| X_x \leftarrow X$, $|X_i| = n, 1 \leqslant i \leqslant x$
4: $H = 0$
5: **for** $i = 1$ to x **do**
6: $\quad H = (H \oplus X_i) \cdot L$
7: **end for**
8: return H

Appendix C RWCTR: the first keyed-function-based BBB-secure AE mode

Appendix C.1 RWCTR

RWCTR consists of an encryption algorithm $\mathcal{E}$ and a decryption algorithm $\mathcal{D}$, which are respectively presented as follows.

1. Encryption algorithm $\mathcal{E}$. The encryption algorithm $\mathcal{E}$ takes two keys $(L, K) \in \mathcal{K}_h \times \mathcal{K}_f$, an n-bit random IV $U \in \{0,1\}^n$, an associated data $A \in \{0,1\}^*$, and a plaintext $M \in \{0,1\}^*$ as input, and returns a ciphertext $C \in \{0,1\}^*$ and an authentication tag $T \in \{0,1\}^n$, where $\mathcal{K}_h$ and $\mathcal{K}_f$ are two non-empty sets of keys. It includes two steps. Firstly, we modify the RWMAC [3] construction and utilize it to generate the authentication tag T. Given a keyed compression function $F : \mathcal{K}_f \times \{0,1\}^{2n} \to \{0,1\}^n$ and an ϵ uniform A(X)U-hash function $H : \mathcal{K}_h \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$, one has $S = H_L(A, M) \oplus U$ and $T = F_K(U, S)$. Then we utilize a randomized CTR-like construction to generate the ciphertext C. Let $M_1 \| \cdots \| M_l \leftarrow M, |M_i| = n, 1 \leqslant i \leqslant l-1$, and $0 < |M_l| \leqslant n$. If the plaintext length in bits is not a multiple of the block length, i.e., $0 < |M_l| < n$, the last ciphertext block is truncated to $|M_l|$-bit.

2. Decryption algorithm $\mathcal{D}$. The decryption algorithm $\mathcal{D}$ is the inverse of the encryption algorithm $\mathcal{E}$. It takes two keys $(L, K) \in \mathcal{K}_h \times \mathcal{K}_f$, an n-bit random IV $U \in \{0,1\}^n$, an associated data $A \in \{0,1\}^*$, a ciphertext $C \in \{0,1\}^*$, and an

Algorithm C1 The encryption algorithm of RWCTR

Input: two keys (L, K), an n-bit random IV U, an associated data A, and a plaintext M
Output: a ciphertext C and an authentication tag T

Partition M into $M_1\|\cdots\|M_l$, $|M_i| = n, 1 \leqslant i \leqslant l-1, 0 < |M_l| \leqslant n$
$S = H_L(A, M) \oplus U$
$T = F_K(U, S)$
for $i = 1$ to $l-1$ **do**
  $S_i = F_K(U, T+i)$
  $C_i = M_i \oplus S_i$
end for
$S_l = F_K(U, T+l)$
$C_l = M_l \oplus (S_l)_{|M_l|}$
$C = C_1\|C_2\|\cdots\|C_l$
return $(C\|T)$

Algorithm C2 The decryption algorithm of RWCTR

Input: two keys (L, K), an n-bit random IV U, an associated data A, a ciphertext C, and an authentication tag T
Output: a plaintext M or $\perp$

Partition C into $C_1\|C_2\|\cdots\|C_l$, $|C_i| = n, 1 \leqslant i \leqslant l-1, 0 < |C_l| \leqslant n$
for $i = 1$ to $l-1$ **do**
  $S_i = F_K(U, T+i)$
  $M_i = C_i \oplus S_i$
end for
$S_l = F_K(U, T+l)$
$M_l = C_l \oplus (S_l)_{|C_l|}$
$M = M_1\|\cdots\|M_l$
$S = H_L(A, M) \oplus U$
$T' = F_K(U, S)$
if $T' = T$ **then**
  return M
else
  return $\perp$ (INVALID)
end if

Figure C1 The encryption and decryption algorithms of RWCTR

authentication tag $T \in \{0,1\}^n$ as input, and returns a plaintext $M \in \{0,1\}^*$ or an invalid symbol $\perp$. Considering that the encryption algorithm $\mathcal{E}$ is actually a stream cipher, the decryption algorithm $\mathcal{D}$ is the same as the encryption algorithm $\mathcal{E}$ except the verification of the tag at the end of the decryption process. It recomputes a new tag T' using the decrypted plaintext. If $T' = T$, then the decryption algorithm $\mathcal{D}$ outputs the decrypted plaintext M. Otherwise it returns $\perp$.

The encryption and decryption algorithms of RWCTR are presented in Figure C1.

Appendix C.2 The security proof of Theorem 1

Before presenting the security proof, we first introduce some notes and lemmas.

Notes. In this paper, we assume that all adversaries are deterministic and can't make trivial queries, i.e., 1). all adversaries never repeat a query; 2). all adversaries never make a decryption query for an output pair obtained by an encryption query; 3). all adversaries never make an encryption query for an output pair obtained by a decryption query.

Lemma 1. If $0 \leqslant x \leqslant 1$, then $1 - x \leqslant e^{-x} \leqslant 1 - (1-1/e)x$. Generally, if $0 \leqslant x_i \leqslant 1$ and $0 \leqslant \sum_{i=1}^{k} x_i \leqslant 1$ for any $1 \leqslant i \leqslant k$, then

$$1 - \sum_{i=1}^{k} x_i \leqslant \prod_{i=1}^{k}(1-x_i) \leqslant e^{-\sum_{i=1}^{k} x_i} \leqslant 1 - (1-\frac{1}{e})\sum_{i=1}^{k} x_i.$$

Lemma 2 (AM-GM-QM inequality)**.** For a set of non-negative real numbers $x_1, x_2, \cdots, x_n$, the following always holds:

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} \leqslant \frac{\sum_{i=1}^{n} x_i}{n} \leqslant \left(\frac{\sum_{i=1}^{n} x_i^2}{n}\right)^{\frac{1}{2}}.$$

Next, we provide the security proof of Theorem 1 by combining the following Lemmas 3 and 4.

Lemma 3. Let $F : \mathcal{K}_f \times \{0,1\}^{2n} \to \{0,1\}^n$ be a keyed compression function. Let $K \xleftarrow{\$} \mathcal{K}_f$ and $R \xleftarrow{\$} Func(2n, n)$. For RWCTR, we replace the underlying keyed compression function F_K with a function R. Then we have

$$Adv^{priv}_{RWCTR[F]}(t, q, \sigma) \leqslant Adv^{prf}_{F}(t', \sigma) + Adv^{priv}_{RWCTR[R]}(t, q, \sigma),$$
$$Adv^{auth}_{RWCTR[F]}(t, q, q_v, \sigma) \leqslant Adv^{prf}_{F}(t', \sigma) + Adv^{auth}_{RWCTR[R]}(t, q, q_v, \sigma),$$

where t or t' is the time complexity, $t' = t + O(n\sigma)$, q is the number of queries, q_v is the number of forgery attempts, and σ is the query complexity.

Proof. Let $\mathcal{A}$ be an adversary, which makes at most q encryption queries to RWCTR. Let $(L, K) \xleftarrow{\$} \mathcal{K}_h \times \mathcal{K}_f$. Then the PRIV-advantage of the adversary $\mathcal{A}$ can be expressed as

$$\begin{aligned}
Adv^{priv}_{RWCTR[F]}(\mathcal{A}) =& |Pr[\mathcal{A}^{\mathcal{E}_{L,K}} \Rightarrow 1] - Pr[\mathcal{A}^{\$} \Rightarrow 1]| \\
=& |Pr[\mathcal{A}^{\mathcal{E}_{L,K}} \Rightarrow 1] - Pr[\mathcal{A}^{\mathcal{E}_L[R]} \Rightarrow 1] + Pr[\mathcal{A}^{\mathcal{E}_L[R]} \Rightarrow 1] - Pr[\mathcal{A}^{\$} \Rightarrow 1]| \\
\leqslant& |Pr[\mathcal{B}^{F_K} \Rightarrow 1] - Pr[\mathcal{B}^{R} \Rightarrow 1]| + |Pr[\mathcal{C}^{\mathcal{E}_L[R]} \Rightarrow 1] - Pr[\mathcal{C}^{\$} \Rightarrow 1]| \\
=& Adv^{prf}_{F}(\mathcal{B}) + Adv^{priv}_{RWCTR[R]}(\mathcal{C}),
\end{aligned}$$

where a new adversary $\mathcal{B}$ (against the pseudorandomness of F_K, making at most σ oracle queries) and a new adversary $\mathcal{C}$ (against the privacy of RWCTR[R], making at most q oracle queries) use $\mathcal{A}$ as a subroutine and simulate oracles for $\mathcal{A}$.

Similarity, we assume that an adversary $\mathcal{A}$ makes at most q encryption queries and q_v forgery attempts to RWCTR. Let $(L, K) \xleftarrow{\$} \mathcal{K}_h \times \mathcal{K}_f$, then the AUTH-advantage of the adversary $\mathcal{A}$ is described as follows:

$$\begin{aligned}
Adv^{auth}_{RWCTR[F]}(\mathcal{A}) =& Pr[\mathcal{A}^{\mathcal{E}_{L,K},\mathcal{D}_{L,K}} \ forges] \\
=& Pr[\mathcal{A}^{\mathcal{E}_{L,K},\mathcal{D}_{L,K}} \ forges] - Pr[\mathcal{A}^{\mathcal{E}_L[R],\mathcal{D}_L[R]} \ forges] + Pr[\mathcal{A}^{\mathcal{E}_L[R],\mathcal{D}_L[R]} \ forges] \\
\leqslant& |Pr[\mathcal{B}^{F_K} \ forges] - Pr[\mathcal{B}^{R} \ forges]| + Pr[\mathcal{C}^{\mathcal{E}_L[R],\mathcal{D}_L[R]} \ forges] \\
=& Adv^{prf}_{F}(\mathcal{B}) + Adv^{auth}_{RWCTR[R]}(\mathcal{C}),
\end{aligned}$$

where a new adversary $\mathcal{B}$ (against the pseudorandomness of F_K, making at most σ oracle queries) and a new adversary $\mathcal{C}$ (against the authenticity of RWCTR[R], making at most q encryption queries and q_v forgery attempts) use $\mathcal{A}$ as a subroutine and simulate oracles for $\mathcal{A}$.

The standard primitive F_K is replaced with an ideal primitive R, which is a classical reduction in the provable security. This lemma states the security loss induced by this reduction. Next, we introduce Lemma 4 to upper bound the security of RWCTR[R].

Lemma 4. Let $H : \mathcal{K}_h \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be an ϵ uniform AXU-hash function. Let $L \in \mathcal{K}_h$ and $R \xleftarrow{\$} Func(2n, n)$. Then, for RWCTR[R], we have

$$\begin{aligned}
Adv^{priv}_{RWCTR[R]}(t, q, \sigma) \leqslant& \frac{q^2\epsilon}{2^{n+1}} + \frac{3\sigma}{2^{n+1}}, \\
Adv^{auth}_{RWCTR[R]}(t, q, q_v, \sigma) \leqslant& \frac{q^2\epsilon}{2^{n+1}} + \frac{3\sigma}{2^{n+1}} + q_v(n-1)\epsilon + \frac{2q_v}{2^n},
\end{aligned}$$

if $q \leqslant min\{2^{n-2}, \sqrt{2^n\epsilon^{-1}}\}$, where t is the time complexity, q is the number of queries, q_v is the number of forgery attempts, and σ is the query complexity.

Proof. Privacy. The idea of our proof comes from [3]. Let $\mathcal{A}$ be an adversary, which makes q queries $(U^1, A^1, M^1), \cdots, (U^q, A^q, M^q)$ to the encryption oracle, either $\mathcal{E}_L[R]$ or \$, generating at most u the ciphertext blocks and q the tag blocks. Let $\sigma = q + u$ be the query complexity (the total number of calls to the underlying keyed-function in q encryption queries). Let l^i be the plaintext block-lengths of the i-th encryption query, i.e., $l^i = \lceil |M^i|/n \rceil$, where $\lceil \cdot \rceil$ is the ceiling function, M^i is the plaintext of the i-th encryption query, and $1 \leqslant i \leqslant q$. Then $u = \sum_{i=1}^{q} l^i$.

Without loss of generality, we assume that the adversary $\mathcal{A}$ is deterministic and never makes the trivial queries. We define two events as follows.

1) Event $\mathbf{E}$ (Collision-freeness): $\mathbf{E} = \{(U^i, S^i) \neq (U^j, S^j), (U^i, S^i) \neq (U^j, T^j + k^j), (U^i, T^i + k^i) \neq (U^j, T^j + k^j)$ *for all distinct* $1 \leqslant i \neq j \leqslant q, 1 \leqslant k^i \leqslant l^i, 1 \leqslant k^j \leqslant l^j\} \cup \{(U^i, S^i) \neq (U^i, T^i + k^i)$ *for all* $1 \leqslant i \leqslant q, 1 \leqslant k^i \leqslant l^i\}$.

2) Event $\mathbf{L}$ (Largest-equivalent-class): The size of U's largest equivalent class is at most α: $\mathbf{L} = \{max_i EC(U_i) \leqslant \alpha\}$, where $EC(U_i) = |\{j \in \{1, \cdots, q\} : U_j = U_i\}|$. Obviously, $\alpha \geqslant 1$ and all Us are distinct when $\alpha = 1$.

Let $\mathbf{E} \wedge \mathbf{L}$ be a *GOOD* event. The output of the oracle $\mathcal{E}_L[R]$ in the *GOOD* case is random and independent, which means that the oracle $\mathcal{E}_L[R]$ is a perfect simulation of the oracle \$ in the *GOOD* case. Therefore $Pr[\mathcal{A}^{\mathcal{E}_L[R]} \Rightarrow 1|GOOD] = Pr[\mathcal{A}^{\$} \Rightarrow 1|GOOD]$. Let $BAD = \neg GOOD = \neg(\mathbf{E} \wedge \mathbf{L}) = \neg\mathbf{E} \vee \neg\mathbf{L}$.

According to the definition of the PRIV-advantage and the total probability formula, we have

$$\begin{aligned}
Adv^{priv}_{RWCTR[R]}(\mathcal{A}) =& |Pr[\mathcal{A}^{\mathcal{E}_L[R]} \Rightarrow 1] - Pr[\mathcal{A}^{\$} \Rightarrow 1]| \\
=& |Pr[\mathcal{A}^{\mathcal{E}_L[R]} \Rightarrow 1|BAD] - Pr[\mathcal{A}^{\$} \Rightarrow 1|BAD]| \cdot Pr[BAD] \\
\leqslant& Pr[BAD] = Pr[\neg\mathbf{E} \vee \neg\mathbf{L}] \\
\leqslant& Pr[\neg\mathbf{E}] + Pr[\neg\mathbf{L}].
\end{aligned} \tag{C1}$$

Next, we need to upper bound the probability of the BAD event.

Firstly, we compute the probability of event $\neg\mathbf{L}$. α is the size of largest class of U, which means that there exists an α-collision but not $(\alpha+1)$-collision. As Us are perfectly random, the probability of event $\neg\mathbf{L}$ (there exists an $(\alpha+1)$-collision, where $\alpha \geqslant 2$) is bounded by

$$Pr[\neg\mathbf{L}] = \binom{q}{\alpha+1} 2^n / 2^{(\alpha+1)n} = \binom{q}{\alpha+1} / 2^{\alpha n} \leqslant O(q^{\alpha+1} 2^{-n\alpha}). \tag{C2}$$

If $\alpha = 2$, we have $Pr[\neg\mathbf{L}] \leqslant q^3/(3 \cdot 2^{2n+1})$. If $q \leqslant 2^{n-2}$ and $\alpha = n - 1$, we have $Pr[\neg\mathbf{L}] \leqslant 1/2^n$.

Then, we evaluate the probability of event $\neg\mathbf{E}$. Let $\mathbf{E}_i$ be the collision-free event in the i-th encryption query, where $1 \leqslant i \leqslant q$, then $\mathbf{E} = \mathbf{E}_1 \wedge \mathbf{E}_2 \wedge \cdots \wedge \mathbf{E}_q$. It follows that, $Pr[\neg\mathbf{E}] = Pr[\neg\mathbf{E}_1 \vee \neg\mathbf{E}_2 \vee \cdots \vee \neg\mathbf{E}_q]$.

Define $p_i = Pr[\neg\mathbf{E}_i | \mathbf{E}_1 \wedge \mathbf{E}_2 \wedge \cdots \wedge \mathbf{E}_{i-1}]$, by the conditional probability formula, we have

$$\begin{aligned}
Pr[\mathbf{E}] =& Pr[\mathbf{E}_1 \wedge \mathbf{E}_2 \wedge \cdots \wedge \mathbf{E}_q] \\
=& Pr[\mathbf{E}_1] Pr[\mathbf{E}_2|\mathbf{E}_1] \cdots Pr[\mathbf{E}_q|\mathbf{E}_1 \wedge \mathbf{E}_2 \wedge \cdots \wedge \mathbf{E}_{q-1}] \\
=& (1-p_1)(1-p_2)\cdots(1-p_q) = \prod_{i=1}^{q}(1-p_i).
\end{aligned}$$

By Lemma 1, we have

$$1-\sum_{i=1}^{q} p_i \leqslant Pr[\mathbf{E}] = \prod_{i=1}^{q}(1-p_i) \leqslant 1-(1-\frac{1}{e})\sum_{i=1}^{q} p_i.$$

It follows that,

$$(1-1/e)\sum_{i=1}^{q} p_i \leqslant Pr[\neg\mathbf{E}] = 1-Pr[\mathbf{E}] \leqslant \sum_{i=1}^{q} p_i.$$

Now we need to evaluate the value of p_i. Let (U^i, A^i, M^i) be the i-th encryption query of the adversary $\mathcal{A}$. Let T^i denote the authentication tag in the i-th encryption query, where $1 \leqslant i \leqslant q$. In the i-th encryption query, the inputs of the oracle R are $(U^i, S^i), (U^i, T^i+1), \cdots, (U^i, T^i+l^i) \in \{0,1\}^{2n}$.

For the i-th encryption query, we consider the following cases.

Case 1. $(U^i, S^i) = (U^j, S^j)$, where $1 \leqslant j < i \leqslant q$. By the properties of ϵ uniform A(X)U hash functions, we have $Pr[S^i = S^j] = Pr[H_L(A^i, M^i) \oplus U^i = H_L(A^j, M^j) \oplus U^j] \leqslant \epsilon$ for any $1 \leqslant j < i \leqslant q$. Then the probability of the i-th query colliding with any of the previous sequences in this case is $\sum_{j=1}^{i-1} \epsilon/2^n$.

Case 2. $(U^i, S^i) = (U^j, T^j + k^j)$, where $1 \leqslant j < i \leqslant q$ and $1 \leqslant k^j \leqslant l^j$. By the properties of ϵ uniform A(X)U hash functions, we have $Pr[S^i = T^j + k^j] = Pr[H_L(A^i, M^i) \oplus U^i = T^j + k^j] \leqslant 1/2^n$ for any $1 \leqslant j < i \leqslant q$ and $1 \leqslant k^j \leqslant l^j$. Then the probability of the i-th query colliding with any of the previous sequences in this case is $\sum_{j=1}^{i-1} l^j/2^{2n}$.

Case 3. $(U^i, S^i) = (U^i, T^i + k^i)$, where $1 \leqslant i \leqslant q$ and $1 \leqslant k^i \leqslant l^i$. By the properties of ϵ uniform A(X)U hash functions, we have $Pr[S^i = T^i + k^i] = Pr[H_L(A^i, M^i) \oplus U^i = T^i + k^i] \leqslant 1/2^n$ for any $1 \leqslant i \leqslant q$ and $1 \leqslant k^i \leqslant l^i$. Then the probability of collision in the i-th query is $l^i/2^n$.

Case 4. $(U^i, T^i + k^i) = (U^j, S^j)$, where $1 \leqslant j < i \leqslant q$ and $1 \leqslant k^i \leqslant l^i$. By the properties of ϵ uniform A(X)U hash functions, we have $Pr[T^i + k^i = S^j] = Pr[H_L(A^j, M^j) \oplus U^j = T^i + k^i] \leqslant 1/2^n$ for any $1 \leqslant j < i \leqslant q$ and $1 \leqslant k^i \leqslant l^i$. Then the probability of the i-th query colliding with any of the previous query in this case is $\sum_{j=1}^{i-1} l^i/2^{2n}$.

Case 5. $(U^i, T^i + k^i) = (U^j, T^j + k^j)$, where $1 \leqslant j < i \leqslant q, 1 \leqslant k^i \leqslant l^i$, and $1 \leqslant k^j \leqslant l^j$. As $Pr[U^i = U^j] = 1/2^n$ and $Pr[T^i + k^i = T^j + k^j] = 1/2^n$, therefore the probability of the i-th query colliding with any of the previous query in this case is $\sum_{j=1}^{i-1}(l^i \cdot l^j)/2^{2n}$.

Summarizing above all cases, the probability of event $\neg\mathbf{E}$ is

$$\begin{aligned} Pr[\neg\mathbf{E}] \leqslant & \sum_{i=1}^{q} p_i = \sum_{i=1}^{q}\left(\sum_{j=1}^{i-1}\frac{\epsilon}{2^n} + \sum_{j=1}^{i-1}\frac{(l^j + l^i + l^i \cdot l^j)}{2^{2n}} + \frac{l^i}{2^n}\right) \\ = & \sum_{i=1}^{q}\sum_{j=1}^{i-1}\left(\frac{\epsilon}{2^n} + \frac{(l^j + l^i + l^i \cdot l^j)}{2^{2n}}\right) + \sum_{i=1}^{q}\frac{l^i}{2^n} \\ \leqslant & \frac{q^2\epsilon}{2^{n+1}} + \frac{2qu+u^2}{2^{2n+1}} + \frac{u}{2^n}. \end{aligned} \tag{C3}$$

Combining Eqs. (C1), (C2), and (C3), the PRIV-advantage of $\mathcal{A}$ against RWCTR[R] is

$$Adv^{priv}_{RWCTR[R]}(\mathcal{A}) \leqslant \binom{q}{\alpha+1}/2^{\alpha n} + \frac{q^2\epsilon}{2^{n+1}} + \frac{2qu+u^2}{2^{2n+1}} + \frac{u}{2^n}.$$

As $q + u = \sigma$, by $0 < \frac{\sigma^2}{2^{2n}} \leqslant \frac{\sigma}{2^n} \leqslant 1$ and Lemma 2, we have

$$Adv^{priv}_{RWCTR[R]}(\mathcal{A}) \leqslant \binom{q}{\alpha+1}\frac{1}{2^{\alpha n}} + \frac{q^2\epsilon}{2^{n+1}} + \frac{3\sigma}{2^{n+1}}.$$

If $\alpha = 2$ and $1/2^n \leqslant \epsilon \leqslant 1/2^{n-1}$, then $\binom{q}{\alpha+1}/2^{\alpha n} \leqslant q^3/(3 \cdot 2^{2n+1}) \leqslant q^{3/2}/2^{n+1}$, $q^2\epsilon/2^{n+1} \leqslant q/2^n \leqslant q^{3/2}/2^n$, and $\sigma/2^n \leqslant \sigma^{3/2}/2^n$. Therefore, the above bound implies $3(\sigma+q)^{3/2}/2^{n+1}$ when $q \leqslant 2^{2n/3}$. This achieves BBB-security. If $q \leqslant 2^{n-2}$ and $\alpha = n-1$, the above bound implies $q^2\epsilon/2^{n+1} + 3\sigma/2^{n+1}$ when $q \leqslant min\{2^{n-2}, \sqrt{2^n\epsilon^{-1}}\}$. This achieves close-to-optimal security. The proof of privacy is finished.

Authenticity. Let $\mathcal{A}$ be an adversary, which has access to an encryption oracle $\mathcal{E}_L[R]$ and a decryption oracle $\mathcal{D}_L[R]$, where $L \in \mathcal{K}_h$. Firstly, the adversary $\mathcal{A}$ makes q queries $\{(U^i, A^i, M^i)\}_{i=1}^q$ to the encryption oracle $\mathcal{E}_L[R]$ and obtains $\{(C^i, T^i) = \mathcal{E}_L[R](U^i, A^i, M^i)\}_{i=1}^q$. Then the adversary $\mathcal{A}$ queries a challenge pair $(U', A', C', T') \notin \{(U^i, A^i, C^i, T^i)\}_{i=1}^q$ to the decryption oracle $\mathcal{D}_L[R]$. By the definition of the AUTH-advantage, we have

$$Adv^{auth}_{RWCTR[R]}(\mathcal{A}) = Pr[\mathcal{A}^{\mathcal{E}_L[R], \mathcal{D}_L[R]} \; forges].$$

Let $\mathbf{F}$ be an event that the forgery attempt succeeds, then $Adv^{auth}_{RWCTR[R]}(\mathcal{A}) = Pr[\mathbf{F}]$. By the total probability formula, we can obtain

$$\begin{aligned} Pr[\mathbf{F}] = & Pr[\mathbf{F}|\neg(\mathbf{E} \wedge \mathbf{L})]Pr[\neg(\mathbf{E} \wedge \mathbf{L})] + Pr[\mathbf{F}|(\mathbf{E} \wedge \mathbf{L})]Pr[\mathbf{E} \wedge \mathbf{L}] \\ \leqslant & Pr[\neg(\mathbf{E} \wedge \mathbf{L})] + Pr[\mathbf{F}|(\mathbf{E} \wedge \mathbf{L})]. \end{aligned}$$

First, $Pr[\neg(\mathbf{E}\wedge\mathbf{L})]$ is at most $\binom{q}{\alpha+1}/2^{\alpha n}+q^2\epsilon/2^{n+1}+3\sigma/2^{n+1}$ as shown in the privacy proof. Note that σ is the query complexity of the encryption and decryption queries. Then we need to bound the probability of an event $\mathbf{F}|(\mathbf{E}\wedge\mathbf{L})$. We analyze some cases as follows.

Case 1: $T'\notin\{T^1,T^2,\cdots,T^q\}$, i.e., T' is new. The probability of guessing this correctly in this case is at most $1/(2^n-q)$, which is its success probability of $\mathcal{A}$.

Case 2: $T'\in\{T^1,T^2,\cdots,T^q\}$, i.e., T' is old. We consider the following two cases.

Case 2-1: $(U',S')\notin\{(U^1,S^1),\cdots,(U^q,S^q)\}$, i.e., (U',S') is new, where $S',S^1,\cdots,S^q$ are generated from an ϵ uniform AXU hash function H. Then the probability of guessing this correctly in this case is at most $1/2^n$, which is its success probability of $\mathcal{A}$.

Case 2-2: $(U',S')\in\{(U^1,S^1),\cdots,(U^q,S^q)\}$, i.e., (U',S') is old, where $S',S^1,\cdots,S^q$ are generated from an ϵ uniform AXU hash function H. As any U_i's equivalent class is of size at most α, then the probability of guessing this correctly in this case is at most $\alpha\epsilon$, which is its success probability of $\mathcal{A}$.

Summarizing above all mutually exclusive cases, we have

$$Pr[\mathbf{F}|(\mathbf{E}\wedge\mathbf{L})]\leqslant max\{\frac{1}{2^n-q},\frac{1}{2^n},\alpha\epsilon\}\leqslant\alpha\epsilon+\frac{1}{2^n-q}\leqslant\alpha\epsilon+\frac{2}{2^n},$$

for a single forgery query.

Then the success probability of the single forgery attempt, i.e., the AUTH-advantage of $\mathcal{A}$ against RWCTR[R], is upper bounded by

$$Adv^{auth}_{RWCTR[R]}(\mathcal{A})=Pr[\mathbf{F}]\leqslant\binom{q}{\alpha+1}/2^{\alpha n}+q^2\epsilon/2^{n+1}+3\sigma/2^{n+1}+\alpha\epsilon+\frac{2}{2^n}.$$

If the adversary $\mathcal{A}$ makes q_v forgery queries after q encryption queries, the AUTH-advantage of $\mathcal{A}$ against RWCTR[R] is easily upper bounded by

$$Adv^{auth}_{RWCTR[R]}(\mathcal{A})\leqslant\binom{q}{\alpha+1}/2^{\alpha n}+q^2\epsilon/2^{n+1}+3\sigma/2^{n+1}+q_v\alpha\epsilon+\frac{2q_v}{2^n}.$$

If $\alpha=2$ and $1/2^n\leqslant\epsilon\leqslant1/2^{n-1}$, then the above bound implies $3(\sigma+q)^{3/2}/2^{n+1}+2q_v\epsilon+2q_v/2^n$ when $q\leqslant2^{2n/3}$. This achieves BBB-security. If $q\leqslant2^{n-2}$ and $\alpha=n-1$, then the above bound implies $q^2\epsilon/2^{n+1}+3\sigma/2^{n+1}+q_v(n-1)\epsilon+2q_v/2^n$ when $q\leqslant min\{2^{n-2},\sqrt{2^n\epsilon^{-1}}\}$. This achieves close-to-optimal security. We conclude the proof of Lemma 4.

Appendix D RWCTRN: the first practical keyed-function-based BBB-secure AE mode

Appendix D.1 RWCTRN

The IV-based AE schemes can be divided into three types: random IV-based AE schemes, nonce IV-based AE schemes, and arbitrary IV-based AE schemes [4]. The adversary has no control over the choice of IV for each encryption in the random IV-based AE schemes. Therefore, this scenario is impractical. The adversary can choose but does not repeat nonce IV values in all encryption queries in the nonce IV-based AE schemes. While, there are no restrictions on the arbitrary IV-based AE schemes. Thus an adversary may choose any IV for each encryption. Deterministic AE (DAE) schemes and online AE (OAE) schemes belong to arbitrary IV-based AE schemes. DAE schemes do not even utilize an IV input, in which case an IV can be embedded into the associated data.

We utilize a random IV U in RWCTR. However, in the real world, the random IV is not practical. Therefore, we present a more practical AE mode RWCTRN, which is based on a nonce IV or an arbitrary IV N.

RWCTRN consists of an encryption algorithm $\mathcal{E}$ and a decryption algorithm $\mathcal{D}$. Let $\mathcal{K}_h$ and $\mathcal{K}_f$ be two non-empty sets of keys. The encryption algorithm $\mathcal{E}$ takes two keys $(L,K)\in\mathcal{K}_h\times\mathcal{K}_f$, an n-bit nonce or arbitrary IV $N\in\{0,1\}^n$, an associated data $A\in\{0,1\}^*$, and a plaintext $M\in\{0,1\}^*$ as input, and returns a ciphertext $C\in\{0,1\}^*$ and an authentication tag $T\in\{0,1\}^n$. The decryption algorithm $\mathcal{D}$ is the inverse of the encryption algorithm $\mathcal{E}$. It takes two keys $(L,K)\in\mathcal{K}_h\times\mathcal{K}_f$, an n-bit nonce or arbitrary IV $N\in\{0,1\}^n$, an associated data $A\in\{0,1\}^*$, a ciphertext $C\in\{0,1\}^*$ and an authentication tag $T\in\{0,1\}^n$ as input, and returns a plaintext $M\in\{0,1\}^*$ or an invalid symbol $\perp$. The encryption and decryption algorithms are presented in Figure D1.

Appendix D.2 The security proof of Theorem 2

The security proof of Theorem 2 is similar to that of Theorem 1 except that we need to consider the collisions of $(N,0)$ and (U,S) or $(U,T+k)$, where $1\leqslant k\leqslant l$ and l is the maximum block-length of the plaintext in all queries. We derive the abbreviated proofs as follows.

In the nonce IV scenario, all Ns are distinct. Therefore, by Definition 1 and Lemma 2, we have $Pr[(N,0)=(U,S)]\leqslant q^2/2^{2n}\leqslant q/2^n$ and $Pr[(N,0)=(U,T+k)]\leqslant qu/2^{2n}\leqslant\sigma^2/2^{2n+2}\leqslant\sigma/2^{n+1}$, where $q+u=\sigma$. Adding the results of Theorem 1, we can easily upper bound the privacy and authenticity of RWCTRN in the nonce IV scenario.

However, in the arbitrary IV scenario, the adversary can choose any multiply IV values such that $N^i=N^j$, where $1\leqslant i\neq j\leqslant q$. It follows that, $U^i=U^j$ for any $1\leqslant i\neq j\leqslant q$. Therefore, $\alpha=q$ in this case and $Pr[\neg\mathbf{L}]=0$. Then, according to Lemma 4, for $1\leqslant j<i\leqslant q$, we have $Pr[(U^i,S^i)=(U^j,S^j)]\leqslant q^2\epsilon/2$, $Pr[(U^i,S^i)=(U^j,T^j+k^j)]\leqslant qu/2^{n+1}$, $Pr[(U^i,S^i)=(U^i,T^i+k^i)]\leqslant u/2^n$, $Pr[(U^i,T^i+k^i)=(U^j,S^j)]\leqslant qu/2^{n+1}$, and $Pr[(U^i,T^i+k^i)=(U^j,T^j+k^j)]\leqslant u^2/2^{n+1}$. Besides that, we add $Pr[(N,0)=(U,S)]\leqslant q/2^n\ll q^2/2^n$ and $Pr[(N,0)=(U,T+k)]\leqslant\sigma/2^{n+1}\ll\sigma^2/2^{n+1}$ to obtain the probability of event $\neg\mathbf{E}$. By Lemma 4, we can easily upper bound the privacy and authenticity of RWCTRN in the arbitrary IV scenario.

Algorithm D1 The encryption algorithm of RWCTRN

Input: two keys (L, K), an n-bit nonce IV or arbitrary IV N, an associated data A, and a plaintext M
Output: a ciphertext C and an authentication tag T

Partition M into $M_1\|\cdots\|M_l$, $|M_i| = n, 1 \leqslant i \leqslant l-1, 0 < |M_l| \leqslant n$
$U = F_K(N, 0^n)$
$S = H_L(A, M) \oplus U$
$T = F_K(U, S)$
for $i = 1$ to $l-1$ **do**
    $S_i = F_K(U, T+i)$
    $C_i = M_i \oplus S_i$
end for
$S_l = F_K(U, T+l)$
$C_l = M_l \oplus (S_l)_{|M_l|}$
$C = C_1\|C_2\|\cdots\|C_l$
return $(C\|T)$

Algorithm D2 The decryption algorithm of RWCTRN

Input: two keys (L, K), an n-bit nonce IV or arbitrary IV N, an associated data A, a ciphertext C, and an authentication tag T
Output: a plaintext M or $\perp$

Partition C into $C_1\|C_2\|\cdots\|C_l$, $|C_i| = n, 1 \leqslant i \leqslant l-1, 0 < |C_l| \leqslant n$
$U = F_K(N, 0^n)$
for $i = 1$ to $l-1$ **do**
    $S_i = F_K(U, T+i)$
    $M_i = C_i \oplus S_i$
end for
$S_l = F_K(U, T+l)$
$M_l = C_l \oplus (S_l)_{|C_l|}$
$M = M_1\|\cdots\|M_l$
$S = H_L(A, M) \oplus U$
$T' = F_K(U, S)$
if $T' = T$ **then**
    return M
else
    return $\perp$ (INVALID)
end if

Figure D1 The encryption and decryption algorithms of RWCTRN

Table E1 Comparison of AE schemes based on various primitives. Let n be the block-size and r be an integer.

	# Keys	IV type	Primitive[1)]	Assumption[2)]	Privacy	Authenticity	Reference
GCM	1	Nonce	E	PRP	$\frac{n}{2}$-bit	$\frac{n}{2}$-bit	[5]
SIV	2	Arbitrary	E	PRP	$\frac{n}{2}$-bit	$\frac{n}{2}$-bit	[6]
BTM	1	Arbitrary	E	PRP	$\frac{n}{2}$-bit	$\frac{n}{2}$-bit	[7]
CHM	1	Nonce	E	PRP	$\frac{2n}{3}$-bit	$\frac{2n}{3}$-bit	[8]
GCM-SIV$_r$	$3r$	Arbitrary	E	PRP	$\frac{rn}{r+1}$-bit	$\frac{rn}{r+1}$-bit	[9]
TAE	1	Nonce	$\widetilde{E}$	TPRP	$\frac{n}{2}$-bit	$\frac{n}{2}$-bit	[10]
SCT	1	Nonce & Arbitrary	$\widetilde{E}$	TPRP	n & $\frac{n}{2}$-bit	n & $\frac{n}{2}$-bit	[11]
SIVx	1	Arbitrary	$\widetilde{E}$	TPRP	n-bit	n-bit	[12]
OMD	1	Nonce	F	PRF	$\frac{n}{2}$-bit	$\frac{n}{2}$-bit	[13]
PMR-OMD	1	Arbitrary	F	PRF	$\frac{n}{2}$-bit	$\frac{n}{2}$-bit	[14]
p-OMD	1	Nonce	F	PRF	$\frac{n}{2}$-bit	$\frac{n}{2}$-bit	[15]
RWCTR	2	Random	F	PRF	$(n-2)$-bit	$(n-\log n)$-bit	This paper
RWCTRN	2	Nonce & Arbitrary	F	PRF	$n-2$ & $\frac{n}{2}$-bit	$n-\log n$ & $\frac{n}{2}$-bit	This paper

Appendix E Comparisons of AE schemes in security and efficiency

RWCTR and RWCTRN are designed as AE modes for a compression function. The motivation of our designs is manifold: 1) the secure hash algorithm (SHA) family has been widely applied to various cryptographic schemes or protocols and these functions can be efficiently implemented; 2) New instructions support performance acceleration of SHA-256 on Intel architecture processors; 3) AE schemes based on different primitives support various settings. The block cipher is used as a primitive, which is suit for the environments with the short block-size, such as DES with the block-size $n = 64$ and AES with the block-size $n = 128$. The TBC is used as a primitive, which is still an ideal cipher. In the real world, most of TBCs are indirectly constructed by block ciphers. The compression function is used as a primitive, which is suit for the environments with the long or short block-size, such as SHA-256 with the block-size $n = 256$ and SHA-512 with the block-size $n = 512$. Therefore, a keyed-function-based AE scheme can provide a stronger security guarantee than a blockcipher-based AE scheme in some environments with the long block-size. The comparison of AE schemes with various primitives is shown in Table E1.

From the perspective of security, RWCTR enjoys at most about $(n-2)$-bit security for adversarial queries and at most

1) "Primitive" shows the components of the scheme. "E" stands for the block cipher, "$\widetilde{E}$" stands for the tweakable blockcipher, and "F" stands for the keyed compression function.

2) "Assumption" shows the security assumption of the components. "PRP" denotes pseudorandom permutation, "T-PRP" denotes tweakable PRP, and "PRF" denotes pseudorandom function.

Table E2 Efficiency of keyed-function-based AE modes for a-block associated data and l-block message.

	# Primitive calls[3)]	# Multiplications[4)]	# Computations[5)]	Reference
OMD	$a+l+2$	$a+l+4$	$2a+2l+6$	[13]
PMR-OMD	$a+2l+4$	$a+l+4$	$2a+3l+8$	[14]
p-OMD	$max\{a,l\}+2$	$max\{a,l\}+2$	$2max\{a,l\}+4$	[15]
RWCTR	$l+1$	$a+l+1$	$a+2l+2$	This paper
RWCTRN	$l+2$	$a+l+1$	$a+2l+3$	This paper

about $(n-\log n)$-bit security for forgery attempts, while RWCTRN enjoys at most about $(n-2)$-bit security for adversarial queries and at most about $(n-\log n)$-bit security for forgery attempts in the nonce IV scenario and offers at most about $n/2$-bit security for adversarial queries and at most about $n/2$-bit security for forgery attempts in the arbitrary IV scenario. Specifically, the privacy of RWCTR ensures at most about $(n-2)$-bit security and the authenticity of RWCTR ensures at most about $(n-\log n)$-bit security. While, the privacy of RWCTRN ensures at most about $(n-2)$-bit security and the authenticity of RWCTRN ensures at most about $(n-\log n)$-bit security in the nonce IV scenario, and the privacy and authenticity of RWCTRN guarantee at most about $n/2$-bit security in the arbitrary IV scenario. From the perspective of efficiency, RWCTR (resp. RWCTRN) invokes $l+1$ (resp. $l+2$) underlying keyed-functions and utilizes $a+l+1$ finite-field multiplications, where l and a are the block-lengths of the message and the associated data, respectively. From the perspective of security and efficiency, RWCTR and RWCTRN are better than other keyed-function-based AE schemes. The related results are shown in Tables E1 and E2.

Appendix F Instantiations

The instantiation of the underlying keyed compression function $F_K:\{0,1\}^{2n}\rightarrow\{0,1\}^n$ is very easy, where $K\in\mathcal{K}_f$. Given a keyless hash function $G:\{0,1\}^n\times\{0,1\}^b\rightarrow\{0,1\}^n$, we can convert it to a keyed compression function $F_K:\{0,1\}^{2n}\rightarrow\{0,1\}^n$ by $F_K(\cdot,\cdot)=G(\cdot,K||0^{b-n-|K|}||\cdot)$, where $b\geqslant n+|K|$.

Given a compression function SHA-256: $\{0,1\}^{256}\times\{0,1\}^{512}\rightarrow\{0,1\}^{256}$, we have $b=512, n=256$, and $|K|\leqslant 256$. Then RWCTR[SHA-256] enjoys at most about 254-bit security for adversarial queries and at most about 248-bit security for forgery attempts, while RWCTRN[SHA-256] enjoys at most about 254-bit security for adversarial queries and at most about 248-bit security for forgery attempts in the nonce IV scenario, and offers at most about 128-bit security for adversarial queries and at most about 128-bit security for forgery attempts in the arbitrary IV scenario. In other words, the privacy of RWCTR[SHA-256] ensures at most about 254-bit security and the authenticity of RWCTR[SHA-256] ensures at most about 248-bit security. While, the privacy of RWCTRN[SHA-256] ensures at most about 254-bit security and the authenticity of RWCTRN[SHA-256] ensures at most about 248-bit security in the nonce IV scenario, and the privacy and authenticity of RWCTRN[SHA-256] guarantee at most about 128-bit security in the arbitrary IV scenario.

Given a compression function SHA-512: $\{0,1\}^{512}\times\{0,1\}^{1024}\rightarrow\{0,1\}^{512}$, we have $b=1024, n=512$, and $|K|\leqslant 512$. Then RWCTR[SHA-512] enjoys at most about 510-bit security for adversarial queries and at most about 503-bit security for forgery attempts, while RWCTRN[SHA-512] enjoys at most about 510-bit security for adversarial queries and at most about 503-bit security for forgery attempts in the nonce IV scenario, and offers at most about 256-bit security for adversarial queries and at most about 256-bit security for forgery attempts in the arbitrary IV scenario. In other words, the privacy of RWCTR[SHA-512] ensures at most about 510-bit security and the authenticity of RWCTR[SHA-512] ensures at most about 503-bit security. While, the privacy of RWCTRN[SHA-512] ensures at most about 510-bit security and the authenticity of RWCTRN[SHA-512] ensures at most about 503-bit security in the nonce IV scenario, and the privacy and authenticity of RWCTRN[SHA-512] guarantee at most about 256-bit security in the arbitrary IV scenario.

References

1 Kurosawa K. Power of a public random permutation and its application to authenticated encryption. IEEE Trans Inf Theory, 2010, 5(10): 5366–5374

2 Wang P, Li Y, Zhang L, et al. Related-key almost universal hash functions: definitions, constructions and applications. In: Proceedings of 23th International Workshop on Fast Software Encryption, Bochum, 2016. 514–532

3 Minematsu K. How to thwart birthday attacks against MACs via small randomness. In: Proceedings of 17th International Workshop on Fast Software Encryption, Seoul, 2010. 230–249

4 Andreeva E, Bogdanov A, Luykx A, et al. How to securely release unverified plaintext in authenticated encryption. In: Proceedings of 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, 2014. 105–125

5 NIST. Recommendation for block cipher modes of operation: Galois Counter Mode (GCM) and GMAC. NIST SP 800-38D, 2007. Available from: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

3) "# Primitive calls" shows the number of calling the underlying primitive.

4) "# Multiplications" shows the number of invoking the finite-field multiplications.

5) "# Computations" is the sum of "# Primitive calls" and "# Multiplications", which reflects the efficiency of AE schemes.

6 Rogaway P, Shrimpton T. A provable security treatment of the key wrap problem. In: Proceedings of 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, 2006. 373–390
7 Iwata T, Yasuda K. BTM: a single-key, inverse-cipher-free mode for deterministic authenticated encryption. In: Proceedings of 16th Annual International Workshop on Selected Areas in Cryptography, Calgary, 2009. 313–330
8 Iwata T. New blockcipher modes of operation with beyond the birthday bound security. In: Proceedings of 13th International Workshop on Fast Software Encryption, Graz, 2006. 310–327
9 Iwata T, Minematsu K. Stronger security variants of GCM-SIV. IACR Transactions on Symmetric Cryptology, 2016, 2016(1): 134–157
10 Liskov M, Rivest R L, Wagner D. Tweakable block ciphers. J Cryptol, 2011, 24(3): 588–613
11 Peyrin T, Seurin Y. Counter-in-tweak: authenticated encryption modes for tweakable block ciphers. In: Proceedings of 36th Annual International Cryptology Conference, Santa Barbara, 2016. 33–63
12 List E, Nandi M. Revisiting full-PRF-secure PMAC and using it for beyond-birthday authenticated encryption. In: Proceedings of the Cryptographer's Track at the RSA Conference, San Francisco, 2017. 258–274
13 Cogliani S, Maimuţ D Ş, Naccache D. OMD: a compression function mode of operation for authenticated encryption. In: Proceedings of 21th International Conference on Selected Areas in Cryptography, Montreal, 2014. 112–128
14 Reyhanitabar R, Vaudenay S, Vizár D. Misuse-resistant variants of the OMD authenticated encryption mode. In: Proceedings of 8th International Conference on Provable Security, Hong Kong, 2014. 55–70
15 Reyhanitabar R, Vaudenay S, Vizár D. Boosting OMD for almost free authentication of associated data. In: Proceedings of 22th International Workshop on Fast Software Encryption, Istanbul, 2015. 411–427