Tweakable Blockciphers for Efficient Authenticated Encryptions with Beyond the Birthday-Bound Security

Yusuke Naito

Mitsubishi Electric Corporation, Kanagawa, Japan
Naito.Yusuke@ce.MitsubishiElectric.co.jp

Abstract. Modular design via a tweakable blockcipher (TBC) offers efficient authenticated encryption (AE) schemes (with associated data) that call a blockcipher once for each data block (of associated data or a plaintext). However, the existing efficient blockcipher-based TBCs are secure up to the birthday bound, where the underlying keyed blockcipher is a secure strong pseudorandom permutation. Existing blockcipher-based AE schemes with beyond-birthday-bound (BBB) security are not efficient, that is, a blockcipher is called twice or more for each data block.

In this paper, we present a TBC, XKK, that offers efficient blockcipher-based AE schemes with BBB security, by combining with efficient TBC-based AE schemes such as ΘCB3 and OTR. XKK is a combination of two TBCs, Minematsu’s TBC and Liskov et al.’s TBC. In the XKK-based AE schemes, a nonce and a counter are taken as tweak; a nonce-dependent blockcipher’s key is generated by using a pseudo-random function \( F \) (from Minematsu); a counter is inputted to an almost xor universal hash function, and the hash value is xor-ed with the input and output blocks of a blockcipher with the nonce-dependent key (from Liskov et al.). For each query to the AE scheme, after the nonce-dependent key is generated, it can be reused, thereby a blockcipher is called once for each data block. We prove that the security bounds of the XKK-based AE schemes become roughly \( \ell^2 q/n \), where \( q \) is the number of queries to the AE scheme, \( n \) is the blockcipher size, and \( \ell \) is the number of blockcipher calls in one AE evaluation. Regarding the function \( F \), we present two blockcipher-based instantiations, the concatenation of blockcipher calls, \( F^{(1)} \), and the xor of blockcipher calls, \( F^{(2)} \), where \( F^{(1)} \) calls a blockcipher \( i + 1 \) times. By the PRF/PRP switch, the security bounds of the XKK-based AE schemes with \( F^{(1)} \) become roughly \( \ell^2 q/2^n + q^2/2^n \), thus if \( \ell \ll 2^{n/2} \) and \( q \ll 2^{n/2} \), these achieve BBB security. By the xor construction, the security bounds of the XKK-based AE schemes with \( F^{(2)} \) become roughly \( \ell^2 q/2^n + q/2^n \), thus if \( \ell \ll 2^{n/2} \), these achieve BBB security.

Keywords: Blockcipher · tweakable blockcipher · efficient authenticated encryption · beyond-birthday-bound security

1 Introduction

Confidentiality and authenticity of data are the most important properties to securely communicate over an insecure channel. In the symmetric-key setting, an authenticated encryption (AE) scheme (with associated data) ensures jointly these properties. AE schemes have been mainly designed from a blockcipher, and designing an efficient AE scheme is a main theme in AE research. In efficient schemes such as OCB3 [KR11] and OTR [Min14], a blockcipher is called once for each data block\(^1\) (for associated data or a

\(^1\) The size of the data block is equal to the block size of the blockcipher.
plaintext). Such AE schemes that we call \textit{efficient} AE schemes\footnote{The efficiency of AE schemes is often measured by “rate” that takes all blockcipher calls including a precomputation phase into consideration. For example, the most efficient AE scheme is rate-1, where the number of blockcipher calls in one AE procedure is the number of data blocks plus 2 (the number of blockcipher calls for a tag and for a nonce in a precomputation phase). On the other hand, the term “efficient” considered in this paper does not take the precomputation phase into account.} have been designed via a \textit{tweakable blockcipher}.

Tweakable blockcipher (TBC) whose concept was introduced by Liskov et al. \cite{LRW02} is a generalization of classical blockcipher. An encryption by a TBC takes an input called tweak in addition to a key and a plaintext. Tweak is a public parameter, where retweaking (changing the tweak value) offers the same functionality as changing its secret key but should be less costly. An efficient blockcipher-based AE scheme is obtained by (1) designing an efficient TBC, that is, a blockcipher is called once; (2) designing an efficient TBC-based AE scheme, that is, a TBC is called once for each data block; (3) combining (1) and (2).

Regarding security, existing efficient blockcipher-based AE schemes are secure up to the \textit{birthday bound}. However, birthday-bound security sometimes becomes unreliable, for example, when a lightweight blockcipher is used, when large amounts of data are processed, or when a large number of connections need to be kept secure. In (2) efficient TBC-based AE schemes with \textit{beyond-birthday-bound} (BBB) security have been proposed such as ΘCB3 \cite{KR11} and OTR \cite{Min14}, whereas in (1) existing efficient TBCs are secure up to the birthday bound. In order to obtain efficient BBB-secure AE schemes in (3), one needs to design a BBB-secure TBC in (1). Note that in (1), a keyed blockcipher is assumed to be a secure strong-pseudo-random permutation (SPRP),\footnote{A blockcipher with a random key is indistinguishable from a random permutation (RP).} and in (2), a keyed TBC is assumed to be a secure tweakable SPRP (TSPRP).\footnote{A TBC with a random key is indistinguishable from a tweakable RP.}

Hereafter, $E_{K_{E}}$ denotes a blockcipher with $n$-bit block having a $k$-bit key $K_{E}$.

\subsection{Existing Efficient TBCs with Birthday-Bound Security}

Liskov et al. \cite{LRW02} proposed an efficient blockcipher-based TBC called LRW2 that has an Even-Mansour-style structure \cite{EM97}, where a tweak is taken by an almost xor universal (AXU) hash function, and a plaintext and a ciphertext of the underlying blockcipher are xor-ed with the hash value. The encryption of LRW2 is illustrated in Figure 1 (Left), where $tw$ is a tweak, $m$ is a plaintext block, $c$ is a ciphertext block, and $h_{K_{h}}$ is an AXU hash function with a key $K_{h}$ that accepts $tw$ and returns an $n$-bit value. Regarding AXU hash functions, several efficient instantiations have been proposed such as powering-up scheme \cite{Rog04}, gray-code-based scheme \cite{KR11, RBBK01} and LFSR-based scheme \cite{CS08, GJMN16}. It was proven that LRW2 is a secure TSPRP up to the birthday bound ($2^{n/2}$ queries) \cite{LRW02}.

\begin{figure}[h]
\centering
\begin{align*}
\text{LRW2:} & \quad m \xrightarrow{tw} h_{K_{h}} \xrightarrow{E_{K_{E}}} c \\
\text{Min:} & \quad m \xrightarrow{F_{K_{E}}} tw \xrightarrow{c}
\end{align*}
\caption{LRW2 (left) and Min (right).}
\end{figure}
1.2 BBB-Secure TBCs

So far, several BBB-secure TBCs have been proposed. Minematsu [Min09] designed a TBC denoted by \( \text{Min} \). In \( \text{Min} \), a tweak-dependent key is defined by using a pseudo-random function (PRF), and a plaintext is encrypted by a blockcipher with the tweak-dependent key. The encryption of \( \text{Min} \) is illustrated in Figure 1 (right), where \( F_{K_F}(tw) = E_{K_F}(tw) \), where \( k = n \). He proved that \( \text{Min} \) is a secure TSPRP up to \( \max\{2^{n/2}, 2^n/N_{tw}\} \) queries, where \( N_{tw} \) is the number of distinct tweaks in the queries. Thus if \( N_{tw} < 2^{n/2} \), \( \text{Min} \) achieves BBB security. Landecker et al. [LST12] proposed a TBC called Chained LRW2 (CLRW2), where LRW2 is iterated twice. They proved that CLRW2 is a secure TSPRP up to \( 2^{2n/3} \) queries. Lampe and Seurin [LS13] considered a more general scheme called \( r \)-CLRW where LRW2 is iterated \( r \) times. They proved that \( r \)-CLRW is a secure TSPRP up to \( 2^{rn/(r+2)} \) queries.

1.3 Open Problem

In \( \ThetaCB3 \) and \( \ThetaTR \) that are efficient nonce-based and TBC-based AE schemes, a plaintext block is encrypted by a TBC that takes a nonce and a counter as tweak, where a nonce is changed for each query, and a counter is changed for each data block. Hence, a tweak is changed for every TBC call. Incorporating \( \text{Min} \) into these AE schemes, for each data block, the resultant schemes call a blockcipher twice, and perform the key scheduling once. Since the same tweak is not repeated, \( N_{tw} = 2^{n/2} \) after \( 2^{n/2} \) TBC calls, and thus the security bound falls into the birthday one (security up to \( 2^{n/2} \) queries). Incorporating CLRW2 into these AE schemes, the resultant schemes achieve BBB security (security up to \( 2^{2n/3} \) queries) but call a blockcipher twice for each data block. Similarly, incorporating \( r \)-CLRW into these AE schemes, the resultant schemes achieve BBB security (security up to \( 2^{rn/(r+2)} \) queries) but call a blockcipher \( r \) times for each data block.

Several blockcipher-based AE schemes have been proposed, which are either efficient or BBB-secure but not both. Existing efficient blockcipher-based AE schemes [RBBK01, Rog04, KR11, Min14] are secure up to the birthday bound. Iwata [Iwa08] proposed an AE scheme that is secure up to \( 2^{2n/3} \) blockcipher calls. In the default setting of the AE scheme, for each 4 data blocks, it requires 6 blockcipher calls, and for each data block, it requires one multiplication. Iwata and Yasuda [IY09a, IY09b] pointed out that a combination of the xor of keyed blockciphers [Luc00] and the Feistel network with six rounds [Pat04] becomes a BBB-secure SPRP, thus offers BBB-secure AE schemes. However, the resultant AE schemes require 6 blockcipher calls for each data block. Iwata and Minematsu [IM16] proposed AE schemes that are secure up to \( 2^{rn/(r+1)} \) blockcipher calls for a parameter \( r \). In the encryption procedure, for each data block, a blockcipher is called \( r \) times. A tag is generated by using \( r \) AXU-hash functions.

As mentioned above, there is no efficient blockcipher-based AE scheme with BBB security. Therefore, the following question arises: Can we design a TBC that offers efficient blockcipher-based AE schemes with BBB security?

1.4 Our Results

We present TBCs that offer efficient AE schemes with BBB security.

1.4.1 Basic Construction

Our TBCs are based on Minematsu’s TBC Min. In order to avoid the frequent key scheduling, we separate counters from the tweak function \( F_{K_F} \), and instead use the tweak function of LRW2 \( h_{K_h} \) to take counters. The basic construction of our TBCs that we call XKX is illustrated in Figure 2 (left), where \((N, \text{ctr})\) is a pair of tweaks such that the first
tweak \( N \) becomes a nonce and the second one \( ctr \) becomes a counter. We prove that XKK is a secure TSPRP as long as the keyed blockcipher is a secure SPRP, the keyed function is a secure PRF, and the keyed hash function is AXU. The security bound is roughly \( \ell^2 q/2^n \) + \( q \times \) (the SPRP-security advantage for \( E \)) + (the PRF-security advantage for \( F \)), where \( q \) is the number of distinct first tweaks (nonces in AE schemes), and \( \ell \) is the number of queries with the same first tweak (the number of blockcipher calls in one AE evaluation). If the SPRP-security advantage becomes roughly \( \ell q/2^k \) (an adversary conducts a naive brute-force attack, see [BKR98]), the security bound becomes \( \ell^2 q/2^n + \ell q^2/2^k + q^2/2^n \). XKK with \( F^{(1)} \) is denoted by XKK\(^{(1)}\).

- The second instantiation \( F^{(2)} \), in order to remove the PRF/PRP-switch term \( q^2/2^n \), uses an xor function of a blockcipher shown in Figure 2 (right), where \( N \) is an \((n-2)\)-bit tweak, and \( K_E \) is a \( k \)-bit key. The PRF-security of the xor function was analyzed in [Pat10, IMV16], where the PRF-bound is roughly \( q/2^n \). Hence, incorporating \( F^{(2)} \) into XKK, the security bound of the TBC becomes roughly \( \ell^2 q/2^n + \ell q^2/2^k + q^2/2^n \). XKK with \( F^{(2)} \) is denoted by XKK\(^{(2)}\).

### 1.4.3 Applications

Incorporating XKK into AE schemes \( \ThetaCB3 \) and \( \ThetaTR \), the resultant schemes are efficient ones, since for a query to the AE scheme, after a nonce-dependent key \( w \) is defined, it can be reused. In order to generate the nonce-dependent key, for each query to the AE scheme, the XKK\(^{(1)}\)-based AE schemes call a blockcipher once \( (k = n) \); twice \( (n < k \leq 2n) \), and the XKK\(^{(2)}\)-based AE schemes call it twice \( (k = n) \); three times \( (n < k \leq 2n) \). In addition to the blockcipher calls, these schemes perform a key scheduling once. For an input to the AXU hash function, the XKK-based schemes take a counter, whereas AE schemes with other BBB-secure TBCs take a counter and a nonce. In efficient AXU hash functions [Rog04, KR11, RBBK01, CS08, GJMN16], if a counter and a nonce are inputted, a nonce is inputted to a blockcipher, then the output \( L \) is updated by counters, e.g., the powering-up scheme [Rog04] updates \( L \) as \( 2 \cdot L, 2^2 \cdot L, 2^3 \cdot L \) etc., where the counters are
Table 1: Comparison of security and efficiency of BBB-secure TBCs. This table considers ΘCB3-based and OTr-based schemes whose underlying TBCs are given in the left most column. In these bounds, it is assumed that the influences of the decryption queries and associated data are sufficiently small. “BC” shows the number of blockcipher calls per one data block. In “Rekey,” “1/TBC” means that a key scheduling is performed for each TBC call, and “1/AE” means that a key scheduling is performed for each AE query. “Hash” shows the number of keyed hash function evaluations in one TBC call, and in the parentheses, the inputs are given. In “Precomp.,” i-E means that a blockcipher is called i times in a precomputation phase.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min [Min09]</td>
<td>$(\ell q)^2/2^n$</td>
<td>2</td>
<td>1/TBC</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>CLRW2 [LST12]</td>
<td>$(\ell q)^3/2^{2n}$</td>
<td>2</td>
<td>—</td>
<td>2 (N, ctr)</td>
<td>—</td>
</tr>
<tr>
<td>r-CLRW [LS13]</td>
<td>$(\ell q)^{r+2}/2^{2n}$</td>
<td>r</td>
<td>—</td>
<td>r (N, ctr)</td>
<td>—</td>
</tr>
<tr>
<td>XKK (1) [Ours]</td>
<td>$(\ell^2 q + q^2)/2^n + \ell q^2/2^k$</td>
<td>1</td>
<td>1/AE</td>
<td>1 (ctr)</td>
<td>1-E or 2-E</td>
</tr>
<tr>
<td>XKK (2) [Ours]</td>
<td>$\ell^2 q/2^n + \ell q^2/2^k$</td>
<td>1</td>
<td>1/AE</td>
<td>1 (ctr)</td>
<td>2-E or 3-E</td>
</tr>
</tbody>
</table>

1, 2, 3 etc., and the multiplication is done in $GF(2^n)$. In the XKK-based schemes, a nonce is not inputted to the AXU hash function, thus the blockcipher call with the nonce can be removed.\(^5\)

From the security bounds of XKK, those of the XKK-based AE schemes are $(\ell^2 q/2^n + \ell q^2/2^k + q^2/2^n)$ (XKK (1)) and $(\ell^2 q/2^n + \ell q^2/2^k)$ (XKK (2)), where $\ell$ is the number of blockciphers calls in one AE evaluation and $q$ is the number of queries to the AE scheme. Thus, if $q \ll 2^{n/2}$ and $\ell \ll 2^{n/2}$, the XKK (1)-based AE schemes achieve BBB security, and if $\ell \ll 2^{n/2}$, the XKK (2)-based AE schemes achieve BBB security. The security and the efficiency of the XKK-based AE schemes and other TBC-based AE schemes are summarized in Table 1, which are based on TBC-based AE schemes ΘCB3 and OTr. Note that since ΘCB3 and OTr are one-pass, online and parallelizable, so are the XKK-based schemes.

\(^{1.4.4}\) Impact in the Practical Setting

Finally, we study the security bounds of the XKK-based AE schemes. We consider the example study in [BL16] which is the HTTP connection where an adversary can make 2900 queries of length 4 Kbyte per second. We use a blockcipher with $n = 64$ (e.g., PRESENT [BKL07] and many other lightweight blockciphers). In this setting, the birthday bound is roughly $2^{35}$ blockcipher calls, and after one hour, the number of blockcipher calls reaches the bound. Next, XKK-based schemes are considered. For the sake of simplicity, we assume that the term $\ell q^2/2^k$ is negligible compared with other terms which can be achieved by using a blockcipher with long-size keys (e.g., $k = 128$). In this setting, the number of blockcipher calls $\ell$ is roughly $2^{9}$. Then the term $\ell^2 q/2^n$ becomes $q/2^{40} = (2^n)^2 q/2^{64}$, thus the term reaches 1/2 if $q = 2^{45}$. The $2^{45}$ AE queries spend $2^{45}/2900$ seconds $\approx 3847$ years. The term $q^2/2^n$ becomes $q^2/2^{64}$, thus the term reaches 1/2 if $q = 2^{31.5}$. The $2^{31.5}$ AE queries spend $2^{31.5}/2900$ seconds $\approx 121$ days. Hence, the security bounds of the AE schemes with XKK (1) (resp., XKK (2)) reach 1/2 after 121 days (resp., 3847 years), and in this setting, it seems hard to break the security of the XKK (2)-based schemes. Note that the birthday term $q^2/2^n$ comes from the PRF/PRP switch for the tweak function $F$. If the underlying blockcipher is not influenced by the PRF/PRP difference, that is, the birthday term can be ignored, then XKK (1)-based schemes have the

\(^5\) In the XKK-based schemes, $L$ can be randomly generated or can be generated by using a blockcipher with a constant input, e.g., $L = E_{K_e}(0^n)$. (In this case, $L$ can be precomputed.)
6 TBCs for Efficient AEs with BBB security

![Diagram of TBCs](image)

**Figure 3:** $\tilde{F}[1]$ (left), $\tilde{F}[2]$ (center), Wang et al.’s TBC $\tilde{E}4$ (right). $m$ is a plaintext block, $c$ is a ciphertext block, $tw$ is a tweak, and $K_E$ is a key. $\otimes$ is the multiplication in $GF(2^n)$.

same level of security as XKK(2)-based ones.

1.5 Related Works

Mennink [Men15] proposed two blockcipher-based TBCs called $\tilde{F}[1]$ and $\tilde{F}[2]$. $\tilde{F}[1]$ calls a blockcipher once and a multiplication once, and is a secure SPRP up to $2^{2n/3}$ queries. $\tilde{F}[2]$ calls a blockcipher twice, and is fully secure (secure up to $2^n$ queries). Note that the security proofs were given in the ideal-cipher model. Wang et al. [WGZ+16] extended the result of Mennink, and proposed 32 fully secure TBCs in the ideal-cipher model. Mennink’s TBCs and one of Wang et al.’s TBCs $\tilde{E}4$ are illustrated in Figure 3. These TBCs offer AE schemes with BBB security in the ideal-cipher model. Note that the security of our TBCs is given in the standard model (the SPRP assumption).

So far, several TBC-based AE schemes have been proposed. Minematsu [Min09] and Coron et al. [CDMS10] proposed 2n-bit blockcipher constructions from a TBC with n-bit block that is a fully secure SPRP. Combining these with birthday-bound AE schemes, the resultant schemes become fully secure AE schemes. Peyrin and Seurin [PS16] proposed an AE scheme that is fully secure against nonce-respecting adversaries and is birthday-bound secure against nonce-misuse adversaries. The AE scheme calls a blockcipher twice, and is not online. List and Nandi [LN17] proposed a fully secure deterministic AE scheme. The AE scheme calls a blockcipher twice and is not online. Again, these AE schemes are TBC-based.

Forler et al. [FLLW16] proposed a BBB-secure deterministic AE scheme that requires a 2n-bit blockcipher, an AXU-hash function, and an encryption scheme accepting variable length plaintexts.

Cogliati and Seurin [CLS15, CS15] proposed tweakable-Even-Mansour-type TBCs with BBB security. These schemes are permutation-based, and the security proofs were given in the random-permutation model. Many permutation-based AE schemes including CAESAR candidates [DEMS, BDP+a, BDP+b, AJN] have the sponge-style structures [BDPA08, BDPA11, JLM14, ADMA15, MRV15], where a permutation is iterated, and data blocks are handled by using several bits of the internal state. The security proofs were given in the random-permutation model, and the security bound is the birthday one.

1.6 Organization

We start by giving notations and security definitions in Section 2. In Section 3, we give the specification of XKK, the security bound, and the security proof. In Section 4, we give

---

6 Wang et al. [WGZ+16] showed that the primary version of $\tilde{F}[2]$ is not a fully secure TBC. After that, Mennink repaired the TBC to become a fully secure one.
blockcipher-based instantiations of $F$. In Section 5, we apply XKK to efficient TBC-based AE schemes $\Theta CB3$ and $\Theta TR$, and give the security bounds of the resultant AE schemes. Finally, in Section 6, we study the security bounds of the AE schemes.

## 2 Preliminaries

### 2.1 Notations

Let $\{0,1\}^*$ be the set of all bit strings, $\{0,1\}^n$ the set of $n$-bit strings, and $0^n$ the bit string of $n$-bit zeroes for an integer $n \geq 0$. Let $[i] := \{1,2,\ldots,i\}$ for a positive integer $i$. For a finite set $\mathcal{X}$, $x \in _{\text{r}} \mathcal{X}$ means that an element is randomly drawn from $\mathcal{X}$ and is assigned to $x$. For a bit string $x$ and a set $\mathcal{X}$, we denote by $|x|$ and $|\mathcal{X}|$ the bit length of $x$ and the number of elements in $\mathcal{X}$, respectively. Let $\text{trunc}_i(x)$ be the first $i$-bit string of a bit string $x$, where $i \leq |x|$. Let $\text{Perm}(B)$ be the set of all permutations over a non-empty set $B$. A random permutation over $B$ is defined as $P \xleftarrow{\$} \text{Perm}(B)$. The inverse is denoted by $P^{-1}$. An adversary $A$ with oracle access to $O$ is denoted by $A^O$. An event that $A^O$ outputs a result $y$ is denoted by $A^O \Rightarrow y$. In this paper, an adversary is a computationally bounded algorithm and the resource is measured in terms of time and query complexities.

### 2.2 Definitions of (Tweakable) Blockciphers

#### 2.2.1 Definition of Classical Blockcipher

Let $\mathcal{B}C(K,B)$ be the set of all encryptions of blockciphers with the set of keys $\mathcal{K}$ and the set of (plain/ciphertext) blocks $B$. Fixing a blockcipher $E \in \mathcal{B}C(K,B)$, $E$ having a key $K \in \mathcal{K}$, denoted by $E(K,\cdot)$ or $E_K(\cdot)$, becomes a permutation over $B$. The decryption function is denoted by $E^{-1}$, and $E^{-1}_K$ becomes the inverse permutation of $E_K$. An ideal cipher is defined as $E \xleftarrow{\$} \mathcal{B}C(K,B)$, and for each $K \in \mathcal{K}$, $E_K$ becomes a random permutation.

We consider Strong-Pseudo-Random Permutation (SPRP) security that is indistinguishability between a (keyed) blockcipher and a random permutation. Let $E \in \mathcal{B}C(K,B)$ be a blockcipher with the sets of keys $\mathcal{K}$ and blocks $B$. The advantage function of an sprp-adversary $A$ that outputs a bit are defined as

$$\text{Adv}_{E}^{\text{sprp}}(A) = \Pr[K \xleftarrow{\$} \mathcal{K}; A^{E_K,E_K^{-1} \Rightarrow 1} \rightarrow 1] - \Pr[P \xleftarrow{\$} \text{Perm}(B); A^{P,P^{-1} \Rightarrow 1} \rightarrow 1],$$

where the probabilities are taken over $A$, $K$ and $P$. We say $A$ is a $(q,t)$-sprp-adversary if $A$ makes $q$ queries and runs in time $t$. Pseudo-Random Permutation (PRP) security is a weaker security notion than SPRP security, where an adversary has access to only $E_K/P$. The advantage function of a prp-adversary $A$ is denoted by $\text{Adv}_{E}^{\text{prp}}(A)$. We say $A$ is a $(q,t)$-prp-adversary if $A$ makes $q$ queries and runs in time $t$.

#### 2.2.2 Definition of Tweakable Blockcipher

Let $\widetilde{\mathcal{B}}C(K,TW,B)$ be the set of all encryptions of tweakable blockciphers (TBCs) with the set of keys $\mathcal{K}$, the set of tweaks $TW$, and the set of (plain/ciphertext) blocks $B$. Fixing a TBC $\widetilde{E} \in \widetilde{\mathcal{B}}C(K,TW,B)$, $\widetilde{E}$ having a key $K \in \mathcal{K}$ and tweak $tw \in TW$, denoted by $\widetilde{E}(K,tw,\cdot)$ or $\widetilde{E}_K(tw,\cdot)$, becomes a permutation over $B$. The decryption function is denoted by $\widetilde{E}^{-1}$, and $\widetilde{E}_K^{-1}(tw,\cdot)$ is the inverse permutation of $\widetilde{E}_K(tw,\cdot)$.

We consider Tweakable-Strong-Pseudo-Random Permutation (TSPRP) security that is indistinguishability between a TBC and a (keyed) tweakable random permutation. Let $\widetilde{\text{Perm}}(TW,B)$ be the set of all tweakable permutations with the sets of tweaks $TW$ and of blocks $B$, where fixing $P \in \widetilde{\text{Perm}}(TW,B)$, $\widetilde{P}$ having a tweak $tw \in TW$ denoted by $\widetilde{P}(tw,\cdot)$.
becomes a permutation over $\mathcal{B}$. The inverse is denoted by $\tilde{P}^{-1}$. The advantage function of a tprp-adversary $A$ that outputs a bit is defined as

$$\text{Adv}_{E}^{\text{tprp}}(A) = \Pr\left[K \leftarrow \mathcal{K}; A^{\tilde{E}_{\mathcal{K}} \tilde{E}_{\mathcal{K}}^{-1}} \Rightarrow 1\right] - \Pr\left[\tilde{P} \leftarrow \tilde{\text{Perm}}(T \mathcal{W}, \mathcal{B}); A^{T} \tilde{P}^{-1} \Rightarrow 1\right],$$

where the probabilities are taken over $A$, $K$ and $\tilde{P}$. We say $A$ is a $(q, t)$-tprp-adversary if $A$ makes at most $q$ queries and runs in time $t$. Tweakable-Pseudo-Random-Permutation (TPRP) security is a weaker security notion than TSPRP security, where an adversary has access to only $\tilde{E}_{\mathcal{K}}/\tilde{P}$. The advantage function of a tprp-adversary $A$ is denoted by $\text{Adv}_{E}^{\text{tprp}}(A)$. We say $A$ is a $(q, t)$-tprp-adversary if $A$ makes at most $q$ queries and runs in time $t$.

2.3 Definition of Pseudo-Random Function

Let $\text{Func}(\mathcal{X}, \mathcal{Y})$ be the set of all functions from a set $\mathcal{X}$ to a set $\mathcal{Y}$. Let $\{F_{K}\}_{K \in \mathcal{K}}$ be a family of keyed functions indexed by the set of keys $\mathcal{K}$ that maps $\mathcal{X}$ to $\mathcal{Y}$. We consider Pseudo-Random-Function (PRF) security that is indistinguishability from a random function (RF), where an RF is defined as $f \leftarrow \tilde{\text{Func}}(\mathcal{X}, \mathcal{Y})$. The advantage function of a prf-adversary $A$ that outputs a bit is defined as

$$\text{Adv}_{f}^{\text{prf}}(A) = \Pr[K \leftarrow \mathcal{K}; A^{F_{K}} \Rightarrow 1] - \Pr[f \leftarrow \tilde{\text{Func}}(\mathcal{X}, \mathcal{Y}); A^{f} \Rightarrow 1],$$

where the probabilities are taken over $A$, $K$ and $f$. We say $A$ is a $(q, t)$-prf-adversary if $A$ makes at most $q$ queries and runs in time $t$.

2.4 Definition of Nonce-Based Authenticated Encryption

In this paper, we apply our TBC to nonce-based Authenticated Encryption (nAE) schemes (with associated data). The syntax and the definition of nAE schemes are given in the following.

An nAE scheme is a pair of encryption and decryption algorithms $\Pi = (\text{Enc}, \text{Dec})$. $\mathcal{K}, \mathcal{N}, \mathcal{M}, \mathcal{C}, \mathcal{A}$ and $\mathcal{T}$ are the sets of keys, nonces, messages, ciphertexts, associated data and tags of the nAE scheme. The encryption algorithm with a key $K \in \mathcal{K}$, $\text{Enc}_{K}$, takes a nonce $N \in \mathcal{N}$, associated data $A \in \mathcal{A}$, and a plaintext $M \in \mathcal{M}$. $\text{Enc}_{K}(N, A, M)$ returns, deterministically, a pair of a ciphertext $C \in \mathcal{C}$ and a tag $\text{tag} \in \mathcal{T}$. The decryption algorithm with a key $K \in \mathcal{K}$, $\text{Dec}_{K}$, takes a tuple $(N, A, \text{tag}) \in \mathcal{N} \times \mathcal{A} \times \mathcal{T}$. $\text{Dec}_{K}(N, A, \text{tag})$ returns, deterministically, either the distinguished invalid symbol $\bot$ or a plaintext $M \in \mathcal{M}$. We require $|\text{Enc}_{K}(N, A, M)| = |\text{Enc}_{K}(N, A, M')|$ when the encryption are strings and $|M| = |M'|$.

We follow the security definition in [BN08, Rog02] that considers privacy and authenticity of an nAE scheme $\Pi$. The privacy advantage of an adversary $A$ is defined as

$$\text{Adv}_{\Pi}^{\text{priv}}(A) = \Pr[K \leftarrow \mathcal{K}; \text{A}^{\text{Enc}_{K}} \Rightarrow 1] - \Pr[\text{A}^{\bot} \Rightarrow 1],$$

where a random-bits oracle $\cdot$ has the same interface as $\text{Enc}_{K}$, and for query $(N, A, M)$ returns a random bit string of length $|\text{Enc}_{K}(N, A, M)|$. The authenticity advantage of an adversary $A$ is defined as

$$\text{Adv}_{\Pi}^{\text{auth}}(A) = \Pr[K \leftarrow \mathcal{K}; A^{\text{Enc}_{K}, \text{Dec}_{K} \text{ forces}}]$$

where “$A^{\text{Enc}_{K}, \text{Dec}_{K} \text{ forces}}$” means that $A$ makes a query to $\text{Dec}_{K}$ whose response is not $\bot$. We demand that $A$ is nonce-respecting, namely, never asks two encryption queries with the same nonce, that $A$ never asks a decryption query $(N, A, C, \text{tag})$ such that there is no prior encryption query with $(C, \text{tag}) = \text{Enc}_{K}(N, A, M)$, and that $A$ never repeats a query.
Yusuke Naito

Algorithm 1 XKK

Procedure XKK[F,E]_{K_F,K_h}((N,ctr),m)
1: \( v \leftarrow h_{K_h}(ctr); \ x \leftarrow v \oplus m \)
2: \( w \leftarrow F_{K_F}(N); \ y \leftarrow E_w(x) \) \hfill \triangleright \text{Minematsu's TBC}
3: \( c \leftarrow y \oplus v \)
4: \text{return } c

2.5 Definition of Almost XOR Universal Hash Function

We will need a class of non-cryptographic functions called universal hash functions \([CW79]\) defined as follows.

Definition 1. Let \( \mathcal{H} = \{h_K\}_{K \in \mathcal{K}} \) be a family of functions from some set \( \mathcal{T W}_N \) to \( \{0,1\}^n \) indexed by the set of keys \( \mathcal{K} \). \( \mathcal{H} \) is said to be \((\epsilon,\delta)\)-almost XOR universal \((\epsilon,\delta)\)-AXU) if

1. for any distinct \( N,N' \in \mathcal{T W}_N \) and any \( c \in \{0,1\}^n \),
   \[
   \Pr[K \leftarrow \mathcal{K} : h_K(N) \oplus h_K(N') = c] \leq \epsilon ,
   \]
2. for any \( N \in \mathcal{T W}_N \) and any \( c \in \{0,1\}^n \),
   \[
   \Pr[K \leftarrow \mathcal{K} : h_K(N) = c] \leq \delta .
   \]

3 XKK

3.1 Specification

XKK is constructed from a blockcipher and two tweak functions. Using XKK in nAE schemes, one of the tweak function is used to take nonces, and the other is used to take counters. These definitions are given in the following.

- The blockcipher is defined as \( E \in BC(\{0,1\}^k,\{0,1\}^n) \), where positive integers \( n \) and \( k \) are the block size and the key size, respectively.

- The first tweak function is a keyed function from a set of (first) tweaks \( \mathcal{T W}_N \) to \( \{0,1\}^k \) whose family is defined as \( \mathcal{F} := \{F_{K_F}\}_{K_F \in \mathcal{K}_F} \) indexed by a set of keys \( \mathcal{K}_F \).

- The second tweak function is a keyed hash function from a set of (second) tweaks \( \mathcal{T W}_{ctr} \) to \( \{0,1\}^n \) whose family is defined as \( \mathcal{H} := \{h_{K_h}\}_{K_h \in \mathcal{K}_h} \) indexed by a set of keys \( \mathcal{K}_h \).

XKK[F,E]_{K_F,K_h} denotes XKK using underlying primitives \( F,E,h \) and keys \( K_F,K_h \).

XKK \( \in BC(\mathcal{K}_F \times \mathcal{K}_h, \mathcal{T W}_N \times \mathcal{T W}_{ctr}, \{0,1\}^n) \) is defined in Algorithm 1 and is illustrated in Figure 2. A tweak taken by the first (resp., second) tweak function is called a first (resp., second) tweak.

3.2 Security of XKK

The upper-bound of the tsprp-advantage for XKK given in the following theorem, assuming the keyed blockcipher is a secure SPRP, the first tweak function is a secure PRF and the family of second tweak functions is AXU.
Theorem 1. Assume that $\mathcal{H}$ is $(\epsilon, \delta)$-AXU. Let $A$ be a $(\sigma, t)$-tsprp-adversary. Here, $q$ is the number of distinct first tweaks, and $\ell_N$ is the number of queries with first tweak $N \in TW_N$. Then, there exist a $(\sigma, t + O(\sigma))$-sprp-adversary $A_E$ and $(q, t + O(\sigma))$-prf-adversary $A_F$ such that

$$\Adv_{\text{sprp}}^{\text{XKX}}(A) \leq q \cdot \Adv_{\text{sprp}}^{\text{E}}(A_E) + \Adv_{\text{sprp}}^{\text{prf}}(A_F) + \sum_{N \in N} \ell_N^2 \cdot \epsilon .$$

3.3 Proof of Theorem 1

Without loss of generality, assume that an adversary $A$ never repeats a query.

3.3.1 Replacing Minematsu’s TBC with TPRP

XKK is based on Minematsu’s TBC [Min09] whose encryption denoted by $\text{Min} \in \mathcal{BC}(K_F, TW_N, \{0,1\}^n)$ is defined as follows:

$$\text{Min}[F,E]_{K_F}(N,m) = E_w(m) \text{ where } w = F_{K_F}(N).$$

In Algorithm 1, Step 2 uses the TBC. Minematsu [Min09] gave the following upper-bound of the tsprp-advantage.

Lemma 1. Let $A$ be a $(\sigma, t)$-tsprp-adversary whose queries include $q$ distinct tweaks. Then there exist a $(\sigma, t + O(\sigma))$-sprp-adversary $A_E$ and a $(q, t + O(\sigma))$-prf-adversary $A_F$ such that

$$\Adv_{\text{sprp}}^{\text{XKX}}(A) \leq q \cdot \Adv_{\text{sprp}}^{\text{E}}(A_E) + \Adv_{\text{sprp}}^{\text{prf}}(A_F) .$$

The term $q \cdot \Adv_{\text{sprp}}^{\text{E}}(A_E)$ comes from the SPRP security of $E$ with $q$-blockcipher’s keys, and the term $\Adv_{\text{sprp}}^{\text{prf}}(A_F)$ comes from the PRF security of $F$.

By the above lemma, $\text{Min}$ can be replaced with a tweakable random permutation $\tilde{P}_R \overset{\$}{\leftarrow} \text{TPerm}(TW_N, \{0,1\}^n)$ with the above security loss. Hereafter, XKK using $\tilde{P}_R$ is denoted by $\tilde{F}[\tilde{P}_R]$.

3.3.2 TSPRP Security of $\tilde{F}[\tilde{P}_R]$

The remaining work is to upper-bound the tsprp-advantage which is defined as

$$\Adv_{\text{sprp}}^{\tilde{F}}(A) \overset{\$}{=} \Pr[\tilde{P}_R \overset{\$}{\leftarrow} \tilde{\text{TPerm}}(TW_N, \{0,1\}^n); K_h \overset{\$}{\leftarrow} K_h; A^{\tilde{F}[\tilde{P}_R|K_h, \tilde{F}[\tilde{P}_R|K_h^{-1} \Rightarrow 1]}; \Pr[\tilde{P}_I \overset{\$}{\leftarrow} \tilde{\text{TPerm}}(TW_N \times TW_{ctr}, \{0,1\}^n); A^{\tilde{P}_I, \tilde{P}_I^{-1} \Rightarrow 1}] .$$

This case can be seen as the multi-key setting of LRW2, where an adversary has oracle access to either $q$ LRW2 oracles with distinct random permutations or $q$ tweakable random permutation. Roughly speaking, since the random permutations of $q$ LRW2 oracles are independently defined, the above difference is upper-bounded by $q \times$ (the tsprp-advantage of LRW2 in the single-key setting). The upper-bound of the tsprp-advantage of LRW2 is given in [LRW02] (or [CLS15] for the more general case), which is $\ell^2 \cdot \epsilon$ for an adversary making at most $\ell$ queries. Hence, the above difference is upper-bounded by $\ell^2 q \cdot \epsilon$.

In the following, the full analysis is given. In this analysis, the values defined at the $\alpha$-th query are denoted by using the superscript character of $\alpha$. The world with $\tilde{F}[\tilde{P}_R|K_h$ is called the real world, and the world with $\tilde{P}_I$ is called the ideal world.

Transcript. This proof permits for $A$ to obtain the key $K_h$ after its interaction but before
outputting a decision bit. In the ideal world, a dummy key is defined as \( K_h \leftarrow K_{K_h} \). After \( A \)'s interaction, it obtains the following transcript.

\[
\tau = \left( K_h, \bigcup_{\alpha=1}^{\sigma} \{(N^\alpha, ctr^\alpha, m^\alpha, c^\alpha)\} \right)
\]

Let \( T_R \) be the transcript in the real world obtained by sampling \( \tilde{P}_R \leftarrow \text{Perm}(TW_N, \{0, 1\}^n) \) and \( K_h \leftarrow K_h \). Let \( T_I \) be the transcript in the ideal world obtained by sampling \( \tilde{P}_I \leftarrow \text{Perm}(TW_N \times TW_{ctr}, \{0, 1\}^n) \) and \( K_h \leftarrow K_h \). We call a transcript \( \tau \) valid if an interaction with their oracles could render this transcript, namely, \( \Pr[T_i = \tau] > 0 \) for \( i \in \{R,I\} \). Then the tspp-advantage is upper-bounded by the statistical distance of transcripts, i.e.,

\[
\text{Adv}_{\tilde{P}}^{\text{sprp}}(A) \leq \text{SD}(T_R, T_I) = \frac{1}{2} \sum_{\tau} |\Pr[T_R = \tau] - \Pr[T_I = \tau]| ,
\]

where the sum is over all valid transcripts.

**Coefficient H Technique.** The statistical distance \( \text{SD}(T_R, T_I) \) can be upper-bounded by the coefficient H technique [CS14, Pat08]. Let \( T \) be valid transcripts. In this technique, \( T \) is partitioned into two transcripts: good transcripts \( T_{\text{good}} \) and bad transcripts \( T_{\text{bad}} \). Then \( \text{SD}(T_R, T_I) \) is upper-bounded by the following lemma.

**Lemma 2.** Let \( 0 \leq \epsilon \leq 1 \) be such that for all \( \tau \in T_{\text{good}}, \frac{\Pr[T_R = \tau]}{\Pr[T_I = \tau]} \geq 1 - \epsilon \). Then, \( \text{SD}(T_R, T_I) \leq \Pr[T_I \in T_{\text{bad}}] + \epsilon \).

Hereafter, first good and bad transcripts are defined. Then \( \epsilon \) and \( \Pr[T_I \in T_{\text{bad}}] \) are upper-bounded. Finally, by the above lemma, the upper-bound of the tspp-advantage is obtained.

**Good and Bad Transcripts.** Bad transcripts \( T_{\text{bad}} \) are defined such that the following condition is satisfied, and good transcripts \( T_{\text{good}} \) are defined such that this condition is not satisfied.

- \( \text{coll} \iff \exists \alpha, \beta \in [\sigma] \) with \( \alpha \neq \beta \) s.t. \( N^\alpha = N^\beta \) and \( x^\alpha = x^\beta \) or \( y^\alpha = y^\beta \).

Note that in the ideal world, \( x^\alpha \) is defined as \( x^\alpha = h_{K_h}(ctr^\alpha) \oplus m^\alpha \), and \( y^\alpha \) is defined as \( y^\alpha = h_{K_h}(ctr^\alpha) \oplus c^\alpha \).

**Upper-Bound of \( \Pr[T_I \in T_{\text{bad}}] \).** Since \( \Pr[T_I \in T_{\text{bad}}] = \Pr[\text{coll}] \), in the following \( \Pr[\text{coll}] \) is upper-bounded.

Consider the condition in \( \text{coll} \): \( N^\alpha = N^\beta \) and \( x^\alpha = x^\beta \). The equation \( x^\alpha = x^\beta \) implies

\[
h_{K_h}(ctr^\alpha) \oplus h_{K_h}(ctr^\beta) = m^\alpha \oplus m^\beta.
\]

Hence, fixing \( \alpha, \beta \) with \( N^\alpha = N^\beta \), since \( H \) is \((\epsilon, \delta)\)-AXU, the probability that \( x^\alpha = x^\beta \) is at most \( \epsilon \). Similarly, fixing \( \alpha, \beta \) with \( N^\alpha = N^\beta \), the probability that \( y^\alpha = y^\beta \) is at most \( \epsilon \).

Since the number of queries with the first tweak \( N \) is \( \ell_N \), we have

\[
\Pr[T_I \in T_{\text{bad}}] = \Pr[\text{coll}] \leq \sum_{N \in TW_N} \left( \ell_N \right) \cdot 2\epsilon \leq \sum_{N \in TW_N} \ell_N^2 \cdot \epsilon .
\]
Upper-Bound of $\epsilon$. Let $\tau$ be a good transcript. Let all$_R$ (resp. all$_I$) be the set of all oracles in the real (resp. ideal) world. Let comp$_R(\tau)$ (resp. comp$_I(\tau)$) be the set of oracles compatible with $\tau$ in the real (resp. ideal) world. Then

$$\Pr[T_R = \tau] = \frac{|\text{comp}_R(\tau)|}{|\text{all}_R|} \quad \text{and} \quad \Pr[T_I = \tau] = \frac{|\text{comp}_I(\tau)|}{|\text{all}_I|}.$$  

Firstly, $|\text{all}_R|$ is counted. Since $K_h \in K_h$ and $\tilde{P}_R \in \tilde{\text{Perm}}(TW_N, \{0, 1\}^n)$, we have

$$|\text{all}_R| = |K_h| \cdot (2^n)^{|TW_N|}.$$  

Secondly, $|\text{all}_I|$ is counted. Since $K_h \in K_h$ and $\tilde{P}_I \in \tilde{\text{Perm}}(TW_N \times TW_{ctr}, \{0, 1\}^n)$, we have

$$|\text{all}_I| = |K_h| \cdot (2^n)^{|TW_N| \times |TW_{ctr}|}.$$  

Thirdly, $|\text{comp}_R(\tau)|$ is counted. By $\neg\text{coll}$, all $\tilde{P}_R$-evaluations with the same first tweak don’t overlap with each other. $K$ is uniquely determined. Hence, we have

$$|\text{comp}_R(\tau)| = \prod_{N \in TW_N} (2^n - \ell_N)!.$$  

Fourthly, $|\text{comp}_I(\tau)|$ is counted. Let $\ell_{N, ctr}$ be the number of queries with the first tweak $N$ and the second tweak $ctr$. Note that $\ell_N = \sum_{ctr \in TW_{ctr}} \ell_{N, ctr}$. Then,

$$|\text{comp}_I(\tau)| = \prod_{N \in TW_N, ctr \in TW_{ctr}} (2^n - \ell_{N, ctr})!$$

$$\leq (2^n)^{|TW_{ctr}|} \cdot \prod_{N \in TW_N} (2^n - \ell_N)!,$$

using $(2^n - a)! \cdot (2^n - b)! \leq 2^n! \cdot (2^n - a - b)!$ for any $0 \leq a, b \leq 2^n$.

Finally,

$$\frac{\Pr[T_R = \tau]}{\Pr[T_I = \tau]} \geq \prod_{N \in TW_N} \frac{(2^n - \ell_N)!}{|K_h| \cdot (2^n)^{|TW_N|}} \times \frac{|K_h| \cdot (2^n)^{|TW_N| \times |TW_{ctr}|}}{(2^n)^{|TW_{ctr}|} \cdot \prod_{N \in TW_N} (2^n - \ell_N)!} = 1.$$  

Thus we have $\epsilon = 0$.

Upper-Bound of $\text{Adv}_{\tilde{F}}^{\text{sprp}}(A)$. Putting the above upper-bounds in Lemma 2 gives

$$\text{Adv}_{\tilde{F}}^{\text{sprp}}(A) \leq \sum_{N \in TW_N} \ell_N^2 \cdot \epsilon.$$  

3.3.3 Conclusion of the Proof

Finally, combining Lemma 1 and the upper-bound of $\text{Adv}_{\tilde{F}}^{\text{sprp}}(A)$ gives

$$\text{Adv}_{\tilde{F}}^{\text{sprp}}(A) \leq q \cdot \text{Adv}_{\tilde{E}}^{\text{sprp}}(A_E) + \text{Adv}_{\tilde{F}}^{\text{sprp}}(A_F) + \sum_{N \in TW_N} \ell_N^2 \cdot \epsilon,$$

where $A_E$ is a $(\sigma, t + O(\sigma))$-sprp-adversary and $A_F$ is a $(q, t + O(\sigma))$-prf-adversary. \(\square\)
3.4 Removing the Output Masking of XKK

In Theorem 1, the upper-bound of the tsprrp-advantage of XKK is given, where an adversary has oracle access to both of the encryption function and the decryption function. In XKK, in order to avoid a collision attack in inputs to the underlying blockcipher from the encryption oracle, the input masking is introduced, and in order to avoid a collision in outputs to the underlying blockcipher from the decryption oracle, the output masking is introduced. Since the decryption oracle is absent in the TPRP-setting, the output masking of XKK can be removed, i.e., the resultant TBC that we call XKK is a secure TPRP. The tsprrp-advantage of XKK can be upper-bounded by the proof similar to Theorem 1, where the condition \( y^o = y^\beta \) in the event coll (in Subsubsection 3.3.2) is not required, and other analyses are the same. Concretely, the upper-bound of the tsprrp-advantage is given below. Assume that \( H \) is \((\epsilon, 0)\)-AXU. Let \( A \) be a \((\sigma, t)\)-tprp-adversary. Here, \( q \) is the number of distinct first tweaks, and \( t_N \) is the number of queries with first tweak \( N \in TW_N \). Then, there exist a \((\sigma, t + O(\sigma))\)-prp-adversary \( A_E \) and \((q, t + O(\sigma))\)-prf-adversary \( A_F \) such that

\[
\text{Adv}_{XK}^{\text{sp}}(A) \leq q \cdot \text{Adv}_{E}^{\text{sp}}(A_E) + \text{Adv}_{F}^{\text{sp}}(A_F) + \sum_{N \in N} 0.5q^2 \cdot \epsilon \, .
\]

4 Instantiations of \( F \)

We show how to construct the first tweak function \( F_{K_n} \) from a blockcipher. The blockcipher is defined as \( E \in BC(\{0, 1\}^k, \{0, 1\}^n) \). We deal with blockciphers with \( n \leq k \leq 2n \), since almost all of blockciphers satisfy the condition.

We define a first tweak function that uses blockcipher outputs.

- \( F_{K_F}^{(1)}(N) = \text{trunc}_k(E_{K_F}(0||N)||E_{K_F}(1||N)) \) where \( K_F = K_E, K_F := \{0, 1\}^k \), and \( TW_N := \{0, 1\}^{n-1} \).

By the PRF/PRP switch, the prf-advantage is upper-bounded the prp-advantage of the blockcipher plus the birthday bound \( O(q^2/2^n) \).

Next, we define a first tweak functions so that the birthday bound is removed. In order to remove the birthday bound, the xor function is used.

- \( F_{K_F}^{(2)}(N) = \text{trunc}_k \left( (E_{K_F}(00||N) \oplus E_{K_F}(01||N)) \oplus (E_{K_F}(00||N) \oplus E_{K_F}(10||N)) \right) \) where \( K_F = K_E, K_F := \{0, 1\}^k \), and \( TW_N := \{0, 1\}^{n-2} \).

In [Pat10], it was proven that the xor function achieves optimal PRF security, thus the prf-advantage is upper-bounded by the prp-advantage of the blockcipher plus the optimal PRF-security bound \( O(q/2^n) \).

These concrete bounds are given in the following, where the upper-bound of \( F^{(1)} \) is obtained by the PRF/PRP switch, and the upper-bound of \( F^{(2)} \) is obtained by using Theorem 2 in [IMV16] (the original analysis is given in Theorem 6 of [Pat10]).

**Lemma 3** (PRF Security of \( F^{(1)} \)). For any \((q, t)\)-prf-adversary \( A \), there exists a \((2q, t + O(q))\)-prp-adversary \( A_E \) such that

\[
\text{Adv}_{F^{(1)}}^{\text{prf}}(A) \leq \text{Adv}_{E}^{\text{sp}}(A_E) + \frac{q^2}{2^n} \, .
\]

**Lemma 4** (PRF Security of \( F^{(2)} \)). For any \((q, t)\)-prf-adversary \( A \), there exists a \((3q, t + O(q))\)-prp-adversary \( A_E \) such that

\[
\text{Adv}_{F^{(2)}}^{\text{prf}}(A) \leq \text{Adv}_{E}^{\text{sp}}(A_E) + \frac{4q}{2^n} \, .
\]
Remark 1. When $n = k$, $F^{(1)}$ (resp., $F^{(2)}$) calls a blockcipher once (resp., twice), and the domain separation bit(s) perpendent to $N$ can be removed (resp., shortened). When $2n < k$, the first tweak functions can be defined by making the bit length longer.

5 Applications

We apply XKK to nAE schemes ΘCB [KR11] and OTR [Min14]. As shown below, the XKK-based schemes achieve BBB security, and become efficient, one-pass, online and parallelizable.

5.1 ΘCB3 with XKK

5.1.1 ΘCB3 [KR11]

ΘCB3 is a TBC generalization of OCB3 [KR11], and is efficient, one-pass, online and parallelizable. In ΘCB3, a plaintext is encrypted by the ECB-like construction (but a tweak is varied for each block), and a tag is generated by encrypting the checksum of plaintext blocks. In the decryption, a ciphertext is decrypted by the decryption of the ECB-like construction, then a tag is generated by encrypting the checksum of the decrypted plaintext blocks (thus a tag is generated by the PMAC-like structure [Rog04]).

We briefly give the construction of ΘCB3, following the notations in [KR11]. Here, a TBC is defined as $\tilde{E} \in \overline{BC}((0, 1)^k, TW, \{0, 1\}^n)$, where $k$ is the key size in bits, $TW$ is the set of tweaks, and $n$ is the block size. The set of tweaks is defined as follows.

$$TW := (N \times N_1) \cup (N \times N_0 \times \{\ast\}) \cup (N \times N_0 \times \{\$\}) \cup (N \times N_0 \times \{\ast\})$$

$$\cup N_1 \cup (N_0 \times \{\ast\})$$

where $N$ is the set of nonces, $N_1$ and $N_0$ are positive and nonnegative integers, respectively. Hence, ΘCB3 uses six types of permutations: $\tilde{E}_K((N,i), \cdot)$, $\tilde{E}_K((N,i, \ast), \cdot)$, $\tilde{E}_K((N,i, \$), \cdot)$, $\tilde{E}_K((N,i, \ast\$), \cdot)$, $\tilde{E}_K(i, \cdot)$, and $\tilde{E}_K((i, \ast), \cdot)$. The first two permutations are used to encrypt plaintext blocks. The next two permutations are used to generate a tag. The last two permutations are used to handle associated data. In each procedure, the latter permutation is used to avoid an additional permutation call by the padding. In the encryption of ΘCB3, for a nonce $N$ and $n$-bit plaintext blocks $M_1, \ldots, M_i$, the $i$-th ciphertext block is defined as $C_i \leftarrow \tilde{E}_K((N,i), M_i)$. Regarding associated data $A_1, \ldots, A_l$, the $i$-th block $A_i$ is inputted to the TBC as $B_i \leftarrow \tilde{E}_K(i, A_i)$. Then the tag is defined as $tag \leftarrow \text{trunc}_c(\tilde{E}_K((N,i, \$), \text{Checksum}) \oplus B_1 \oplus \cdots \oplus B_n)$, where Checksum is the checksum of the plaintext blocks. Note that if the length of the message is not multiple of $n$, then permutations with tweaks including "$\$" are used to encrypt the last block and generate a tag. Similarly, if the length of associated data is not multiple of $n$, then $\tilde{E}_K((i, \ast), \cdot)$ is used to process the last block of associated data. The encryption of ΘCB3 is illustrated in Figure 4. In the decryption of ΘCB3, the inverse procedure of the encryption is performed. Please see Subsection 4.2 in [KR11] for the concrete construction of ΘCB3.

In [KR11], the security of ΘCB3 was analyzed in the information-theoretic model, that is, the keyed TBC is replaced with a tweakable random permutation. Regarding the privacy, for each TBC call, a distinct tweak is used, thus each ciphertext block is randomly drawn from $\{0, 1\}^n$. Hence, for any adversary $A$,

$$\text{Adv}_{\text{priv}}^{\text{TCB3}}(A) = 0.$$ 

Regarding the authenticity, two cases are considered: an adversary $A$ makes a decryption query such that (1) the nonce appeared in the previous encryption queries; (2) the nonce has not appeared in the previous encryption queries. In order to forge a tag, in (1), $A$
Figure 4: Encryption of \( \Theta CB3 \). The procedure (1) encrypts four \( n \)-bit plaintext blocks \( M_1, M_2, M_3, M_4 \), and returns a tag. The procedure (2) encrypts three \( n \)-bit plaintext blocks \( M_1, M_2, M_3 \) and a plaintext block of length less than \( n \) bits \( M^* \), and returns a tag. The procedure (3) handles three \( n \)-bit associated data blocks \( A_1, A_2, A_3 \). The procedure (4) handles two \( n \)-bit associated data blocks \( A_1, A_2 \) and an associated data block \( A_3 \) of length less than \( n \) bits \( (A_3 \| 10^* \) is an \( n \)-bit string, where 1 is appended to \( A_3 \) and an appropriate number of bits 0 is appended so that the bit length becomes \( n \)).

should occur a collision of the checksum values with the same nonces (yielding the same tags), and then makes a query with the same tag; in (2), \( A \) should hit a tag that is randomly drawn. In [KR11], it was proven that these probabilities are at most \( 2^{n-\tau}/(2^n-1) \). Hence, for any adversary \( A \) making at most \( q_D \) decryption queries,

\[
\text{Adv}_{\Theta CB3}^{\text{adv}}(A) \leq \frac{q_D 2^{n-\tau}}{2^n-1}.
\]

5.1.2 \( \Theta CB3 \) with \( XKX \)

We apply \( XKX \) to \( \Theta CB3 \). The resultant scheme is denoted by \( \Theta CB3[XKX] \). The set of first tweaks is defined as \( TW_N := N \cup \{0\} \) such that \( 0 \notin N \). “0” is used to define a blockcipher’s key to handle associated data. The set of second tweaks is defined as \( TW_{ctr} := N_1 \cup (N_0 \times \{\ast\}) \cup (N_0 \times \{$$\}) \cup (N_0 \times \{\ast$$\}) \cup N_1 \cup (N_0 \times \{\ast\}) \). In \( XKX \), for each encryption or decryption query, a blockcipher’s key is defined by the first tweak function \( F_{K_{\tau}} \) whose input is a nonce \( N \) and is fixed, thus \( F_{K_{\tau}} \) is called once for each encryption or decryption query. Namely, for each data block, \( \Theta CB3[XKX] \) calls a blockcipher once (and calls an AXU hash function once). Hence, \( \Theta CB3[XKX] \) is efficient, one-pass, online and parallelizable. Since inputs of the AXU hash function do not include nonces, the hash values can be precomputed. If there is a storage that keeps the hash values, then the hash...
computations can be removed.

Regarding the security of ΘCB3[XKX], since XKX can be used as a tweakable random permutation up to the security bound given in Theorem 1, the security bounds of ΘCB3[XKX] are obtained by summing the security bound given in Theorem 1 and the security bounds of ΘCB3. The details are given in following. Here, \( \mathcal{H} \) is assumed to be \((\epsilon, \delta)\)-AXU.

First, the privacy of ΘCB3[XKX] is considered. Let \( A \) be an adversary that makes \( q_E \) encryption queries and runs in time \( t \) such that \( \sigma_A \) is the number of blockcipher calls by associated data and \( \ell_N \) is the total number of blockcipher calls by queries with the nonce \( N \in \mathcal{N} \). Let \( \sigma_E = \sigma_A + \sum_{N \in \mathcal{N}} \ell_N \). In this setting, at most \( q_E \) “nonce-dependent” blockcipher’s keys are defined, which are used to encrypt plaintexts and generate tags, and another key is used to handle associated data. Hence, at most \( q_E + 1 \) blockcipher’s keys are defined. Combining Theorem 1 and the privacy bound of ΘCB3, the following result is obtained: There exist a \((\sigma_E, t + O(\sigma_E))\)-sprp-adversary \( A_E \) and a \((q_E + 1, t + O(\sigma_E))\)-prf-adversary \( A_F \) such that

\[
\text{Adv}_{\text{priv}_{\Theta CB3[XKX]}}^E(A) \leq (q_E + 1) \cdot \text{Adv}_{E}^{\text{sprp}}(A_E) + \text{Adv}_{F}^{\text{prf}}(A_F) + \left( \frac{\sigma_A^2}{2} + \sum_{N \in \mathcal{N}} \ell_N^2 \right) \cdot \epsilon .
\]

Next, the authenticity of ΘCB3[XKX] is considered. Let \( A \) be an adversary that makes \( q_E \) encryption/\( q_D \) decryption queries and runs in time \( t \) such that the number of blockcipher calls by associated data is \( \sigma_A \) and the number of blockcipher calls by queries with the nonce \( N \in \mathcal{N} \) is \( \ell_N \). Let \( q = q_E + q_D \) and \( \sigma = \sigma_E + \sum_{N \in \mathcal{N}} \ell_N \). In this setting, at most \( q \) nonce-dependent blockcipher’s keys are defined to encrypt plaintexts, decrypt ciphertexts and generate tags, and another key is defined to handle associated data. Hence, at most \( q + 1 \) blockcipher’s keys are defined. Combining Theorem 1 and the authenticity bound of ΘCB3, the following result is obtained: There exist a \((\sigma, t + O(\sigma))\)-sprp-adversary \( A_E \) and a \((q + 1, t + O(\sigma))\)-prf-adversary \( A_F \) such that

\[
\text{Adv}_{\text{auth}_{\Theta CB3[XKX]}}^E(A) \leq (q + 1) \cdot \text{Adv}_{E}^{\text{sprp}}(A_E) + \text{Adv}_{F}^{\text{prf}}(A_F) + \left( \frac{\sigma_A^2}{2} + \sum_{N \in \mathcal{N}} \ell_N^2 \right) \cdot \epsilon + \frac{qD2^{n-\tau}}{2^n - 1}.
\]

### 5.2 OTR with XKX

OTR [Min14] is a variant of ΘCB3, which is also efficient, one-pass, online and parallelizable (under two-block partition). OTR encrypts two plaintext blocks by two-round Feistel permutation, where in each round a TBC (or a function) is called once. By the Feistel permutation, OTR does not require the decryption function of the underlying TBC. Hence, adopting XKX to OTR, the resultant scheme is efficient, one-pass, and parallelizable without a decryption function of a blockcipher.

OTR has the same level of security as ΘCB (the constant factors in the authenticity bounds are distinct). Hence, OTR with XKX also achieves the same level of security as ΘCB[XKX]. Since OTR does not require a decryption function of a blockcipher, the security proofs of the OTR-based AE scheme do not require the SPRP assumption but require the PRP one.

### 5.3 Remark

Since OTR does not require the decryption function of a TBC, XKX can be replaced with XK. In ΘCB3, the decryption function is not required for handling associated data and generating a tag. Hence, for the TBC calls, XKX can be replaced with XK.
6 Discussions

6.1 Study of Security Bounds of XKX-based Schemes

We study the security bounds of \( \Theta_{CB3}[XKX] \) given in Subsection 5.1. Note that this study is applicable to OTR with XKX. For the sake of simplicity, we assume that for each encryption query, the number of blockcipher calls except for those handling associated data is \( \ell \). We use an optimal parameter \( \epsilon = 1/2^n \).

6.1.1 Security Bounds of \( \Theta_{CB3}[XKX] \)

The privacy bound becomes roughly
\[
q_\ell \cdot \text{Adv}_{E}^{\text{sprp}}(A_E) + \text{Adv}_{F}^{\text{prf}}(A_F) + \frac{\sigma_{A}^{2} + \ell^{2}q_{\ell}}{2^n} \quad \text{(privacy)}.
\]

Regarding the authenticity bound, \( \sum_{N \in \mathbb{N}} \ell^2 N \cdot \epsilon \) becomes maximum if an adversary makes decryption queries whose nonces are the same and appear in some encryption query, thus this term is at most \( \left( \ell^2(q - 1) + (\ell + \sigma_{D})^2 \right)/2^n \). Hence, the authenticity bound becomes roughly
\[
q \cdot \text{Adv}_{E}^{\text{sprp}}(A_E) + \text{Adv}_{F}^{\text{prf}}(A_F) + \sigma_{A}^{2} + \ell^2q_{\ell} + \ell^2q_{\ell} \quad \text{(authenticity)}.
\]

Next, we assume that \( \text{Adv}_{E}^{\text{sprp}}(A_E) \approx \ell q_{E}/2^k \) or \( \text{Adv}_{F}^{\text{sprp}}(A_E) \approx \ell q_{E}/2^k \) (achieved by the exhaustive key search, see [BKR98]), and terms with \( \sigma_{A} \) and \( \sigma_{D} \) are sufficiently small (e.g., associated data is fixed and the number of fails in decryption queries is limited). Then the privacy and authenticity bounds become roughly
\[
\frac{\ell q_{E}^2}{2^k} + \text{Adv}_{E}^{\text{prf}}(A_E) + \frac{\ell^2q_{\ell}}{2^n} \quad \text{(privacy)} \quad \frac{\ell q_{E}^2}{2^k} + \text{Adv}_{F}^{\text{prf}}(A_F) + \frac{\ell^2q_{\ell}}{2^n} \quad \text{(authenticity)}.
\]

These bounds ensure that if the key terms \( \ell q_{E}/2^k \) and \( \text{Adv}_{F}^{\text{prf}}(A_F) \) can be negligible, then the security bounds of the XKX-based schemes become \( O(\ell^2q_{E}/2^n) \).

6.1.2 Blockcipher-based Instantiations

We adopt instantiations given in Section 4 to XKX. Here, XKX using function \( F^{(i)} \) is denoted by \( \text{XKX}^{(i)} \). By Lemma 3 and 4, the privacy and authenticity bounds become roughly

- \( \Theta_{CB3}[\text{XKX}^{(1)}] \):
  \[
  \frac{\ell q_{E}^2}{2^k} + \frac{q_{E}^2}{2^n} + \frac{\ell^2q_{\ell}}{2^n} \quad \text{(privacy)} \quad \frac{\ell q_{E}^2}{2^k} + \frac{q_{E}^2}{2^n} + \frac{\ell^2q_{\ell}}{2^n} \quad \text{(authenticity)} \quad (1)
  \]

- \( \Theta_{CB3}[\text{XKX}^{(2)}] \):
  \[
  \frac{\ell q_{E}^2}{2^k} + \frac{\ell^2q_{\ell}}{2^n} \quad \text{(privacy)} \quad \frac{\ell q_{E}^2}{2^k} + \frac{\ell^2q_{\ell}}{2^n} \quad \text{(authenticity)} \quad (2)
  \]

Hence, these bounds are beyond the birthday ones (The birthday bound is \( (\ell q^2)/2^n \)). The term \( \ell^2q_{E}/2^k \) depends on the key size \( k \), thus using a blockcipher with long-size keys, these terms can be negligible, e.g., \( k = 2n \). AES and many lightweight blockciphers support this parameter (AES: \( n = 128 \) and \( k = 256 \), PRESENT: \( n = 64 \) and \( k = 128 \), etc.). If this term can be negligible, then the above bounds become \( O(q^2/2^n + \ell^2q_{E}/2^n) \) (\( \Theta_{CB3}[\text{XKX}^{(1)}] \)) and \( O(\ell^2q_{E}/2^n) \) (\( \Theta_{CB3}[\text{XKX}^{(2)}] \)).
6.2 Study of the Key Terms $q \cdot \text{Adv}^\text{sprp}_E(A_E)$ and $\text{Adv}^\text{prf}_E(A_F)$

The security bounds of XKK-based schemes have the term $q \cdot \text{Adv}^\text{sprp}_E(A_E)$. This term comes from the construction of XKK, where $q$-blockcipher’s keys are defined via the first tweak function. As mentioned in [ST16], the hybrid factor is not real, but rather an artifact of the proof technique, and this term might be improved by analyzing the term in the ideal cipher model (ICM). On the other hand, the ICM analysis provides only a security heuristic, and seems particularly inappropriate when the underlying blockcipher is known to have obvious non-ideal behavior for certain weak-key class matters, or to suffer from related-key attack. However, in the XKK-based schemes the blockcipher’s keys are randomly drawn, thereby the presence of weak keys is unlikely to be a real issue in the XKK-based schemes. In this paper, we study the security of XKK-based schemes in the ICM, and show that this term can be improved.

Recall the proof of Theorem 1. The terms $q \cdot \text{Adv}^\text{sprp}_E(A_E)$ and $\text{Adv}^\text{prf}_E(A_F)$ are introduced in Subsubsection 3.3.1, where Min is replaced with a tweakable random permutation. The following lemma analyzes this replacement in the ICM, where the underlying blockcipher is replaced with an ideal cipher, and the first tweak function is replaced with a random function. Here, direct queries to the ideal cipher are called online queries, and queries to XKK are called online queries. XKK using an ideal cipher $E$, a random function $f$ and a key of the second tweak function $K_h$ is denoted by XKK[$f, E|K_h$].

**Lemma 5.** Assume that $H$ is $(\epsilon, \delta)$-AXU. Let $A$ be a computationally unbounded adversary trying to distinguish World1 from World2. Here, $A$ makes $\sigma$ online queries with $q$ first tweaks and $Q$ offline queries. Then we have

$$\Pr[\text{World1}] - \Pr[\text{World2}] \leq \frac{2\sigma Q\delta}{2^{k-\tau}} + \frac{q^2}{2^{k+\tau}},$$

where

**World1** :=

$$\left( E \overset{\$}{\leftarrow} \text{BC}([0,1]^k, \{0,1\}^n); f \overset{\$}{\leftarrow} \text{Func}(\mathcal{T}W_N, \{0,1\}^k); K_h \overset{\$}{\leftarrow} K_h; A^{\text{XKK}[f, E|K_h], E^\pm} \Rightarrow 1 \right)$$

**World2** :=

$$\left( E \overset{\$}{\leftarrow} \text{BC}([0,1]^k, \{0,1\}^n); \tilde{R} \overset{\$}{\leftarrow} \text{Perm}(\mathcal{T}W_N, \{0,1\}^n); K_h \overset{\$}{\leftarrow} K_h; A^{\tilde{R}[\tilde{R}|K_h], E^\pm} \Rightarrow 1 \right).$$

**Proof sketch.** In this part, a sketch of the security proof is given, and the full proof is given in Subsection 6.3.

From World1 to World2, Min is replaced with a tweakable random permutation. Hence, we need to evaluate the distinguishing probability coming from the structural differences. There are two differences given in the following.

- **Difference 1:** The difference comes from the (in)dependence between online and offline queries. In World1, responses of online and offline queries are defined by using an ideal cipher. On the other hand, in World2, responses of online queries are defined independently of an ideal cipher.

- **Difference 2:** The difference comes from the (in)dependence between online queries with distinct first tweaks. In World1, for two online queries with distinct first tweaks, if the blockcipher’s keys are the same (that is, a collision occurs in outputs of $f$), the responses are defined by the same permutation (or a blockcipher with the same key). On the other hand, in World2, for two online queries with distinct first tweaks, the responses are defined by the distinct permutations.
Regarding Difference 1, in World1, the independence between online and offline queries might not be ensured, if blockcipher’s inputs (or outputs) by online queries and by offline ones overlap with each other. This means that one of pairs for \((w, x)\) (or \((w, y)\)) appears in offline queries, where \(x\) is the input block of the blockcipher, \(w\) is the blockcipher’s key, and \(y\) is the output of the blockcipher, defined in XKK. Since \(w\) is randomly drawn from \(\{0, 1\}^n\) by a random function \(f\), and \(x\) and \(y\) are defined by the AXU hash function, the collision (or overlapping) probability is at most \(2\sigma Q \cdot \delta / 2^k\). The factor \(\cdot 2^n\) comes from collisions for input blocks and for output blocks.

Regarding Difference 2, in World1, the independence between online queries with distinct first tweaks might not be ensured if a collision occurs in outputs of \(f\). Since there are at most \(q\) inputs to \(f\), the collision probability is, by the birthday analysis, at most \(q^2 / 2^{k+1}\).

By summing these probabilities, the upper-bound of the lemma is obtained.

Using the optimal parameters \(\epsilon = \delta = 1/2^n\) and assuming \(Q \approx \ell q\), the above bound becomes roughly \((\ell q)^2 / 2^{n+k} + q^2 / 2^k\). This bound offers the upper-bounds of the ICM-security of \(\ThetaCB3[\text{XKK}(1)]\) and \(\ThetaCB3[\text{XKK}(2)]\) given in the following.

- \(\ThetaCB3[\text{XKK}(1)]\):
  \[
  \frac{q^2}{2^n} + \frac{\ell^2 q^2}{2^n} \quad \text{(privacy)} \quad \frac{q^2}{2^n} + \frac{\ell^2 q}{2^n} \quad \text{(authenticity)}
  \]
- \(\ThetaCB3[\text{XKK}(2)]\):
  \[
  \frac{q^2}{2^k} + \frac{\ell^2 q^2}{2^n} \quad \text{(privacy)} \quad \frac{q^2}{2^k} + \frac{\ell^2 q}{2^n} \quad \text{(authenticity)}
  \]

In the security bounds of XKK-based schemes, the term in the standard model, \(\ell q^2 / 2^k\), is improved to \(q^2 / 2^n\) in the ICM. Note that \((\ell q)^2 / 2^{n+k}\) can be negligible compared with \(\ell^2 q^2 / 2^n\). Hence, query length \(\ell\) becomes independent from the key size in the ICM. Finally, the term \(q^2 / 2^n\) comes from the PRF/PRP switch for the first tweak function \(F\). If the underlying blockcipher is not influenced by the switch, that is, the term \(q^2 / 2^n\) can be eliminated, then the XKK(1)-based schemes achieve the same level of security as the XKK(2)-based ones. Indeed, from World1 to World2, the term \(q^2 / 2^n\) is introduced by Difference 2 that considers a collision in outputs of \(f\). Using \(F(1)\) instead of \(f\), since such collision does not occur due to the use of blockcipher’s outputs, one does not have to consider Difference 2 and the term \(q^2 / 2^n\) is eliminated.

### 6.3 Proof of Lemma 5

In this proof, the upper-bound of the difference \(\Pr[\text{World1}] - \Pr[\text{World2}]\) is given, where

\[
\begin{align*}
\text{World1} := & \left( E \xleftarrow{} \mathcal{B}C(\{0, 1\}^k, \{0, 1\}^n); f \xleftarrow{} \text{Func}(TW_N, \{0, 1\}^k); \\
& K_h \xleftarrow{} \mathcal{K}_h; A^{XKK[f,k]_h,E\pm}_h \Rightarrow 1 \right)
\end{align*}
\]

\[
\begin{align*}
\text{World2} := & \left( E \xleftarrow{} \mathcal{B}C(\{0, 1\}^k, \{0, 1\}^n); \tilde{P}_R \xleftarrow{} \text{Perm}(TW_N, \{0, 1\}^n); \\
& f \xleftarrow{} \text{Func}(TW_N, \{0, 1\}^k); K_h \xleftarrow{} \mathcal{K}_h; A^{\tilde{P}_R|_h,E\pm}_h \Rightarrow 1 \right).
\end{align*}
\]

Note that in World2, a random function \(f\) (on the underlined statement) is introduced. It is used in this proof but is not used to define responses of \(A\)'s queries. Thus this modification does not change \(A\)'s behavior. In this proof, the query-response triple at
the \(\alpha\)-th online query are denoted by \((tw^\alpha, m^\alpha, c^\alpha)\), and the corresponding values such as \(v, w, x, y\) are denoted by using the superscript character of \(\alpha\). The query-response triple at the \(\beta\)-th offline query are denoted by \((W^\beta, X^\beta, Y^\beta)\), where \(Y^\beta = E(W^\beta, X^\beta)\) if the \(\beta\)-th offline query is a query to \(E\) and \(X^\beta = E^{-1}(W^\beta, Y^\beta)\) if the \(\beta\)-th offline query is a query to \(E^{-1}\).

**Transcript.** After \(A\)'s interaction, it obtains the following list:

\[
\left( \bigcup_{\alpha=1}^{q} \{(tw^\alpha, m^\alpha, c^\alpha)\}, \bigcup_{\beta=1}^{Q} \{(W^\beta, X^\beta, Y^\beta)\} \right).
\]

The list is referred as transcript. Let \(T_1\) be the transcript in \(\text{World}1\) obtained by sampling \(E \xleftarrow{\$} \text{BC}(\{0, 1\}^k, \{0, 1\}^n)\), \(f \xleftarrow{\$} \text{Func}(\mathcal{TW}_N, \{0, 1\}^k)\) and \(K_h \xleftarrow{\$} K_h\). Let \(T_2\) be the transcript in \(\text{World}2\) obtained by sampling \(E \xleftarrow{\$} \text{BC}(\{0, 1\}^k, \{0, 1\}^n)\), \(\tilde{P}_R \xleftarrow{\$} \text{Perm}(\mathcal{TW}_N, \{0, 1\}^n)\), \(f \xleftarrow{\$} \text{Func}(\mathcal{TW}_N, \{0, 1\}^k)\) and \(K_h \xleftarrow{\$} K_h\). We call a transcript \(\tau\) valid if an interaction with their oracles could render this transcript, namely, \(\Pr[T_1 = \tau] > 0\) for \(i \in \{1, 2\}\). Then \(\Pr[\text{World}1] - \Pr[\text{World}2]\) is upper-bounded by the statistical distance of transcripts, i.e.,

\[
\Pr[\text{World}1] - \Pr[\text{World}2] \leq \text{SD}(T_1, T_2) = \frac{1}{2} \sum_\tau |\Pr[T_1 = \tau] - \Pr[T_2 = \tau]|,
\]

where the sum is over all valid transcripts.

**Coefficient H Technique.** In this proof, the coefficient H technique [CS14, Pat08] is used, where valid transcripts \(T\) are partitioned into good transcripts \(T_{\text{good}}\) and bad transcripts \(T_{\text{bad}}\). Then \(\text{SD}(T_1, T_2)\) can be upper-bounded by the following lemma.

**Lemma 6.** Let \(0 \leq \varepsilon \leq 1\) be such that for all \(\tau \in T_{\text{good}}\), \(\frac{\Pr[T_1 = \tau]}{\Pr[T_2 = \tau]} \geq 1 - \varepsilon\). Then, \(\text{SD}(T_1, T_2) \leq \Pr[T_2 \in T_{\text{bad}}] + \varepsilon\).

Hereafter, good and bad transcripts are defined. Then \(\varepsilon\) and \(\Pr[T_2 \in T_{\text{bad}}]\) are upper-bounded. Finally, the upper-bound of the difference \(\Pr[\text{World}1] - \Pr[\text{World}2]\) is obtained by putting these upper-bounds into the above lemma.

**Good and Bad Transcripts.** Bad transcripts \(T_{\text{bad}}\) are defined so that one of the following conditions is satisfied, and good transcripts \(T_{\text{good}}\) are defined so that these conditions are not satisfied.

- hit \(\iff \exists \alpha \in [q], \beta \in [Q] \text{ s.t. } (w^\alpha, x^\alpha) = (W^\beta, X^\beta) \text{ or } (w^\alpha, y^\alpha) = (W^\beta, Y^\beta)\).
- coll \(\iff \exists \alpha, \beta \in [q] \text{ with } N^\alpha \neq N^\beta \text{ s.t. } w^\alpha = w^\beta\).

In \(\text{World}2\), \(w^\alpha\) is defined as \(w^\alpha \leftarrow f(N^\alpha)\). The first condition considers a collision in query-response triples among online and offline queries. The second condition considers a collision in query-response triples by online queries (more precisely, a collision in blockcipher’s keys).

**Upper-Bound of \(\Pr[T_2 \in T_{\text{bad}}]\).** By the definition of bad transcripts, \(\Pr[T_2 \in T_{\text{bad}}] = \Pr[\text{hit} \lor \text{coll}] \leq \Pr[\text{hit}] + \Pr[\text{coll}]\).

Note that these probabilities are considered in \(\text{World}2\). Hereafter, \(\Pr[\text{hit}]\) and \(\Pr[\text{coll}]\) are upper-bounded.
First, $\Pr[\text{hit}]$ is upper-bounded. Fix $\alpha \in [q]$ and $\beta \in [Q]$. Then $(w^\alpha, x^\alpha) = (W^\beta, X^\beta)$ implies
\[
(w^\alpha = f(N^\alpha) = W^\beta) \land (m^\alpha \oplus h_{K_h}(ctr^\alpha) = X^\beta).
\]
Since $w^\alpha$ is randomly drawn from $\{0, 1\}^k$, the probability that $w^\alpha = W^\beta$ is at most $1/2^k$. By the AXU hash function, the probability that $m^\alpha \oplus h_{K_h}(ctr^\alpha) = X^\beta$ is at most $\delta/2^k$. Similarly, the probability that $(w^\alpha, y^\alpha) = (W^\beta, Y^\beta)$ is at most $\delta/2^k$. By $\alpha \in [q]$ and $\beta \in [Q]$, we have $\Pr[\text{hit}] \leq 2qQ\delta/2^k$.

Next, $\Pr[\text{coll}]$ is upper-bounded. Since there are $q$ distinct first tweaks, we have $\Pr[\text{coll}] \leq \binom{q}{2}/2^k \leq q^2/2^{k+1}$.

Combining these upper-bounds gives
\[
\Pr[T_2 \in T_{\text{bad}}] \leq \frac{2qQ\delta}{2^k} + \frac{q^2}{2^{k+1}}.
\]

**Upper-Bound of $\varepsilon$.** Let $\tau \in T_{\text{good}}$ be a good transcript that does not satisfy hit and coll. Let $N_{\text{on}}(w)$ be the number of online queries such that the corresponding $f$’s outputs equal $w$ (in World1 $w$ is a blockcifer’s key defined by an online query), $N_{\text{off}}(W)$ the number of offline queries whose keys equal $W$.

Let $\text{all}_1$ (resp. $\text{all}_2$) be the set of all oracles in World1 (resp. World2). Let $\text{comp}_1(\tau)$ (resp. $\text{comp}_2(\tau)$) be the set of oracles compatible with $\tau$ in World1 (resp., World2). Then
\[
\Pr[T_1 = \tau] = \frac{|\text{comp}_1(\tau)|}{|\text{all}_1|} \quad \text{and} \quad \Pr[T_2 = \tau] = \frac{|\text{comp}_2(\tau)|}{|\text{all}_2|}.
\]

By $E \in \text{BC}([0,1]^k, \{0,1\}^n)$ and $f \in \text{Func}(TW_N, \{0,1\}^k)$, we have
\[
|\text{all}_1| = (2^n!)^2 \cdot |\text{Func}(TW_N, \{0,1\}^k)|.
\]

By $\tilde{P}_R \in \tilde{\text{Perm}}(TW_N, \{0,1\}^n)$, $E \in \text{BC}([0,1]^k, \{0,1\}^n)$ and $f \in \text{Func}(TW_N, \{0,1\}^k)$, we have
\[
|\text{all}_2| = (2^n)!^{TW_N} \cdot (2^n!)^2 \cdot |\text{Func}(TW_N, \{0,1\}^k)|.
\]

$|\text{comp}_1(\tau)|$ is counted. Let $N_{f,\tau}$ be the number of functions in Func($TW_N, \{0,1\}^k$) compatible with $\tau$. By $\neg$hit, blockcipher evaluations by online and offline queries don’t overlap with each other, thereby for each $w \in \{0,1\}^k$, the number of elements whose keys equal $w$ is $N_{\text{on}}(w) + N_{\text{off}}(w)$. $K_h$ is uniquely determined. Hence, we have
\[
|\text{comp}_1(\tau)| = N_{f,\tau} \cdot \prod_{w \in \{0,1\}^k} (2^n - N_{\text{on}}(w) - N_{\text{off}}(w))!.
\]

Next, $|\text{comp}_2(\tau)|$ is counted. Let $N_{E,\text{on}}(tw)$ be the number of online queries whose first tweaks equal $tw$.
\[
|\text{comp}_2(\tau)|
= N_{f,\tau} \cdot \prod_{tw \in TW_N} (2^n - N_{E,\text{on}}(tw))! \cdot \prod_{w \in \{0,1\}^k} (2^n - N_{\text{off}}(w))
= N_{f,\tau} \cdot (2^n)!^{TW_N} \cdot \prod_{w \in \{0,1\}^k} (2^n - N_{\text{on}}(w))! \cdot \prod_{w \in \{0,1\}^k} (2^n - N_{\text{off}}(w))!
\leq N_{f,\tau} \cdot (2^n)!^{TW_N} \cdot \prod_{w \in \{0,1\}^k} (2^n - N_{\text{on}}(w) - N_{\text{off}}(w))!.
\]
Regarding (5) = (6), since \( tw^\alpha = tw^\beta \Rightarrow w^\alpha = w^\beta \) and \( tw^\alpha \neq tw^\beta \Rightarrow \neg \text{coll} \), the equality is satisfied. Regarding (6) \( \leq \) (7), using \((2^n - a)! \cdot (2^n - b) \leq 2^n! \cdot (2^n - a - b)! \) for any \( 0 \leq a, b \leq 2^n \), the inequality is satisfied.

Finally,

\[
\frac{\Pr[\mathcal{T}_R = \tau]}{\Pr[\mathcal{T}_I = \tau]} = \frac{N_{f,\tau} \cdot \prod_{w \in \{0,1\}^k} (2^n - N_{\text{on}}(w) - N_{\text{off}}(w))!}{(2^n!)^{2k} \cdot |\text{Func}(\mathcal{T}W_N, \{0,1\}^k)|} \times \frac{(2^n!)^{\lceil \mathcal{T}W_N \rceil} \cdot (2^n!)^{2k} \cdot |\text{Func}(\mathcal{T}W_N, \{0,1\}^k)|}{N_{f,\tau} \cdot (2^n!)^{\lceil \mathcal{T}W_N \rceil} \cdot \prod_{w \in \{0,1\}^k} (2^n - N_{\text{on}}(w) - N_{\text{off}}(w))!} = 1.
\]

Thus we have \( \varepsilon = 0 \).

**Conclusion of the Proof.** Putting the above bounds to Lemma 6 gives

\[
\Pr[\text{World1}] - \Pr[\text{World2}] \leq \frac{2qQ\delta}{2^k} + \frac{q^2}{2^{k+1}}.
\]

\[\square\]

**Acknowledgements**

We would like to thank Jérémy Jean and the anonymous reviewers for their comments and suggestions.

**References**


