XOCB: Beyond-Birthday-Bound Secure Authenticated Encryption Mode with Rate-One Computation

Zhenzhen Bao[1,4][0000−0003−2839−6687], Seongha Hwang[2][0000−0002−2166−6421], Akiko Inoue[3][0000−0002−0173−7245], Byeonghak Lee[2][0000−0003−2736−6830], Jooyoung Lee[2][0000−0001−5471−9350], and Kazuhiko Minematsu[0000−0002−3427−6772][3]

[1] Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University, Beijing, China zzbao@tsinghua.edu.cn
[2] KAIST, Daejeon, Korea {mathience98,lbh0307,hicalf}@kaist.ac.kr
[3] NEC, Kawasaki, Japan {a_inoue,k-minematsu}@nec.com
[4] Zhongguancun Laboratory, Beijing, China

Abstract. We present a new block cipher mode of operation for authenticated encryption (AE), dubbed XOCB, that has the following features: (1) beyond-birthday-bound (BBB) security based on the standard pseudorandom assumption of the internal block cipher if the maximum block length is sufficiently smaller than the birthday bound, (2) rate-1 computation, and (3) supporting any block cipher with any key length. Namely, XOCB has effectively the same efficiency as the seminal OCB while having stronger quantitative security without any change in the security model or the required primitive in OCB. Although numerous studies have been conducted in the past, our XOCB is the first mode of operation to achieve these multiple goals simultaneously.

Keywords: Authenticated encryption, Block cipher, OCB, Beyond-birthday-bound security

1 Introduction

Authenticated Encryption. Since the formalization of authenticated encryption (AE) [6, 25, 36], constructing an efficient and secure AE[5] scheme has been one of the central topics in symmetric-key cryptography for decades. OCB, first proposed by Rogaway et al. at CCS 2001 [38], has been known to be a seminal scheme for its efficiency and security. OCB operates at rate 1, *i.e.*, each input block needs only one block cipher call used inside[6]. In addition, it is parallelizable. XOCB is much more efficient than the generic composition schemes

[5] We use the term AE to mean nonce-based AEAD [36] throughout the paper, unless otherwise stated.
[6] By convention, we ignore the constant number of block cipher calls per message.

that need at least two block cipher calls (thus rate $\leq 1/2$) and its variant, most notably GCM [1], which is specified by NIST SP800-38D and now quite widely deployed. The security of OCB can be reduced to the standard computational assumption on the block cipher used: namely, if the block cipher is a strong pseudorandom permutation (SPRP), OCB is shown to be provably secure. OCB has three versions [38, 37, 27], and the latest one (OCB3 [27]) is one of the winners of CAESAR competition[7] and is specified in RFC 7253. OCB3[8] has been implemented by OpenSSL and many other cryptographic libraries.

Beyond OCB. The security guarantee of (any version of) OCB is up to the birthday bound (upBB)[9], that is, if the internal block cipher has n-bit block, OCB is broken by attacks of data complexity $O(2^{n/2})$. This significantly limits the practical value of OCB with a small – most typically 64-bit – block cipher, because the limit of $2^{n/2}$ data per each secret key can be too severe. A very impactful exposition of such a risk is Sweet32 attack against TLS/SSL using 64-bit block ciphers [7]. Even if we use 128-bit block ciphers, such as AES, this is not a threat to the distant future.

For example, NIST[10] has recently been reviewing FIPS 197 (specifying AES), and several comments received in conjunction with this review process, more specifically from Microsoft and Amazon, warn that continued use of 128-bit block ciphers with GCM will be a problem in the near future. In particular, it is mentioned that exabyte ($10^{18} \approx 2^{60}$) data is already in use and zettabyte ($10^{21} \approx 2^{70}$) in the near future.

Transitioning to a new (possibly wide-block) cipher would not be easy and take time. If one wants to use AES (or, more generally, any n-bit block cipher where $n/2$-bit security can be a concern), a promising approach is to employ a *beyond-birthday-bound* (BBB) secure AE mode that resists attacks of complexity $O(2^{n/2})$. Moreover, the advancement of lightweight cryptography produces many block ciphers having application/platform-specific advantage over AES, in terms of various metrics, such as hardware size [10, 4], energy [3], latency [11], and software performance on low-end platforms [5]. To make it lightweight while achieving security equivalent to AES-128, it is quite often that these ciphers have key and block lengths at most 128 bits.

A BBB-secure AE mode has been extensively studied. Iwata proposed CHM [21], and CIP [22] that combine CENC [21], a BBB-secure nonce-based encryption mode, with a universal hash (UH) function using field multiplications. These schemes are provably secure under the standard pseudorandom assumption and roughly have $2n/3$-bit provable security. While the encryption part (CENC) is efficient, the need for the UH function makes the total cost (both for computation time and implementation memory) largely similar to GCM.

[7] https://competitions.cr.yp.to/caesar.html
[8] We may simply write OCB to mean OCB3.
[9] The second version OCB2 is flawed and allows devastating attacks, though a simple fix is possible [20].
[10] https://csrc.nist.gov/News/2022/proposal-to-revise-sp-800-38a

Another approach is to instantiate a tweakable block cipher (TBC) [28] using a block cipher and adopt a BBB-secure AE mode of a TBC as a template. The most popular template is ΘCB3, which has n-bit security using a TBC of n-bit block and about $3n$-bit tweak (required tweak length depends on the length of nonce and a maximum of message length etc.). If we instantiate such a TBC by an upBB-secure block cipher mode such as XEX, we obtain OCB3, and the resulting AE is also upBB-secure at best. Instantiating a BBB-secure TBC will break this barrier, however, is far from trivial. The cascaded LRW achieves BBB-security, but it needs two or more block cipher calls plus UH functions. Naito's XKX [32] requires a block cipher of more than n-bit keys and rekeying per nonce for BBB security. This allows us to use, say AES-256, but excludes a large number of lightweight ciphers for its key size as described above; hence it is not a perfect solution, and we cannot benefit from the state-of-the-art lightweight ciphers. Other TBC constructions, such as Mennink's F1 and F2 [29], or Jha *et al.*'s XHX [24] are efficient and work with a block cipher of about n-bit key. However, they need the ideal-cipher model for security reduction. Obtaining a standard security reduction for these constructions is considered to be hard [30]. This poses a non-trivial gap between GCM or OCB, which have been proved under the standard model. That is, the previous BBB-secure AE modes require either a significant increase in computation, making the rate close to 1/2, or a change in the cryptographic primitive supported by OCB, that is, an n-bit SPRP of any key length. The natural question here is *if we can achieve a BBB-secure AE maintaining the advantages of* OCB *as much as possible.*

Our Contributions. In this paper, we present a solution that answers the above question positively. Our proposal, dubbed XOCB, is an AE mode that can be based on an n-bit block cipher, and achieves BBB, namely $2n/3$-bit security for a constant maximum input length, assuming that the block cipher is an SPRP (for its use of both block cipher forward and inverse operations). The rate is one. Unlike XKX, XOCB does not need a rekeying while operating, making it possible to be instantiated with ciphers of k-bit keys for any k, and $k = n = 128$ allows using AES-128. When the maximum input length is not a constant, XOCB still maintains upBB security. Namely, it can securely encrypt a message of $\ll O(2^{n/2})$ blocks. In addition, XOCB is fully parallellizable as OCB. Despite numerous previous works, XOCB is the first mode of operation that achieves these goals[11]. See Table 1 for comparison.

The main innovation of XOCB is an encryption part that can be seen as an amalgamation of CENC and OCB's encryption part. We add one more output-masking layer to (a variant of) OCB's internal XEX mode throughout encryption or decryption. This additional mask is computed once for each nonce. Hence the rate is one. In more detail, for m-block message and a-block associated data (AD), XOCB needs $m+a$ plus 7 to 8 calls. The security depends on the maximum input length l (in n-bit blocks) and ranges between $n/2$ to $2n/3$ bits depending on the maximum length of a message. In more detail, the concrete bound is

[11] In concurrent to our work, Bhattacharjee, Bhaumik, and Nandi [8] presented an AE scheme combining SPRP and PRF that has some structural similarity to XOCB.

shown at Theorem 1, and its leading terms are $l\sigma/2^n + l\sigma^3/2^{2n}$ ignoring the constants, where σ denotes the total number of input blocks and l denotes the maximum input length in n bits. At first glance, the security improvement of XOCB over OCB appears limited because of the length factor. I.e., it is birthday-secure concerning l. However, many practical communication protocols specify a maximum packet length, also known as a Maximum transmission unit (MTU), that is not large. For example, the Internet Protocol (IP) has an MTU of 65535 ($= 2^{15}$) bytes. With this limit, XOCB with AES-128 can encrypt at most around $2^{80.3}$ bytes of input blocks, while OCB is limited to 2^{68} bytes. For low-power communication protocols, MTU is much smaller, such as 257 bytes for Bluetooth (specifically BLE 4.2). We also point out that XKX includes $l^2q/2^n$ [32, 31] in its bound for q queries, that is, a birthday term for l. We provide a numerical bound comparison for practical message lengths at Table 1. This exhibits the stronger security of XOCB for real-world use cases.

While the main routine of XOCB is structurally similar to CENC, we need a quite different analysis. This is because (1) the block cipher inputs in CENC are all determined by a single variable (nonce), while in our case, all inputs are determined by message blocks, independently for each block, and (2) the decryption of XOCB involves a block cipher inverse, which is absent in CENC. Note that CENC implements a nonce-based additive encryption by a BBB-secure expanding PRF. Hence the encryption and decryption are symmetric and do not need the block cipher inverse. These differences require us to develop a dedicated security analysis, which is much more involved than the case of CENC. We employ the framework developed by Kim et al. [26] for analyzing DBHtS MAC [15] (that is also based on the standard Coefficient-H) for proofs. This helps to reduce the proof complexity and gains accessibility, but it remains a lot of involved bad cases, which turns out to be a challenging task.

Finally, we stress that our security goal is the standard AE security under nonce-respecting adversaries. Due to its online computation algorithm, the nonce-misuse resistance security [39] is impossible to achieve by nature. Similarly, we do not claim security under the release of unverified plaintext (RUP) introduced by Andreeva et al. [2]. We consider classical single-user security, and analyzing multi-user security is left open.

Implementations. We present implementations of XOCB's AES instantiation on both high-end CPUs and low-end microprocessors to show its practical relevance. The implementation results show that on a modern 64-bit CPU (Intel's Tiger Lake family), AES-XOCB can encrypt and authenticate a 4096-byte message plus a 16-byte AD at a speed of 0.5 cycles per byte (cpb), while AES-ECB runs at 0.3 cpb at the same platform. Comparatively, AES-OCB and AES-CIP with the same implementation of AES executed at a speed of 0.4 and 1.2 cpb, respectively. On an 8-bit AVR processor (AVR ATmega328P), AES-XOCB requires 8556 bytes of ROM to support both encryption and decryption, and processes a 128-byte message plus a 16-byte AD at a speed of 306 cpb; In contrast, an opti-

mized implementation of AES-GCM requires 11012 bytes of ROM and executes at a speed of 880 cpb.

Table 1: Comparison of AE schemes that can use an n-bit block cipher of any key length. MUL denotes a field multiplication over $\mathsf{GF}(2^n)$. The cost of MUL depends on the platform and implementation, and we simply assume it is equivalent to the block cipher used. The "Security" column denotes the bit security ignoring the contribution of the maximum input length. The "Lead Terms" column denotes the leading terms in the nAE advantage (Refer to the main texts for more details).

Scheme	Primitive	Rate	Security	Lead Terms*	Ref
OCB	SPRP	1	$n/2$	$\sigma^2/2^n + q/2^{n\dagger}$	[27]
GCM	PRP, MUL	1/2	$n/2$	$\sigma^2/2^n + q/2^n$	[23, 33]
CHM,CIP	PRP, MUL	1/2	$2n/3$	$\sigma^3/2^{2n} + \sigma/2^n$	[21, 22]
XOCB	SPRP	1	$2n/3$	$l\sigma^3/2^{2n} + l\sigma/2^n$	This paper

* σ: total queried blocks in n-bit blocks, q: total number of queries, and l: the maximum block length of a query. We assume $O(1)$ AD blocks

† Bhaumik and Nandi [9] improved the bound with respect to the decryption queries

2 Preliminaries

Basic Notation. For a positive integer n, we write $N = 2^n$ and $[n] = \{1, \ldots, n\}$. For two nonnegative integers m and n such that $m \leq n$, we write $[m..n] = \{m, m+1, \ldots, n\}$. Given a nonempty set $\mathcal{X}$, $x \leftarrow_{\$} \mathcal{X}$ denotes that x is chosen uniformly randomly from $\mathcal{X}$. The set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ is denoted $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$, and the set of all permutations on $\mathcal{X}$ is denoted $\mathrm{Perm}(\mathcal{X})$. For simplicity, $\mathrm{Perm}(n)$ denotes the set of all permutations on $\{0,1\}^n$. For integers a and b such that $1 \leq a \leq b$, we write $(b)_a = b(b-1)\ldots(b-a+1)$, and $(b)_0 = 1$ by convention.

For a positive integer n, let $\{0,1\}^n$ be the set of n-bit strings and $\{0,1\}^{\leq n} = \bigcup_{i \in [0..n]} \{0,1\}^i$. Let 0^n be the string of n zero bits. Note that $0^0 = \varepsilon$. We write $\{0,1\}^*$ to denote the set of all arbitrary-length strings, including the empty string, and let $\{0,1\}^+ = \{0,1\}^* \setminus \{\varepsilon\}$. The set $\{0,1\}^n$ is sometimes regarded as a set of integers $\{0, 1, \ldots, 2^n - 1\}$ by converting an n-bit string $a_{n-1} \ldots a_1 a_0 \in \{0,1\}^n$ to an integer $2^{n-1}a_{n-1} + \cdots + 2a_1 + a_0$. An element $x \in \{0, 1, \ldots, 2^c - 1\}$ for some positive integer c may be denoted by $\langle x \rangle_c \in \{0,1\}^c$ following the above (standard) encoding. We also identify $\{0,1\}^n$ with a finite field $\mathsf{GF}(2^n)$ with 2^n elements, assuming that 2 cyclically generates all the nonzero elements of $\mathsf{GF}(2^n)$.

For $X \in \{0,1\}^*$, let $|X|$ be the bit length of X. For a positive integer n and $X \in \{0,1\}^+$, let $|X|_n = \lceil |X| / n \rceil$ where $\lceil x \rceil$ is the smallest integer y such

that $y \geq x$ and let $|\varepsilon|_n = 1$. For a positive integer n and a string $X \in \{0,1\}^*$, $(X_1, X_2, \ldots, X_m) \xleftarrow{n} X$ denotes that X is partitioned into strings $X_1, \ldots, X_m$, where $m = |X|_n$, $|X_1| = \cdots = |X_{m-1}| = n$, and $0 < |X_m| \leq n$ if $X \neq \varepsilon$, and $X_m = \varepsilon$ otherwise. For a positive integer n and $X \in \{0,1\}^*$, let $\mathsf{pad}(X) = X \,\|\, 1 \,\|\, 0^{n-(|X| \bmod n)-1}$. Note that pad is an injective function. For a positive integer n and $X \in \{0,1\}^*$, $\mathtt{ozp}(X)$ and $\overline{X}$ denote one-zero padding; $\mathtt{ozp}(X) = \overline{X} = X$ if $|X| = 0 \bmod n$, and $\mathtt{ozp}(X) = \overline{X} = \mathsf{pad}(X)$ if $|X| \neq 0 \bmod n$. For a positive integer $t \leq n$ and $X \in \{0,1\}^n$, $\mathtt{msb}_t(X)$ denotes a string of the most significant t bits of X. For $X, Y \in \{0,1\}^*$, let

$$X \oplus_{\mathtt{msb}} Y = \begin{cases} X \oplus \mathtt{msb}_{|X|}(Y) & \text{if } |X| < |Y|\,. \\ \mathtt{msb}_{|Y|}(X) \oplus Y & \text{if } |X| \geq |Y|\,. \end{cases}$$

Security Notions. Let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a keyed permutation with key space $\mathcal{K}$, where $E(K, \cdot)$ is a permutation for each $K \in \mathcal{K}$. We will denote $E_K(X)$ for $E(K, X)$. A (q,t)-distinguisher against E is an algorithm $\mathcal{D}$ with oracle access to an n-bit permutation and its inverse, making at most q oracle queries, running in time at most t, and outputting a single bit. The advantage of $\mathcal{D}$ in breaking the PRP-security of E, i.e., in distinguishing E from a uniform random permutation $\pi \leftarrow_{\$} \mathsf{Perm}(n)$, is defined as

$$\mathsf{Adv}^{\mathrm{sprp}}_{E}(\mathcal{D}) = \left| \Pr\left[K \leftarrow_{\$} \mathcal{K} : \mathcal{D}^{E_K, E_K^{-1}} = 1\right] - \Pr\left[\pi \leftarrow_{\$} \mathrm{Perm}(n) : \mathcal{D}^{\pi, \pi^{-1}} = 1\right] \right|.$$

In our security proof, the underlying block cipher E will be replaced by a truly random permutation up to the above adversarial distinguishing advantage.

Given key space $\mathcal{K}$, nonce space $\mathcal{N}$, associate data (AD) space $\mathcal{A}$, message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, and tag space $\mathcal{T}$, a nonce-based authenticated encryption (nAE) scheme is defined by a tuple

$$\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{C}, \mathsf{Enc}, \mathsf{Dec}),$$

where Enc and Dec denote encryption and decryption schemes, respectively. More precisely,

$$\begin{aligned} \mathsf{Enc} &: \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \longrightarrow \mathcal{C} \times \mathcal{T}, \\ \mathsf{Dec} &: \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \longrightarrow \mathcal{M} \cup \{\perp\}, \end{aligned}$$

where for $\mathsf{Enc}(K, N, A, M) = (C, T)$, we require $|C| = |M|$ and

$$\mathsf{Dec}(K, N, A, C, T') = \begin{cases} M & \text{if } T = T', \\ \perp & \text{otherwise.} \end{cases}$$

We will write $\mathsf{Enc}_K(N, A, M)$ and $\mathsf{Dec}_K(N, A, C)$ to denote $\mathsf{Enc}(K, N, A, M)$ and $\mathsf{Dec}(K, N, A, C)$, repectively. Throughout this paper, we will fix $\mathcal{N} = \{0,1\}^{n-2}$, $\mathcal{A} = \mathcal{M} = \mathcal{C} = \{0,1\}^*$ and $\mathcal{T} = \{0,1\}^n$.

Against the nonce-based authenticated encryption security of Π, an adversary $\mathcal{D}$ aims at distinguishing the real world $(\mathsf{Enc}_K, \mathsf{Dec}_K)$ and the ideal world $(\mathsf{Rand}, \mathsf{Rej})$, where Rand returns a random string of length $|M| + n$ for every encryption query $\mathsf{Enc}_K(N, A, M)$ and Rej always returns $\perp$ for every decryption query.

In this paper, we assume that $\mathcal{D}$ is nonce-respecting; it does not repeat nonces in *encryption* queries. Furthermore, $\mathcal{D}$ is non-trivial, i.e., $\mathcal{D}$ never repeats the same encryption/decryption query nor makes a decryption query (N, A, C, T) once (C, T) has been obtained by a previous encryption query $\mathsf{Enc}_K(N, A, M)$. Then the advantage of $\mathcal{D}$ against the nonce-based authenticated encryption security of Π is defined as

$$\mathsf{Adv}^{\mathrm{nAE}}_{\Pi}(\mathcal{D}) = \left|\Pr\left[K \leftarrow_{\$} \mathcal{K} : \mathcal{D}^{\mathsf{Enc}_K, \mathsf{Dec}_K} = 1\right] - \Pr\left[\mathcal{D}^{\mathsf{Rand}, \mathsf{Rej}} = 1\right]\right|.$$

We say that $\mathcal{D}$ is a (q_e, q_d, σ, l, t)-adversary against the nonce-based AE security of Π if $\mathcal{D}$ makes at most q_e encryption queries and at most q_d decryption queries, and running in time at most t, where the length of each encryption/decryption query (with a nonce and a tag excluded)[12] is at most l blocks of n bits. The total length of the encryption and decryption queries (with nonces and tags excluded) is at most σ blocks of n bits. When considering information-theoretic security, we will drop the parameter t.

Coefficient-H Technique. We will use Patarin's coefficient-H technique. The goal of this technique is to upper bound the adversarial distinguishing advantage between a real construction and its ideal counterpart. In the real (resp. ideal) world, an information-theoretic adversary $\mathcal{D}$ is allowed to make queries to a oracle denoted $\mathcal{O}_{\mathrm{real}}$ (resp. $\mathcal{O}_{\mathrm{ideal}}$). The interaction between $\mathcal{D}$ and the oracle determines a transcript. It contains all the information obtained during the interaction. We write that transcript τ is attainable if the probability of obtaining τ in the ideal world is non-zero. We also write T_{id} and T_{re} to denote the probability distribution of the transcript τ induced by the ideal world and the real world, respectively. By extension, we use the same notation to denote a random variable distributed according to each distribution.

We partition the set of attainable transcripts Γ into a set of "good" transcripts Γ_{good}, where the probability to obtain $\tau \in \Gamma_{\mathrm{good}}$ is close in the real world and the ideal world, and a set of "bad" transcripts Γ_{bad}, where the probability of obtaining $\tau \in \Gamma_{\mathrm{bad}}$ is small in the ideal world. Then the coefficient-H technique is summarized as the following lemma.

Lemma 1. *Let $\Gamma = \Gamma_{good} \sqcup \Gamma_{bad}$ be a partition of the set of attainable transcripts, where there exists a non-negative number ϵ_1 such that for any $\tau \in \Gamma_{good}$,*

$$\frac{\Pr\left[\mathsf{T}_{\mathrm{re}} = \tau\right]}{\Pr\left[\mathsf{T}_{\mathrm{id}} = \tau\right]} \geq 1 - \epsilon_1,$$

[12] More precisely, the block length of an encryption (resp. decryption) query is defined as $|A|_n + |M|_n$ (resp. $|A|_n + |C|_n$), while the length of the "empty" query is 1.

and there exists a non-negative number ϵ_2 *such that* $\Pr[\mathsf{T}_{\mathrm{id}} \in \Gamma_{bad}] \leq \epsilon_2$. *Then for any adversary* $\mathcal{D}$, *one has*

$$\left|\Pr\left[\mathcal{D}^{\mathcal{O}_{real}} = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{O}_{ideal}} = 1\right]\right| \leq \epsilon_1 + \epsilon_2,$$

where $\mathcal{D}^{\mathcal{O}_{real}}$ *and* $\mathcal{D}^{\mathcal{O}_{ideal}}$ *denote the adversarial outputs in the real and the ideal worlds, respectively.*

We refer to [19] for the proof of Lemma 1.

Extended Mirror Theory. Patarin's Mirror theory [34, 35] is a very powerful tool to estimate the number of solutions to a certain type of system of equations. At the beginning, there were some uncertainties in the proof of Mirror theory, but now there are several results on the full proof of Mirror theory up to n-bit security [17, 13, 14]. In this paper, we will use the *extended Mirror theory* [16, 18], which is a variant of Mirror theory, and estimates the number of solutions to a system of equations as well as non-equations.

We will represent a system of equations and non-equations by a graph. Each vertex corresponds to an n-bit *distinct* unknown. We will assume that the number of vertices is at most $2^n/4$, and by abuse of notation, identify the vertices with the values assigned to them. We distinguish two types of edges, namely, $=$-labeled edges and $\neq$-labeled edges that correspond to equations and non-equations, respectively. Each of the edges is additionally labeled by an element in $\{0,1\}^n$. So, if two vertices P and Q are adjacent by an edge with label $(\lambda, =)$ (resp. $(\lambda, \neq)$) for some $\lambda \in \{0,1\}^n$, then it would mean that $P \oplus Q = \lambda$ (resp. $P \oplus Q \neq \lambda$).

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$, where $\mathcal{E}^{=}$ and $\mathcal{E}^{\neq}$ denote the set of $=$-labeled edges and the set of $\neq$-labeled edges, respectively. Then $\mathcal{G}$ can be seen as a superposition of two subgraphs $\mathcal{G}^{=} =^{\mathrm{def}} (\mathcal{V}, \mathcal{E}^{=})$ and $\mathcal{G}^{\neq} =^{\mathrm{def}} (\mathcal{V}, \mathcal{E}^{\neq})$. Let $P \overset{\lambda}{-} Q$ denote a $(\lambda, =)$-labeled edge in $\mathcal{G}^{=}$. For $\ell > 0$ and a trail[13]

$$\mathcal{L} : P_0 \overset{\lambda_1}{-} P_1 \overset{\lambda_2}{-} \cdots \overset{\lambda_\ell}{-} P_\ell$$

in $\mathcal{G}^{=}$, its label is defined as

$$\lambda(\mathcal{L}) \overset{\mathrm{def}}{=} \lambda_1 \oplus \lambda_2 \oplus \cdots \oplus \lambda_\ell.$$

In this work, we will focus on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$ with certain properties, as listed below.

1. $\mathcal{G}^{=}$ contains no cycle.
2. $\lambda(\mathcal{L}) \neq \mathbf{0}$ for any trail $\mathcal{L}$ in $\mathcal{G}^{=}$.
3. If P and Q are connected with a $(\lambda, \neq)$-labeled edge, then they are not connected by a λ-labeled trail in $\mathcal{G}^{=}$.

[13] A trail is a walk in which all edges are distinct.

Any graph $\mathcal{G}$ satisfying the above properties will be called a *nice* graph. Given a nice graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$, an assignment of *distinct* values to the vertices in $\mathcal{V}$ satisfying all the equations in $\mathcal{E}^{=}$ and all the non-equations in $\mathcal{E}^{\neq}$ is called a *solution* to $\mathcal{G}$. We remark that if we assign any value to a vertex P, then $=$-labeled edges determine the values of all the other vertices in the component containing P in $\mathcal{G}^{=}$, where the assignment is unique since $\mathcal{G}^{=}$ contains no cycle. The values in the same component are all distinct since $\lambda(\mathcal{L}) \neq \mathbf{0}$ for any trail $\mathcal{L}$. Furthermore, any non-equation between two vertices in the same component will be redundant due to the third property above.

In the following lemma, we partition the set of vertices $\mathcal{V}$ into two disjoint sets, denoted $\mathcal{V}_{\mathsf{kn}}$ and $\mathcal{V}_{\mathsf{uk}}$, respectively, and fix an assignment of distinct values to the vertices in $\mathcal{V}_{\mathsf{kn}}$. Subject to this assignment, the number of possible assignments of distinct values to the vertices in $\mathcal{V}_{\mathsf{uk}}$ can be lower bounded (in a way that the entire assignment becomes a solution to $\mathcal{G}$).

Lemma 2. *For a positive integer q and a nonnegative integer v, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$ be a nice graph such that $|\mathcal{E}^{=}| = q$ and $|\mathcal{E}^{\neq}| = v$. Suppose that*

1. *$\mathcal{V}$ is partitioned into two subsets, denoted $\mathcal{V}_{\mathsf{kn}}$ and $\mathcal{V}_{\mathsf{uk}}$;*
2. *there is no $=$-labeled edge that is incident to a vertex in $\mathcal{V}_{\mathsf{kn}}$;*
3. *there is no $\neq$-labeled edge connecting two vertices in $\mathcal{V}_{\mathsf{kn}}$.*

Suppose that $\mathcal{G}^{=}_{\mathsf{uk}} = (\mathcal{V}_{\mathsf{uk}}, \mathcal{E}^{=})$ is decomposed into k components $\mathcal{C}_1, \ldots, \mathcal{C}_k$ for some k. Given a fixed assignment of distinct values to the vertices in $\mathcal{V}_{\mathsf{kn}}$, the number of solutions to $\mathcal{G}$, denoted $h(\mathcal{G})$, satisfies

$$\frac{h(\mathcal{G})N^q}{(N - |\mathcal{V}_{\mathsf{kn}}|)_{|\mathcal{V}_{\mathsf{uk}}|}} \geq 1 - \frac{|\mathcal{V}|^2}{N^2} \sum_{i=1}^{k} |\mathcal{C}_i|^2 - \frac{2v}{N}.$$

We refer to [12] for the proof of Lemma 2.

3 Description of XOCB

We define our proposed scheme XOCB. The algorithms are shown in Figs. 1 and 2, and the figures are shown in Figs. 3 and 4. Below we describe the encryption of XOCB. For the decryption, please refer to Figs. 1 and 2.

Given an n-bit block cipher E, the encryption routine of XOCB takes a triple of nonce, associate data and message $(N, A, M) \in \{0,1\}^{n-2} \times \{0,1\}^* \times \{0,1\}^*$ by computing $(C, T) \in \{0,1\}^* \times \{0,1\}^n$ as follows. Here, $|C| = |M|$ holds for any M.

1. Break the associated data A and the message M into n-bit blocks:

$$(A[1], \ldots, A[a]) \xleftarrow{n} \mathsf{pad}(A),$$
$$(M[1], \ldots, M[m]) \xleftarrow{n} M.$$

   Note that $0 \leq |M[m]| \leq n$ and $|M[\alpha]| = n$ for $\alpha \in [m-1]$.

2. Compute masking values:

$$\begin{aligned}\Delta_1 &= E_K(N \,\|\, \langle 0\rangle_2) \oplus E_K(N \,\|\, \langle 1\rangle_2),\\ \Delta_2 &= E_K(N \,\|\, \langle 0\rangle_2) \oplus E_K(N \,\|\, \langle 2\rangle_2),\\ \Delta_3 &= E_K(N \,\|\, \langle 0\rangle_2) \oplus E_K(N \,\|\, \langle 3\rangle_2).\end{aligned}$$

3. Compute the inputs and the outputs for block cipher calls:
   (a) for $\alpha \in [0..m]$,

$$X[\alpha] = \begin{cases} 2^\alpha \Delta_1 \oplus \Delta_2 & \text{if } \alpha = 0,\\ 2^\alpha \Delta_1 \oplus \Delta_2 \oplus M[\alpha] & \text{if } \alpha > 0, \text{ and } |M[\alpha]| = n,\\ 2^\alpha \Delta_1 & \text{if } \alpha = m \text{ and } |M[m]| < n,\end{cases}$$
$$Y[\alpha] = E_K(X[\alpha]);$$

   (b) for $\alpha \in [a]$, $U[\alpha] = 2^\alpha \Delta_2 \oplus A[\alpha]$, and $V[\alpha] = E_k(U[\alpha])$;
   (c) for $\alpha \in \{0, 1\}$,

$$P[\alpha] = \begin{cases} 2^m \Delta_1 \oplus 2^\alpha \Delta_3 & \text{if } \alpha = 0,\\ 2^m \Delta_1 \oplus 2^\alpha \Delta_3 \oplus \bigoplus_{i\in[m]} \overline{M[i]}; & \text{if } \alpha = 1,\end{cases}$$
$$Q[\alpha] = E_K(P[\alpha]).$$

4. Compute ciphertext C and tag T:
   (a) for $\alpha \in [m]$,

$$C[\alpha] = \begin{cases} Y[0] \oplus Y[\alpha] \oplus (2^\alpha + 1)\Delta_1 & \text{if } |M[\alpha]| = n,\\ (Y[0] \oplus Y[m] \oplus (2^m + 1)\Delta_1 \oplus \Delta_2) \oplus_{\mathtt{msb}} M[m] & \text{otherwise;}\end{cases}$$

   (b) output (C, T) where

$$\begin{aligned} C &= C[1] \,\|\, \cdots \,\|\, C[m],\\ T &= Q[0] \oplus Q[1] \oplus 3\Delta_3 \oplus \bigoplus_{\alpha\in[a]} V[\alpha].\end{aligned}$$

XOCB AND OCB. The major difference between XOCB from OCB is its additional output masking. In more detail, in its message encryption, XOCB adds an extra masking to the ciphertext blocks so that each ciphertext block can be viewed as a sum of two XEX outputs:

$$C = E(M \oplus \Delta) \oplus \Delta \oplus E(\Delta') \oplus \Delta'.$$

Since each ciphertext block is built from two block cipher calls, unlike OCB, XOCB allows a single collision of input blocks between two queries. Instead, ciphertext blocks in a single query share additional masking, so one can break XOCB if there exists an input collision in a single query, and this is the fundamental reason why the security bound of XOCB is given as $\sigma l/2^n$ instead of $\sigma^2/2^n$ for OCB.

Algorithm $\mathsf{XOCB}.\mathcal{E}_{E_K}(N, A, M)$

$\Sigma \leftarrow 0^n$
$(\Delta_1, \Delta_2, \Delta_3) \leftarrow \mathsf{Init}_{E_K}(N)$
$L \leftarrow \mathsf{XEXX}_{E_K}(0^n, \Delta_1 \oplus \Delta_2, 0^n)$
$(M[1], \ldots, M[m]) \xleftarrow{n} M$
for $i = 1$ **to** $m - 1$
  $\Delta_1 \leftarrow 2\Delta_1$
  $C[i] \leftarrow \mathsf{XEXX}_{E_K}(M[i], \Delta_1 \oplus \Delta_2, L)$
  $\Sigma \leftarrow \Sigma \oplus M[j]$
end for
$\Delta_1 \leftarrow 2\Delta_1$
if $|M[m]| = n$ **then**
  $C[i] \leftarrow \mathsf{XEXX}_{E_K}(M[i], \Delta_1 \oplus \Delta_2, L)$
else
  $Z \leftarrow \mathsf{XEXX}_{E_K}(0^n, \Delta_1, L)$
  $C[m] \leftarrow \mathtt{msb}_{|M[m]|}(Z) \oplus M[m]$
end if
$\Delta_1^* \leftarrow \Delta_1 \oplus \Delta_3$
$\Delta_2^* \leftarrow \Delta_1 \oplus 2\Delta_3$
$\Sigma \leftarrow \Sigma \oplus \mathtt{ozp}(M[m])$
$C \leftarrow C[1] \| \ldots \| C[m]$
$T \leftarrow \mathsf{XEXX}_{E_K}(0^n, \Delta_1^*, L) \oplus \mathsf{XEXX}_{E_K}(\Sigma, \Delta_2^*, L)$
$\Gamma \leftarrow \mathsf{PHASH}_{E_K}(A, \Delta_2)$
$T \leftarrow T \oplus \Gamma$
return (C, T)

Algorithm $\mathsf{XOCB}.\mathcal{D}_{E_K}(N, A, C, T)$

$\Sigma \leftarrow 0^n$
$(\Delta_1, \Delta_2, \Delta_3) \leftarrow \mathsf{Init}_{E_K}(N)$
$L \leftarrow \mathsf{XEXX}_{E_K}(0^n, \Delta_1 \oplus \Delta_2, 0^n)$
$(C[1], \ldots, C[m]) \xleftarrow{n} C$
for $i = 1$ **to** $m - 1$
  $\Delta_1 \leftarrow 2\Delta_1$
  $M[i] \leftarrow \mathsf{XEXX}^{-1}_{E_K}(C[i], \Delta_1 \oplus \Delta_2, L)$
  $\Sigma \leftarrow \Sigma \oplus M[j]$
end for
$\Delta_1 \leftarrow 2\Delta_1$
if $|C[m]| = n$ **then**
  $M[i] \leftarrow \mathsf{XEXX}^{-1}_{E_K}(C[i], \Delta_1 \oplus \Delta_2, L)$
else
  $Z \leftarrow \mathsf{XEXX}_{E_K}(0^n, \Delta_1, L)$
  $M[m] \leftarrow \mathtt{msb}_{|C[m]|}(Z) \oplus C[m]$
end if
$\Delta_1^* \leftarrow \Delta_1 \oplus \Delta_3$
$\Delta_2^* \leftarrow \Delta_1 \oplus 2\Delta_3$
$\Sigma \leftarrow \Sigma \oplus \mathtt{ozp}(M[m])$
$M \leftarrow M[1] \| \ldots \| M[m]$
$\widehat{T} \leftarrow \mathsf{XEXX}_{E_K}(0^n, \Delta_1^*, L) \oplus \mathsf{XEXX}_{E_K}(\Sigma, \Delta_2^*, L)$
$\Gamma \leftarrow \mathsf{PHASH}_{E_K}(A, \Delta_2)$
$\widehat{T} \leftarrow \widehat{T} \oplus \Gamma$
if $\widehat{T} = T$ **then return** M
else return $\perp$

Fig. 1: Algorithms of XOCB. Subroutines are shown at Fig. 2.

Algorithm $\mathsf{Init}_{E_K}(N)$

$\Delta_1 \leftarrow E_K(N \| \langle 0 \rangle_2) \oplus E_K(N \| \langle 1 \rangle_2)$
$\Delta_2 \leftarrow E_K(N \| \langle 0 \rangle_2) \oplus E_K(N \| \langle 2 \rangle_2)$
$\Delta_3 \leftarrow E_K(N \| \langle 0 \rangle_2) \oplus E_K(N \| \langle 3 \rangle_2)$
return $(\Delta_1, \Delta_2, \Delta_3)$

Algorithm $\mathsf{XEXX}_{E_K}(X, S, V)$

$Y \leftarrow E_K(X \oplus S) \oplus S \oplus V$
return Y

Algorithm $\mathsf{XEXX}^{-1}_{E_K}(Y, S, V)$

$X \leftarrow E_K^{-1}(Y \oplus S \oplus V) \oplus S$
return X

Algorithm $\mathsf{PHASH}_{E_K}(A, \Delta)$

$\Sigma \leftarrow 0^n$
$(A[1], \ldots, A[a]) \xleftarrow{n} A$
for $i = 1$ **to** a
  $\Delta \leftarrow 2\Delta$
  $\Sigma \leftarrow \Sigma \oplus E_K(\mathtt{ozp}(A[i]) \oplus \Delta)$
if $|A[a]| = n$
  $\Delta \leftarrow 2\Delta$
  $\Sigma \leftarrow \Sigma \oplus E_K(10^{n-1} \oplus \Delta)$
end if
return Σ

Fig. 2: Subroutines for XOCB.

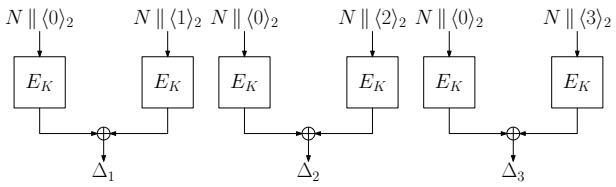

Fig. 3: Generation of masking values for XOCB.

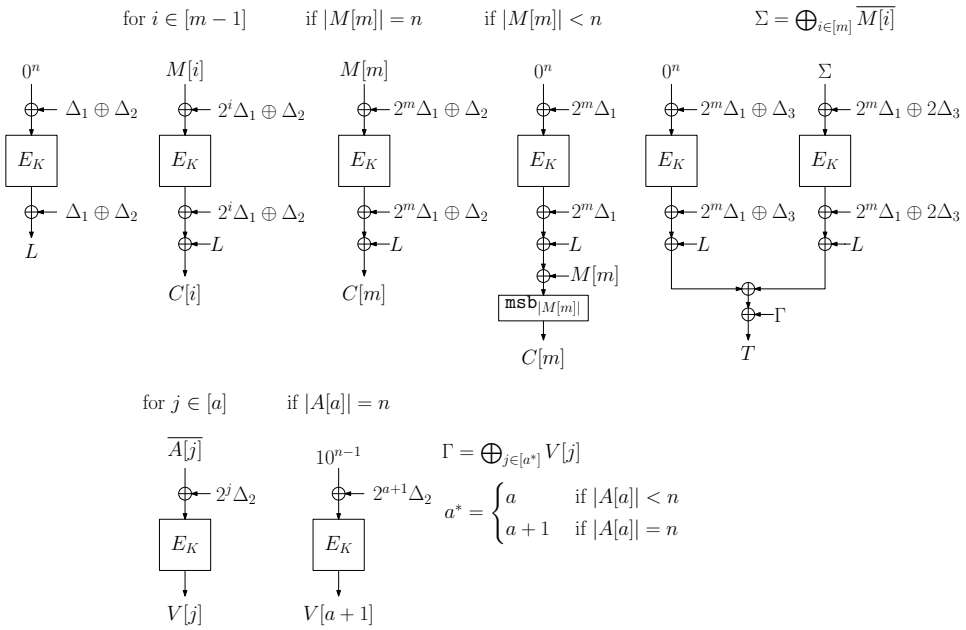

Fig. 4: Encryption of XOCB. (Top) Encryption of plaintext. (Bottom) Processing of Associated data. For $X \in \{0,1\}^{\leq n}$, $\overline{X}$ denotes the one-zero padding (see Section 2). The computation of T involves redundant output mask values, which is omitted in the text description of Section 3.

4 Security of XOCB

Let XOCB[π] denote an idealized version of XOCB where the underlying n-bit keyed block cipher E_K is replaced by a random n-bit (secret) permutation π. We can prove the security of XOCB[π] as follows. Deriving the standard model security bound by using a block cipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ (for a certain key space $\mathcal{K}$) instead of π is standard, thus omitted here.

Theorem 1. *Let $\mathcal{D}$ be a (q_e, q_d, σ, l)-adversary against* nAE*-security of* XOCB[π] *(see . Then we have*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{nAE}}_{\mathsf{XOCB}[\pi]}(\mathcal{D}) \le & \frac{28q + 2\sigma + 1.5l(q+\sigma)}{2^n} \\ & + \frac{4q\sigma^2 + (30q^2 + 10q)\sigma + 93q^3 + 44q^2}{2^{2n}} \\ & + \frac{(9\sigma^3 + 8\sigma^2 q + 45\sigma q^2 + 6q^3)l}{2^{2n+1}},\end{aligned}$$

where $q = q_e + q_d$.

As defined in Section 2 (Security Notion), q_e denotes the number of encryption queries, q_d denotes the number of decryption queries, l denotes the maximum query length in n-bit blocks, and σ denotes the total queried blocks in n-bit blocks.

The leading terms in the bound of Theorem 1 are $l \cdot \sigma/2^n + l \cdot \sigma^3/2^{2n}$, hence XOCB achieves $2n/3$-bit security if $l = O(1)$. In general, it achieves BBB security if l is sufficiently smaller than $2^{n/2}$. As mentioned earlier, the previous schemes such as XKX have a similar limitation on input length. From the next subsection, we provide the proof of Theorem 1.

Bound Comparison. To get an idea on how XOCB improves security in the practical use cases, we show a quick comparison of bounds in Figure 5 for the case $n = 128$. We note that providing a precise and compact comparison is fairly difficult as each scheme employs different parameters. To make it compact, we apply our notations of l and σ to the bound of each mode, focusing on the leading terms (shown in Table 1) and ignoring the constants. We assume no tag truncation and $O(1)$-block AD. Furthermore, we assume $q_e = q_d$ and that all the messages are of the same length, thus $lq = \sigma$. As we mentioned in Introduction, we observed a significant gain over GCM/OCB if l is not large ($l = 2^8$, about 4Kbyte). If l is large ($l = 2^{30}$, about 17 GBytes), the gain of XOCB is reduced but still remains. CIP offers stronger security, in particular for the latter case. However, it is costlier than XOCB for the use of a universal hash function.

In the full version, we also present graphs for the aforementioned settings taking constants into consideration to see their effect on the bound. It turns out that the bounds of OCB and XOCB do not change significantly.

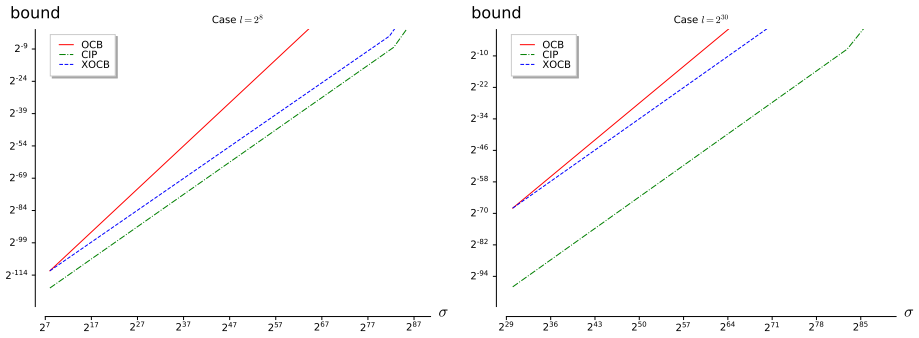

Fig. 5: nAE bound comparison. (Left) $l = 2^8$ (Right) $l = 2^{30}$. The bound of GCM is identical to that of OCB in our setting, hence omitted.

4.1 Proof Setup

Let $\mathcal{D}$ be a (q_e, q_d, σ, l)-adversary against the nAE-security of $\mathsf{XOCB}[\pi]$. We assume that $\mathcal{D}$ does not make any redundant query and makes exactly q_e encryption queries and q_d decryption queries without loss of generality. Let

$$\tau_e = (N_i, A_i, M_i, C_i, T_i)_{i\in[q_e]}$$
$$\tau_d = (N'_j, A'_j, C'_j, T'_j, b'_j)_{i\in[q_d]}$$

denote the list of encryption queries and decryption queries, respectively. Note that $\mathcal{D}$ always has $b'_j = \perp$ for $j \in [q_d]$ if $\mathcal{D}$ interacts with the ideal oracle. At the end of the game, we assume that the real world oracle reveals all the inputs and the outputs for π calls made during the query phase so the (extended) transcript is of the form $\tau = (\tau_e, \tau_d, \Pi)$, where Π denotes the set of the permutation input and output pairs on π. In the ideal world, the corresponding values should be carefully sampled and revealed to the adversary. The sampling process is described in Section 4.2.

For $i \in [q_e]$ (resp. $j \in [q_d]$), let $m_i = |M_i|_n$ (resp. $m'_j = |M'_j|_n$) be the number of blocks in M_i (resp. M'_j), and let $a_i = |\mathsf{pad}(A_i)|_n$ (resp. $a'_j = |\mathsf{pad}(A'_j)|_n$) be the number of blocks in A_i (resp. A'_j). Let $l_i = m_i + a_i$ and let $l'_j = m'_j + a'_j$. For $i \in [q_e]$, A_i, M_i and C_i are divided into n-bit blocks, written as follows.

$$(A_i[1], \ldots, A_i[a_i]) \xleftarrow{n} \mathsf{pad}(A_i),$$
$$(M_i[1], \ldots, M_i[m_i]) \xleftarrow{n} M_i,$$
$$(C_i[1], \ldots, C_i[m_i]) \xleftarrow{n} C_i.$$

Similarly, for $i \in [q_d]$, we write

$$(A'_i[1], \ldots, A'_i[a'_i]) \xleftarrow{n} \mathsf{pad}(A'_i),$$
$$(C'_i[1], \ldots, C'_i[m'_i]) \xleftarrow{n} C'_i.$$

Let $q = q_e + q_d$. We define N_i, A_i, and a_i for $i \in [q]$ by letting $N_{j+q_e} = N'_j$, $A_{j+q_e} = A'_j$, and $a_{j+q_e} = a'_j$ for $j \in [q_d]$. With this extension, we can write

$$(A_i[1], \ldots, A_i[a_i]) \xleftarrow{n} \mathsf{pad}(A_i)$$

for $i \in [q]$, where $A_{j+q_e}[\alpha] = A'_j[\alpha]$ for $j \in [q_d]$, and $\alpha \in [a'_j]$.

For π calls made in the i-th encryption query, we use the following notations:

- for $\alpha \in [m_i]$, $X_i[\alpha]$ and $Y_i[\alpha]$ denote the input and output of π, respectively, corresponding to $M_i[\alpha]$;
- for $\alpha \in [a_i]$, $U_i[\alpha]$ and $V_i[\alpha]$ denote the input and output of π, respectively, corresponding to $A_i[\alpha]$;
- $(P_i[0], P_i[1])$ and $(Q_i[0], Q_i[1])$ denote the pairs of inputs, and the pairs of outputs corresponding to the two π calls for tag generation.

Similarly, for π calls made in the i-th decryption query, we use the following notations:

- for $\alpha \in [m'_i]$, $X'_i[\alpha]$ and $Y'_i[\alpha]$ denote the input and output of π, respectively, corresponding to $C'_i[\alpha]$;
- for $\alpha \in [m'_i]$, $M'_i[\alpha]$ denote the message block corresponding to $C'_i[\alpha]$;
- for $\alpha \in [a'_i]$, $U'_i[\alpha] (= U_{i+q_e}[\alpha])$ and $V'_i[\alpha] (= V_{i+q_e}[\alpha])$ denote the input and output of π, respectively, corresponding to $A'_i[\alpha]$;
- $(P'_i[0], P'_i[1])$ and $(Q'_i[0], Q'_i[1])$ denote the pairs of inputs, and the pairs of outputs corresponding to the two π calls for tag generation.

4.2 Simulating π in the Ideal World

In the ideal world, the underlying π is simulated at the end of the attack. The π-evaluations are recorded in a set Π, initialized as the empty set. The π-evaluations are sampled consistently with all the encryption and decryption queries made during the attack. In other words, such evaluations will uniquely determine all the queries. Whenever an evaluation $\pi(X) = Y$ is fixed, (X, Y) will be included in Π. In this way, Π grows. The set of inputs X (resp. outputs Y) of Π will be denoted $\mathsf{dom}(\pi)$ (resp. $\mathsf{rng}(\pi)$). We now describe the sampling process, which might abort if a certain bad event happens.

Step 1. For each $i \in [q_e]$, $\Delta_{i,1}$, $\Delta_{i,2}$, $\Delta_{i,3}$ are sampled uniformly at random from $\{0,1\}^n$. For each $j \in [q_d]$, $(\Delta'_{j,1}, \Delta'_{j,2}, \Delta'_{j,3})$ is set to $(\Delta_{i,1}, \Delta_{i,2}, \Delta_{i,3})$ if $N_i = N'_j$ for some $i \in [q_e]$, and otherwise $\Delta'_{j,1}$, $\Delta'_{j,2}$, $\Delta'_{j,3}$ are sampled uniformly at random from $\{0,1\}^n$.

Let $(\Delta_{i+q_e,1}, \Delta_{i+q_e,2}, \Delta_{i+q_e,3}) = (\Delta'_{i,2}, \Delta'_{i,3}, \Delta'_{i,3})$ for $i \in [q_d]$. For $i \in [q]$ and $\alpha \in [0..3]$, we will write $N_{i,\alpha} = N_i \,\|\, \langle \alpha \rangle_2$. Let

$$\begin{aligned}
\mathcal{P} &\stackrel{\text{def}}{=} \{(i,\alpha) : i \in [q_e], \alpha \in [0..m_i]\}\,,\\
\mathcal{P}_{\$} &\stackrel{\text{def}}{=} \{(i,m_i) : i \in [q_e], |M_i[m_i]| < n\}\,,\\
\mathcal{P}_2 &\stackrel{\text{def}}{=} \{(i,\alpha,\beta) : (i,\alpha),(i,\beta) \in \mathcal{P}, \alpha \neq \beta\}\,,\\
\mathcal{N} &\stackrel{\text{def}}{=} \{(i,\alpha) : i \in [q], \alpha \in [0..3]\}\,,\\
\mathcal{N}_2 &\stackrel{\text{def}}{=} \{(i,\alpha,\beta) : (i,\alpha),(i,\beta) \in \mathcal{N}, \alpha \neq \beta\}\,.
\end{aligned}$$

For each $(i,m_i) \in \mathcal{P}_{\$}$, $\mathbf{s}_i$ is sampled uniformly at random from $\{0,1\}^{n-|M_i[m_i]|}$. For $(i,\alpha) \in \mathcal{P}$, set:

$$X_i[\alpha] = \begin{cases} \Delta_{i,1} \oplus \Delta_{i,2} & \text{if } \alpha = 0,\\ 2^\alpha \Delta_{i,1} & \text{if } (i,\alpha) \in \mathcal{P}_{\$},\\ 2^\alpha \Delta_{i,1} \oplus \Delta_{i,2} \oplus M_i[\alpha] & \text{otherwise;} \end{cases}$$

$$Z_i[\alpha] = \begin{cases} 0 & \text{if } \alpha = 0,\\ (2^\alpha+1)\Delta_{i,1} \oplus \Delta_{i,2} \oplus ((C_i[\alpha] \oplus M_i[\alpha]) \,\|\, \mathbf{s}_i) & \text{if } (i,\alpha) \in \mathcal{P}_{\$},\\ (2^\alpha+1)\Delta_{i,1} \oplus C_i[\alpha] & \text{otherwise.} \end{cases}$$

We now define a bad event as follows.

$$\mathsf{badA} \Leftrightarrow \mathsf{badA}_1 \vee \mathsf{badA}_2 \vee \mathsf{badA}_3 \vee \mathsf{badA}_4 \vee \mathsf{badA}_5,$$

where

- $\mathsf{badA}_1 \Leftrightarrow$ there exists $(i,\alpha,\beta) \in \mathcal{P}_2$ such that $X_i[\alpha] = X_i[\beta]$;
- $\mathsf{badA}_2 \Leftrightarrow \mathsf{badA}_{2a} \vee \mathsf{badA}_{2b} \vee \mathsf{badA}_{2c} \vee \mathsf{badA}_{2d}$, where
  - $\mathsf{badA}_{2a} \Leftrightarrow$ there exist $(i,\alpha,\beta) \in \mathcal{P}_2$, $(j,\alpha'),(k,\beta') \in \mathcal{P}$ such that $X_i[\alpha] = X_j[\alpha']$ and $X_i[\beta] = X_k[\beta']$;
  - $\mathsf{badA}_{2b} \Leftrightarrow$ there exist $(i,\alpha,\beta) \in \mathcal{P}_2$, $(j,\alpha') \in \mathcal{P}$, $(k,\beta') \in \mathcal{N}$ such that $X_i[\alpha] = X_j[\alpha']$ and $X_i[\beta] = N_{k,\beta'}$;
  - $\mathsf{badA}_{2c} \Leftrightarrow$ there exist $(i,\alpha,\beta) \in \mathcal{P}_2$, $(j,\alpha'),(k,\beta') \in \mathcal{N}$ such that $X_i[\alpha] = N_{j,\alpha'}$ and $X_i[\beta] = N_{k,\beta'}$;
  - $\mathsf{badA}_{2d} \Leftrightarrow$ there exist $(i,\alpha,\beta) \in \mathcal{N}_2$, $(j,\alpha'),(k,\beta') \in \mathcal{P}$ such that $N_{i,\alpha} = X_j[\alpha']$ and $N_{i,\beta} = X_k[\beta']$;
- $\mathsf{badA}_3 \Leftrightarrow \mathsf{badA}_{3a} \vee \mathsf{badA}_{3b}$, where
  - $\mathsf{badA}_{3a} \Leftrightarrow$ there exist three distinct $(i,\alpha),(j,\beta),(k,\gamma) \in \mathcal{P}$ such that $X_i[\alpha] = X_j[\beta] = X_k[\gamma]$;
  - $\mathsf{badA}_{3b} \Leftrightarrow$ there exist distinct $(i,\alpha),(j,\beta), \in \mathcal{P}$ and $(k,\gamma) \in \mathcal{N}$ such that $X_i[\alpha] = X_j[\beta] = N_{k,\gamma}$;
- $\mathsf{badA}_4 \Leftrightarrow \mathsf{badA}_{4a} \vee \mathsf{badA}_{4b}$, where

- $\mathsf{badA}_{4a} \Leftrightarrow$ there exists $(i, \alpha, \beta) \in \mathcal{P}_2$ such that $Z_i[\alpha] = Z_i[\beta]$;
- $\mathsf{badA}_{4b} \Leftrightarrow$ there exist $i \in [q]$, $(\alpha, \beta) \in [3]^{*2}$ such that either $\Delta_{i,\alpha} = 0$ or $\Delta_{i,\alpha} = \Delta_{i,\beta}$;

– $\mathsf{badA}_5 \Leftrightarrow \mathsf{badA}_{5a} \vee \mathsf{badA}_{5b}$, where
- $\mathsf{badA}_{5a} \Leftrightarrow$ there exist distinct $(i, \alpha, \alpha'), (j, \beta, \beta') \in \mathcal{P}_2$ such that $X_i[\alpha] = X_j[\beta]$ and $Z_i[\alpha] \oplus Z_i[\alpha'] = Z_j[\beta] \oplus Z_j[\beta']$;
- $\mathsf{badA}_{5b} \Leftrightarrow$ there exist $(i, \alpha, \alpha') \in \mathcal{P}_2$, $(j, \beta, \beta') \in \mathcal{N}_2$ such that $X_i[\alpha] = N_{j,\beta}$ and

$$Z_i[\alpha] \oplus Z_i[\alpha'] = \begin{cases} \Delta_{j,\beta} & \text{if } \beta' = 0, \\ \Delta_{j,\beta'} & \text{if } \beta = 0, \\ \Delta_{j,\beta} \oplus \Delta_{j,\beta'} & \text{otherwise.} \end{cases}$$

If badA occurs, then the sampling process aborts.

Step 2. In this step, we construct a system of equations in Y-variables, representing the images of X-variables under π. For $(i, \alpha) \in \mathcal{P}$, let $Y_i[\alpha] = \pi(X_i[\alpha])$. It should be the case that

$$Y_i[\alpha] \oplus Y_i[0] = Z_i[\alpha]$$

for each $\alpha > 0$. Let $\mathcal{L}$ denote a system of equations obtained by collecting all these equations, as well as

$$\begin{aligned} \pi(N_{i,0}) \oplus \pi(N_{i,1}) &= \Delta_{i,1}, \\ \pi(N_{i,0}) \oplus \pi(N_{i,2}) &= \Delta_{i,2}, \\ \pi(N_{i,0}) \oplus \pi(N_{i,3}) &= \Delta_{i,3} \end{aligned}$$

for $i \in [q]$. A solution to $\mathcal{L}$ is sampled uniformly at random from the set of all solutions to $\mathcal{L}$, and the corresponding π-evaluations are included in Π. We will show later that a solution to $\mathcal{L}$ does exist as long as badA does not happen.

Step 3. In this step, we handle the associated data. For $i \in [q]$ and $\alpha \in [a_i]$, set $U_i[\alpha] = 2^\alpha \Delta_{i,2} \oplus A_i[\alpha]$ and

– $V_i[\alpha] = \pi(U_i[\alpha])$ if $U_i[\alpha] \in \mathsf{dom}(\pi)$,
– $V_i[\alpha] \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{rng}(\pi)$ otherwise, where $\pi(U_i[\alpha]) = V_i[\alpha]$ is added to Π.

Step 4. In this step, we handle the decryption queries. Let

$$\begin{aligned} \mathcal{P}' &\stackrel{\text{def}}{=} \{(i, \alpha) : i \in [q_d], \alpha \in [m]\}, \\ \mathcal{P}'_0 &\stackrel{\text{def}}{=} \{(i, 0) : i \in [q_d]\}, \\ \mathcal{P}'_{\$} &\stackrel{\text{def}}{=} \{(i, m'_i) \in \mathcal{P}' : i \in [q_d], |C'_i[m'_i]| < n\}. \end{aligned}$$

For $(i, \alpha) \in \mathcal{P}'$, set:

$$Z'_i[\alpha] = \begin{cases} 0 & \text{if } \alpha = 0, \\ ((2^\alpha + 1)\Delta'_{i,1} \oplus \Delta'_{i,2}) \oplus_{\mathtt{msb}} C'_i[\alpha] & \text{if } (i, \alpha) \in \mathcal{P}'_{\$}, \\ (2^\alpha + 1)\Delta'_{i,1} \oplus C'_i[\alpha] & \text{otherwise.} \end{cases}$$

For $(i,\alpha) \in \mathcal{P}'_{\$} \cup \mathcal{P}'_0$, set $X'_i[\alpha] = \Delta'_i[\alpha]$ and

- $Y'_i[\alpha] = \pi(X'_i[\alpha])$ if $X'_i[\alpha] \in \mathsf{dom}(\pi)$,
- $Y'_i[\alpha] \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{rng}(\pi)$ otherwise, where $\pi(X'_i[\alpha]) = Y'_i[\alpha]$ is added to Π.

Next, for $(i,\alpha) \in \mathcal{P}' \setminus (\mathcal{P}'_{\$} \cup \mathcal{P}'_0)$, set $Y'_i[\alpha] = Y'_i[0] \oplus Z'_i[\alpha]$ and

- $X'_i[\alpha] = \pi^{-1}(Y'_i[\alpha])$ if $Y'_i[\alpha] \in \mathsf{rng}(\pi)$,
- $X'_i[\alpha] \leftarrow_{\$} \{0,1\}^n \setminus \mathsf{dom}(\pi)$ otherwise, where $\pi(X'_i[\alpha]) = Y'_i[\alpha]$ is added to Π.

Finally, for $(i,\alpha) \in \mathcal{P}'$, set

$$M'_i[\alpha] = \begin{cases} (Y'_i[\alpha] \oplus Y'_i[0]) \oplus_{\mathtt{msb}} Z'_i[\alpha] & \text{if } (i,\alpha) \in \mathcal{P}'_{\$}, \\ X'_i[\alpha] \oplus 2^{\alpha}\Delta'_{i,1} \oplus \Delta'_{i,2} & \text{otherwise.} \end{cases}$$

Step 5. In this step, we sample the π-evaluations needed for tag generation. For each $i \in [q_e]$, set:

$$\begin{aligned} P_i[0] &= 2^{m_i}\Delta_{i,1} \oplus \Delta_{i,3}; \\ P_i[1] &= 2^{m_i}\Delta_{i,1} \oplus 2\Delta_{i,3} \oplus \bigoplus_{\alpha \in [m_i]} \overline{M_i[\alpha]}; \\ Z_{i,*} &= T_i \oplus 3\Delta_{i,3} \oplus \bigoplus_{\alpha \in [a_i]} V_i[\alpha]. \end{aligned}$$

For each $j \in [q_d]$, set:

$$\begin{aligned} P'_i[0] &= 2^{m'_i}\Delta'_{i,1} \oplus \Delta'_{i,3}; \\ P'_i[1] &= 2^{m'_i}\Delta'_{i,1} \oplus 2\Delta'_{i,3} \oplus \bigoplus_{\alpha \in [m'_i]} \overline{M'_i[\alpha]}; \\ Z'_{i,*} &= T'_i \oplus 3\Delta'_{i,3} \oplus \bigoplus_{\alpha \in [a'_i]} V'_i[\alpha]. \end{aligned}$$

Let

$$\begin{aligned} \mathcal{P}^* &\stackrel{\text{def}}{=} \{(i,\alpha) : i \in [q_e], \alpha \in \{0,1\}\}, \\ \mathcal{P}^*_{\mathsf{coll}} &\stackrel{\text{def}}{=} \{(i,\alpha) \in \mathcal{P}^* : P_i[\alpha] \in \mathsf{dom}(\pi) \text{ or } (j,\beta) \in \mathcal{P}^* \setminus \{(i,\alpha)\} \\ &\qquad \text{such that } P_i[\alpha] = P_j[\beta]\}. \end{aligned}$$

We now define bad events badB and badC; let

$$\mathsf{badB} \Leftrightarrow \mathsf{badB}_1 \vee \mathsf{badB}_2 \vee \mathsf{badB}_3 \vee \mathsf{badB}_4 \vee \mathsf{badB}_5,$$

where

- $\mathsf{badB}_1 \Leftrightarrow$ there exists $i \in [q_e]$ such that $(i,0),(i,1) \in \mathcal{P}^*_{\mathsf{coll}}$.

– $\mathsf{badB}_2 \Leftrightarrow$ there exists $i \in [q_e]$ such that $Z_{i,*} = 0$.
– $\mathsf{badB}_3 \Leftrightarrow$ there exists $(i, \alpha) \in \mathcal{P}^*$ such that $P_i[\alpha] \in \mathsf{dom}(\pi)$ and $\pi(P_i[\alpha]) \oplus Z_{i,*} \in \mathsf{rng}(\pi)$.
– $\mathsf{badB}_4 \Leftrightarrow$ there exist three distinct $(i, \alpha), (j, \beta), (k, \gamma) \in \mathcal{P}^*$ such that

$$P_i[\alpha] = P_j[\beta] = P_k[\gamma].$$

– $\mathsf{badB}_5 \Leftrightarrow$ there exist $(i, \alpha), (j, \beta) \in \mathcal{P}^*$ such that $i \neq j$, $P_i[\alpha] = P_j[\beta]$, and $Z_{i,*} = Z_{j,*}$,

and let

$$\mathsf{badC} \Leftrightarrow \mathsf{badC}_1 \vee \mathsf{badC}_2 \vee \mathsf{badC}_3 \vee \mathsf{badC}_4,$$

where

– $\mathsf{badC}_1 \Leftrightarrow$ there exists $i \in [q_d]$ such that $P'_i[0] \in \mathsf{dom}(\pi)$, $P'_i[1] \in \mathsf{dom}(\pi)$ and $\pi(P'_i[0]) \oplus \pi(P'_i[1]) = Z'_{i,*}$.
– $\mathsf{badC}_2 \Leftrightarrow$ there exist $i \in [q_d]$, $\alpha \in \{0, 1\}$, and $(j, \beta) \in \mathcal{P}^*_{\mathsf{coll}}$ such that $P'_i[\alpha] \in \mathsf{dom}(\pi)$, $P'_i[1 - \alpha] = P_j[1 - \beta]$, and $\pi(P'_i[\alpha]) \oplus \pi(P_j[\beta]) = Z'_{i,*} \oplus Z_{j,*}$.
– $\mathsf{badC}_3 \Leftrightarrow$ there exist $i \in [q_d]$, and $(j, \alpha) \in \mathcal{P}^*$ such that $P'_i[0] = P_j[\alpha]$, $P'_i[1] = P_j[1 - \alpha]$, and $Z'_{i,*} = Z_{j,*}$.
– $\mathsf{badC}_4 \Leftrightarrow$ there exist $i \in [q_d]$, $(j, \alpha), (k, \beta) \in \mathcal{P}^*$ such that $j \neq k$, $P'_i[0] = P_j[\alpha]$, $P'_i[1] = P_k[\beta]$, $P_j[1 - \alpha] = P_k[1 - \beta]$, and $Z'_{i,*} = Z_{j,*} \oplus Z_{k,*}$.

Without badB, one can use Mirror theory in the ideal world, while the adversarial forgery is prevented by excluding badC. Assuming $\neg\mathsf{badB} \wedge \neg\mathsf{badC}$, we establish a system of equations in $\pi(P_i[0])$ and $\pi(P_i[1])$ and then sample one solution uniformly at random from the set of all possible solutions. The corresponding π-evaluations, namely $Q_i[\alpha] = \pi(P_i[\alpha])$ and $Q'_j[\beta] = \pi(P'_j[\alpha])$ are includeded in Π for $i \in [q_e]$ and $j \in [q_d]$. Let $\mathcal{L}'$ denote a system of equations and non-equations in Q-variables constructed by the following rules: for each $i \in [q_e]$,

– if $P_i[0] \in \mathsf{dom}(\pi)$, add $\pi(P_i[1]) = \pi(P_i[0]) \oplus Z_{i,*}$ to Π,
– if $P_i[1] \in \mathsf{dom}(\pi)$, add $\pi(P_i[0]) = \pi(P_i[1]) \oplus Z_{i,*}$ to Π,
– otherwise, add an equation $Q_i[0] \oplus Q_i[1] = Z_{i,*}$ to $\mathcal{L}'$,

and for each $i \in [q_d]$,

– if $P'_i[0] \in \mathsf{dom}(\pi)$ and $P'_i[1] \notin \mathsf{dom}(\pi)$, add $Q'_i[1] \neq \pi(P'_i[0]) \oplus Z'_{i,*}$ to $\mathcal{L}'$,
– if $P'_i[1] \in \mathsf{dom}(\pi)$ and $P'_i[0] \notin \mathsf{dom}(\pi)$, add $Q'_i[0] \neq \pi(P'_i[1]) \oplus Z'_{i,*}$ to $\mathcal{L}'$,
– otherwise, add $Q'_i[0] \oplus Q'_i[1] \neq Z'_{i,*}$ to $\mathcal{L}'$.

Once $\mathcal{L}'$ is established, one solution is sampled uniformly at random from the set of solutions to $\mathcal{L}'$ such that none of the values is contained in $\mathsf{rng}(\pi)$. There is at least one such solution assuming $\neg\mathsf{badB} \wedge \neg\mathsf{badC}$. For $i \in [q_e], j \in [q_d], \alpha \in \{0, 1\}$, the following π-evaluatoins are added to Π:

$$\pi(P_i[\alpha]) = Q_i[\alpha],$$
$$\pi(P'_j[\alpha]) = Q'_j[\alpha].$$

Once all the steps are finished without abortion, the following transcript is returned:

$$\tau = \left\{(N_i, A_i, M_i, C_i, T_i)_{i\in[q_e]}, (N'_j, A'_j, C'_j, T'_j, b_j)_{j\in[q_d]}, \Pi\right\}.$$

4.3 Proof of Theorem 1

We are now ready to prove Theorem 1. The transcript τ will be called *bad* if badA, badB, or badC occurs. Let $\mathcal{T}_{\mathsf{bad}}$ be the set of all the bad transcripts. Then the probability that a transcript is bad in the ideal world is upper bounded as follows.

Lemma 3.

$$\begin{aligned}\Pr\left[\mathsf{T}_{\mathrm{id}} \in \mathcal{T}_{\mathsf{bad}}\right] \leq & \frac{25q + 2\sigma + 1.5l(q+\sigma)}{2^n} \\ & + \frac{4q\sigma^2 + (30q^2 + 4q)\sigma + 93q^3 + 44q^2}{2^{2n}} \\ & + \frac{(\sigma^3 + 8\sigma^2 q + 45\sigma q^2 + 6q^3)l}{2^{2n+1}}.\end{aligned}$$

Lemma 3 holds since

$$\Pr\left[\mathsf{T}_{\mathrm{id}} \in \mathcal{T}_{\mathsf{bad}}\right] \leq \Pr\left[\mathsf{badA}\right] + \Pr\left[\mathsf{badB}\right] + \Pr\left[\mathsf{badC}\right]$$

and by the following lemmas.

Lemma 4.

$$\Pr\left[\mathsf{badA}\right] \leq \frac{1.5l(q+\sigma) + 14q}{2^n} + \frac{(\sigma^3 + 8\sigma^2 q + 45\sigma q^2 + 6q^3)l}{2^{2n+1}}.$$

Lemma 5.

$$\Pr\left[\mathsf{badB}\right] \leq \frac{3q + 2\sigma}{2^n} + \frac{73q^3 + 22q^2\sigma + 4q^2}{2^{2n}}.$$

Lemma 6.

$$\Pr\left[\mathsf{badC}\right] \leq \frac{8q}{2^n} + \frac{4q\sigma^2 + 8q^2\sigma + 20q^3 + 4q\sigma + 40q^2}{2^{2n}}.$$

The proof of the above lemmas is given in the full version.

If a transcript is not bad, then such a transcript will be called *good*. The ratio of probabilities of obtaining any good transcript in the ideal and the real worlds is lower bounded as follows.

Lemma 7. *For any transcript $\tau \notin \mathcal{T}_{\mathsf{bad}}$,*

$$\frac{\Pr\left[\mathsf{T}_{\mathrm{re}} = \tau\right]}{\Pr\left[\mathsf{T}_{\mathrm{id}} = \tau\right]} \geq 1 - \frac{4\sigma^3 l + 6\sigma q}{2^{2n}} - \frac{3q_d}{2^n}.$$

Proof. Fix a transcript $\tau \notin \mathcal{T}_{\mathsf{bad}}$. Let $\mathcal{B} = \left\{j \in [q_d] : N'_j \neq N_i \text{ for } \forall i \in [q_e]\right\}$. Let L denote the number of input/output pairs given to the adversary. Since the probability that $\mathcal{D}$ obtains $b_j = \perp$ is exactly $1 - \frac{1}{2^n}$ for each $j \in [q_d]$, we have

$$\Pr\left[\mathsf{T}_{\mathrm{re}} = \tau\right] = \frac{(2^n - L)!}{(2^n)!} \cdot \left(1 - \frac{1}{2^n}\right)^{q_d} \geq \frac{1}{(2^n)_L} \cdot \left(1 - \frac{q_d}{2^n}\right). \tag{1}$$

For the set of π-evaluations obtained in the ideal world $\mathit{\Pi}$, let

$$\begin{aligned}
L_1 &= \{X_i[\alpha] : (i,\alpha) \in \mathcal{P}, i \in [q_e]\} \cup \{N_i \parallel \langle\alpha\rangle : (i,\alpha) \in \mathcal{N}\}, \\
L_2 &= \{U_i[\alpha] : (i,\alpha) \in [q] \times [a_i]\} \setminus L_1, \\
L_3 &= \left\{X'_j[\alpha] : (j,\alpha) \in \mathcal{P}'\right\} \setminus (L_1 \cup L_2), \\
L_4 &= \left(\{P_i[\alpha] : (i,\alpha) \in [q_e] \times \{0,1\}\} \cup \left\{P'_j[\alpha] : (j,\alpha) \in [q_d] \times \{0,1\}\right\}\right) \\
&\quad \setminus (L_1 \cup L_2 \cup L_3).
\end{aligned}$$

Note that $|L_1|$, $|L_2|$, $|L_3|$, $|L_4|$ are the number of π-evaluations determined by step 2, step 3, step 4 and step 5, respectively, and hence $|L_1|+|L_2|+|L_3|+|L_4| = L$. Then, we make the following observation.

1. Since $\mathbf{s}_i$ is sampled for each partial block $(i, m_i) \in \mathcal{P}_\$$, the probability that $\mathcal{D}$ obtains $(C_i[m_i], \mathbf{s}_i)$ is exactly $\frac{1}{2^n}$ for $(i,\alpha) \in \mathcal{P}_\$$. Since ciphertexts and tags are chosen uniformly and independently at random, the probability of $\mathcal{D}$ obtaining them is at most

$$\frac{1}{(2^n)^{\sigma_e}(2^n)^{q_e}}$$

   where $\sigma_e = \sum_{i \in [q_e]} m_i$.
2. At step 1, $\mathit{\Delta}_{i,1}$, $\mathit{\Delta}_{i,2}$, $\mathit{\Delta}_{i,3}$ are sampled uniformly and independently at random from $\{0,1\}^n$ for each $i \in [q_e]$. Also, $\mathit{\Delta}'_{j,1}$, $\mathit{\Delta}'_{j,2}$, $\mathit{\Delta}'_{j,3}$ are sampled in the same way. Therefore, the probability that $\mathcal{D}$ obtains the masking values (in the transcript) is given as

$$\frac{1}{(2^n)^{3(q_e+|\mathcal{B}|)}}.$$

3. At step 2, we determine the π-evaluations used in the mask generations and message encryptions. For $i \in [q_e]$, let

$$\begin{aligned}
\mathcal{V}_{i,\mathsf{X}} &= \{\pi(X_i[\alpha]) : (i,\alpha) \in \mathcal{P}\}, \\
\mathcal{E}_{i,\mathsf{X}} &= \{(\pi(X_i[0]), \pi(X_i[\alpha])) : \alpha \in [m_i]\}, \\
\mathcal{G}_{i,\mathsf{X}} &= (\mathcal{V}_{i,\mathsf{X}}, \mathcal{E}_{i,\mathsf{X}})
\end{aligned}$$

   where $(\pi(X_i[0]), \pi(X_i[\alpha])) \in \mathcal{E}_{i,\mathsf{X}}$ has label $(Z_i[\alpha], =)$. For $i \in [q]$, let

$$\begin{aligned}
\mathcal{V}_{i,\mathsf{N}} &= \{\pi(N_i[\alpha]) : \alpha \in \{0,1,2,3\}\}, \\
\mathcal{E}_{i,\mathsf{N}} &= \{(\pi(N_{i,0}), \pi(N_{i,\alpha})) : \alpha \in [3]\} \\
\mathcal{G}_{i,\mathsf{N}} &= (\mathcal{V}_{i,\mathsf{N}}, \mathcal{E}_{i,\mathsf{N}})
\end{aligned}$$

where $(\pi(N_{i,0}), \pi(N_{i,\alpha})) \in \mathcal{E}_{i,\mathsf{N}}$ has label $(\Delta_{i,\alpha}, =)$. Note that $\mathcal{G}_{i,\mathsf{X}}$ for $i \in [q_e]$ and $\mathcal{G}_{i,\mathsf{N}}$ for $i \in [q]$ are all connected graphs, and we will call these graphs by 'segments'. Now $\mathcal{G}$ be the union of all segments, i.e.,

$$\mathcal{G} = \left(\bigcup_{i \in [q_e]} \mathcal{G}_{i,\mathsf{X}} \right) \cup \left(\bigcup_{i \in [q]} \mathcal{G}_{i,\mathsf{N}} \right).$$

Then, $\mathcal{G}$ has the following properties.

- No more than two segments are included in a single connected component of $\mathcal{G}$. Otherwise, either there exist three segments meeting in one vertex, which implies badA_3, or three segments meeting in two different vertices, which implies badA_2.
- $\mathcal{G}$ does not have any cycle. If there exists a cycle in a single segment, then it implies badA_1, and if there exists a cycle contained in two (connected) segments, there should be at least two different collisions, which implies badA_2.
- Let u denote the number of components in $\mathcal{G}^{=}$, and let $\mathcal{C}_1, \ldots, \mathcal{C}_u$ be the components of $\mathcal{G}^{=}$. Then obviously $\sum_{i=1}^{u} |\mathcal{C}_i| = |L_1|$ and $|\mathcal{C}_i| \leq 4l$ for each $i = 1, \ldots, u$. Therefore we have

$$\sum_{i=1}^{u} |\mathcal{C}_i|^2 \leq 4l\,|L_1|\,. \tag{2}$$

- $\lambda(\mathcal{L}) \neq 0$ for any trail $\mathcal{L}$ in $\mathcal{G}(= \mathcal{G}^{=})$ since otherwise such a trail will be included in a single segment or both the endpoints of the trails are included in the two different segments respectively. The former case implies badA_4, and the latter case implies badA_5. Recall that any three segments are not included in a single component.

By Lemma 2, we can lower bound the number of the possible assignments such that the evaluations sampled in step 2 are the same as the corresponding part of the transcript. Let $h(\mathcal{G})$ denote the possible assignments of distinct values to the vertices of $\mathcal{G}$. In step 2, one of the possible $h(\mathcal{G}')$ assignments is chosen uniformly at random. Note that $|\mathcal{E}^{=}| = \sigma_e + 3q_e + 3\,|\mathcal{B}|$. By Lemma 2 and (2),

$$\begin{aligned} h(\mathcal{G}) &\geq \frac{(N)_{|L_1|}}{N^{\sigma_e + 3q_e + 3|\mathcal{B}|}} \times \left(1 - \frac{|L_1|^2}{N^2} \sum_{i=1}^{u} |\mathcal{C}_i|^2\right) \\ &\geq \frac{(N)_{|L_1|}}{N^{\sigma_e + 3q_e + 3|\mathcal{B}|}} \times \left(1 - \frac{4l\,|L_1|^3}{N^2}\right). \end{aligned}$$

4. At step 3, the oracle samples $V_i[\alpha]$'s in the encryption queries and $V'_j[\beta]$'s in the decryption queries from $\{0,1\}^n$ excluding $|L_1|$ numbers of the evaluations determined in step 2. Therefore, the probability that $\mathcal{D}$ obtains $V_i[\alpha]$'s (in the transcript) is $\frac{1}{(2^n - |L_1|)_{|L_2|}}$.

5. At step 4, the oracle samples the primitive calls for the message blocks in the decryption queries. For $i \in [q_d]$, let $X'_i[0] = \Delta'_i[0]$. The oracle samples $Y'_i[0] = \pi(X'_i[0])$ from $\{0,1\}^n$ excluding $|L_1| + |L_2|$ numbers of evaluations determined in step 2 and step 3. Then $Y'_i[\alpha]$'s are determined by $Z'_i[\alpha] \oplus Y'_i[0]$. After that, $X'_i[\alpha]$'s are sampled uniformly at random from $\{0,1\}^n$. Therefore, the probability that $\mathcal{D}$ obtains $Y'_i[0]$'s and $X'_i[\alpha]$'s (in the transcript) is $\frac{1}{(2^n - |L_1| - |L_2|)_{|L_3|}}$.
6. At step 5, we determine the π-evaluations used to generate tags. Note that there is no successful forgery assuming $\neg\mathsf{badC}$. Let

$$\begin{aligned}
\mathcal{W} &= \{\pi(P_i[\alpha]) : (i,\alpha) \in \mathcal{P}_*, P_i[\alpha] \in \mathsf{dom}(\pi)\} \\
&\quad \cup \left\{\pi(P'_j[\beta]) : (j,\beta) \in [q_d] \times \{0,1\}, P'_j[\beta] \in \mathsf{dom}(\pi)\right\}, \\
\mathcal{V}'_e &= \bigcup_{i \in [q_e]} \{\pi(P_i[0]), \pi(P_i[1])\} \setminus \mathcal{W}, \\
\mathcal{V}'_d &= \bigcup_{i \in [q_d]} \{\pi(P'_i[0]), \pi(P'_i[1])\} \setminus \mathcal{W}, \\
\mathcal{V}' &= \mathcal{V}'_e \cup \mathcal{V}'_d,
\end{aligned}$$

where the elements of $\mathcal{V}'$ are unknown. Define a graph $\mathcal{G}' = \left(\mathcal{V}' \sqcup \mathcal{W}, \mathcal{E}'^{=} \sqcup \mathcal{E}'^{\neq}\right)$, where

$$\begin{aligned}
\mathcal{E}'^{=} &= \{(\pi(P_i[0]), \pi(P_i[1])) : i \in [q_e]\}, \\
\mathcal{E}'^{\neq} &= \{(\pi(P'_i[0]), \pi(P'_i[1])) : i \in [q_d]\},
\end{aligned}$$

$(P_i[0], P_i[1]) \in \mathcal{E}'^{=}$ has label $(Z_{i,*}, =)$, and $(P'_i[0], P'_i[1]) \in \mathcal{E}'^{\neq}$ has label $(Z'_{i,*}, \neq)$. The graph $\mathcal{G}'$ has the following properties.

- $\mathcal{G}'$ contains no cycle since otherwise there should be two indices $(i,0)$ and $(i,1)$ that are contained in $\mathcal{P}^*_{\mathsf{coll}}$, which implies badB_1.
- No more than two edges are included in one component. Otherwise, either three edges should meet in one vertex which implies badB_4 or we have
$$(i,0), (i,1) \in \mathcal{P}^*_{\mathsf{coll}},$$
which implies badB_1.
- Let u' denote the number of components in $\mathcal{G}'^{=}$, and let $\mathcal{C}'_1, \mathcal{C}'_2, \ldots, \mathcal{C}'_{u'}$ be the components of $\mathcal{G}'^{=}$. Then $\sum_{i=1}^{u'} |\mathcal{C}'_i| = |L_4|$ and $|\mathcal{C}'_i| \leq 3$ for each $i = 1, \ldots, u'$. Therefore we have
$$\sum_{i=1}^{u'} |\mathcal{C}'_i|^2 \leq 3\,|L_4|\,. \tag{3}$$
- For any trail $\mathcal{L}$ in $\mathcal{G}'^{=} = (\mathcal{V}' \sqcup \mathcal{W}, \mathcal{E}'^{=})$, $\lambda(\mathcal{L}) \neq 0$, since otherwise such a trail $\mathcal{L}$ is a single zero-labeled edge or both endpoints of the trail are included in the two different edges respectively. The former case implies badB_2, and the latter case implies badB_5. Recall that any three edges cannot be included in a single component.

Similarly to the analysis for step 2, we use Lemma 2 to lower bound the number of possible assignments such that the evaluations sampled in step 5 are the same as the corresponding part of the transcript. Let $h(\mathcal{G}')$ denote the possible assignments of distinct values to the vertices of $\mathcal{G}'$. In step 5, one of the possible $h(\mathcal{G}')$ assignments is chosen uniformly at random. Note that $|\mathcal{E}^{=}| \le q_e$ and $\left|\mathcal{E}^{\neq}\right| \le q_d$. By Lemma 2 and (3), we have

$$\begin{aligned} h(\mathcal{G}') &\ge \frac{(N-|L_1|-|L_2|-|L_3|)_{|L_4|}}{N^{q_e}} \left(1 - \frac{L}{N^2}\sum_{i=1}^{k}|\mathcal{C}'_i|^2 - \frac{2q_d}{N}\right) \\ &\ge \frac{(N-|L_1|-|L_2|-|L_3|)_{|L_4|}}{N^{q_e}} \left(1 - \frac{3L\,|L_4|}{N^2} - \frac{2q_d}{N}\right). \end{aligned}$$

By the above argument, we have

$$\begin{aligned} \frac{1}{\Pr\left[\mathsf{T}_{\mathrm{id}} = \tau\right]} &= (2^n)^{\sigma_e} \cdot (2^n)^{q_e} \cdot (2^n)^{3(q_e+|\mathcal{B}|)} \cdot h(\mathcal{G}) \cdot (2^n - |L_1|)_{|L_2|} \\ &\quad \times (2^n - |L_1| - |L_2|)_{|L_3|} \cdot h(\mathcal{G}') \\ &\ge (2^n)^{\sigma_e+q_e} \cdot (2^n)^{3(q_e+|\mathcal{B}|)} \cdot (2^n - |L_1|)_{|L_2|+|L_3|} \\ &\quad \times \frac{(2^n)_{|L_1|}}{(2^n)^{\sigma_e+3q_e+3|\mathcal{B}|}} \cdot \left(1 - \frac{4l\,|L_1|^3}{2^{2n}}\right) \\ &\quad \times \frac{(N-|L_1|-|L_2|-|L_3|)_{|L_4|}}{2^{nq_e}} \cdot \left(1 - \frac{3L\,|L_4|}{2^{2n}} - \frac{2q_d}{2^n}\right) \\ &\ge \frac{(2^n)^{3(q_e+|\mathcal{B}|)}}{(2^n)^{3(q_e+|\mathcal{B}|)}} \cdot (2^n)_L \cdot \left(1 - \frac{4l\,|L_1|^3 + 3L\,|L_4|}{2^{2n}} - \frac{2q_d}{2^n}\right) \\ &\ge (2^n)_L \cdot \left(1 - \frac{4l\,|L_1|^3 + 3L\,|L_4|}{2^{2n}} - \frac{2q_d}{2^n}\right). \end{aligned} \tag{4}$$

Therefore by (1) and (4), we have

$$\begin{aligned} \frac{\Pr\left[\mathsf{T}_{\mathrm{re}} = \tau\right]}{\Pr\left[\mathsf{T}_{\mathrm{id}} = \tau\right]} &\ge \left(1 - \frac{4l\,|L_1|^3 + 3L\,|L_4|}{2^{2n}} - \frac{2q_d}{2^n}\right) \cdot \left(1 - \frac{q_d}{2^n}\right) \\ &\ge 1 - \frac{4\,|L_1|^3\, l + 3L\,|L_4|}{2^{2n}} - \frac{3q_d}{2^n} \\ &\ge 1 - \frac{4\sigma^3 l + 6\sigma q}{2^{2n}} - \frac{3q_d}{2^n} \end{aligned}$$

where the last inequality holds since $L < \sigma$ and $|L_4| \le 2q$. □

5 On the tightness of the bound of XOCB

We show a brief analysis of the tightness of the bound in Theorem 1 by presenting an authentication attack against XOCB. The attack tries to invoke the event

corresponding to badA_1. For a positive integer $s \geq 2$, the attack requires $l \approx 2^{n/s}$, $q_e \approx 2^{(s-2)n/s}$, and $\sigma_e = lq_e \approx 2^{(s-1)n/s}$. This is not tight for our claim of $2n/3$-bit security with $l = O(1)$. However, if l is not constant, especially when $s = 2$, the attack complexity is $l \approx 2^{n/2}$, $q_e = O(1)$, and $\sigma_e \approx 2^{n/2}$; thus, it is a tight attack. When $O(1) < l < 2^{n/2}$, the attack is not tight for our claim in Theorem. 1. For example, if $s = 3$, the attack complexity is $l \approx 2^{n/3}$, $q_e \approx 2^{n/3}$, and $\sigma_e \approx 2^{2n/3}$. The gap from Theorem 1 increases as s increases.

The attack procedure is as follows:

1. The adversary queries (N, A, M) to the encryption oracle such that $M = M[1]\|M[2]\|\cdots\|M[m]$ and $|M[m]| = n-1$. Then it obtains (C, T), where $C = C[1] \,\|\, C[2] \,\|\, \cdots \,\|\, C[m]$, and also obtains $(n-1)$-bit value $Z = M[m] \oplus C[m]$.
2. Assume that a collision $M[i] \oplus 2^i\Delta_1 \oplus \Delta_2 = M[j] \oplus 2^j\Delta_1 \oplus \Delta_2$ occurs for $i, j \in [m-1]$ and $i \neq j$. Then, $M[i] \oplus M[j] = C[i] \oplus C[j]$ holds; thus, the adversary can detect the collision.
3. The adversary compute $\Delta_1 = (2^i \oplus 2^j)^{-1}(M[i] \oplus M[j])$.
4. The adversary queries (N', A', C', T') to the decryption oracle such that $N' = N$, $A' = A$, $T' = T$, $C' = C'[1] \,\|\, C'[2] \,\|\, \cdots \,\|\, C'[m]$, $C'[1] = 2\Delta_1$, $C'[2] = 2^2\Delta_1$, $C'[i] = C[i]$ for $i \in [3..m-1]$, $|C'[m]| = n-1$, and $C'[m] = Z \oplus \mathtt{msb}_{n-1}(2\Delta_1 \oplus 2^2\Delta_1 \oplus M[1] \oplus M[2]) \oplus M[m]$.

The last decryption query is accepted with a high probability. For $i \in [m]$, let $M'[i]$ and Σ' be a valid i-th decrypted plaintext block and a valid checksum of the last decryption query (N', A', C', T'), respectively.

$$\begin{aligned}\Sigma' &= \bigoplus_{i\in[m]} \mathtt{ozp}(M'[i]) = M'[1] \oplus M'[2] \oplus \mathtt{ozp}(M'[m]) \oplus \bigoplus_{i\in[3..m-1]} M'[i] \\ &= E_K^{-1}(\Delta_2 \oplus L) \oplus 2\Delta_1 \oplus \Delta_2 \oplus E_K^{-1}(\Delta_2 \oplus L) \oplus 2^2\Delta_1 \oplus \Delta_2 \\ &\quad \oplus \mathtt{ozp}(\mathtt{msb}_{n-1}(2\Delta_1 \oplus 2^2\Delta_1 \oplus M[1] \oplus M[2]) \oplus M[m]) \oplus \bigoplus_{i\in[3..m-1]} M[i]\end{aligned}$$

If the adversary has

$$\begin{aligned}&\mathtt{ozp}(\mathtt{msb}_{n-1}(2\Delta_1 \oplus 2^2\Delta_1 \oplus M[1] \oplus M[2]) \oplus M[m]) \\ &\quad = 2\Delta_1 \oplus 2^2\Delta_1 \oplus M[1] \oplus M[2] \oplus \mathtt{ozp}(M[m]),\end{aligned} \tag{5}$$

it obtains $\Sigma' = \bigoplus_{i=1}^{m} \mathtt{ozp}(M[i]) = \Sigma$, and T becomes the valid tag for (N', A', C'). The adversary can check whether (5) holds before the last decryption query; thus, if (5) does not hold, the adversary can make a successful forgery by changing C' accordingly, for example, setting $C'[2] = 2^2\Delta_1$, $C'[3] = 2^3\Delta_1$, $C'[i] = C[i]$ for $i \in \{1\} \cup \{4, \ldots, m-1\}$, and $C'[m] = Z \oplus \mathtt{msb}_{n-1}(2^2\Delta_1 \oplus 2^3\Delta_1 \oplus M[2] \oplus M[3]) \oplus M[m]$, or changing the length of $C'[m]$ to smaller bits.

Next, we discuss the attack complexity. In step 2, the adversary requires the collision $M[i] \oplus 2^i\Delta_1 \oplus \Delta_2 = M[j] \oplus 2^j\Delta_1 \oplus \Delta_2$ for $i, j \in [m-1]$ and $i \neq j$. To obtain this collision with a high probability, the adversary needs to query a sufficiently long plaintext M in step 1. Assuming that $m \approx 2^{n/s}$ for a

positive integer s, the collision probability is approximately $m^2/2^n \approx 2^{(2-s)n/s}$. Repeating step 1 with $m \approx 2^{n/s}$ $q_e \approx 2^{(s-2)n/s}$ times, the adversary obtains the collision with a high probability. Thus, the attack requires $l \approx 2^{n/s}$, $q_e \approx 2^{(s-2)n/s}$, and $\sigma_e = lq_e \approx 2^{(s-1)n/s}$ when l is not a constant. If $l = O(1)$, the collision probability of step 2 is $\approx 1/2^n$ and the attack complexity is $q_e \approx \sigma_e \approx 2^n$, much larger than what the bound tells ($2^{2n/3}$). Further analysis is open.

6 Implementations of XOCB

This section presents the implementations for the instantiation of XOCB using AES – AES-XOCB [14].

On 64-Bit High-End Processors. Using the parallelizability of XOCB, our implementation of AES-XOCB can take advantage of the pipelined execution of AES-NI on high-end CPUs, resulting in an asymptotic speed of 0.5 cpb. This performance is as expected since the fully pipelined AES-ECB runs at 0.3 cpb and doubling in GF(2^{128}) runs at 0.2 cpb using SIMD instructions in our timing environment.

We compared the relative performance of AES-XOCB against AES-OCB and AES-CIP using the same AES-NI-based AES implementation, SIMD-based doubling in GF(2^{128}), and PCLMULQDQ-based multiplication in GF(2^{128}) supporting pipelined execution on multiple blocks. Our testing included the time cost of the entire procedure, including setting up keys, generating masks, and performing encryption and authentication. We used plaintexts of various lengths for testing, ranging from 16 to 4096 bytes (with a 16-byte AD).

Comparing the results, AES-XOCB has a slightly inferior performance compared to AES-OCB but is still close. AES-XOCB's initialization procedure uses five AES calls for computing mask initial values, which slightly impacts performance for short messages. However, for message lengths exceeding 512 bytes, the difference narrows to 0.1∼0.2 cpb, which is the cost of a doubling. AES-XOCB outperforms AES-CIP for both short and long messages. Figure 6 shows how the performance of AES-XOCB changes with plaintext length, and how it compares to AES-OCB and AES-CIP.

On 8-Bit Low-End Microprocessors. We demonstrate the practical relevance of XOCB in constrained environments by implementing AES-XOCB on an 8-bit AVR. The simulation result on ATmega328P shows that AES-XOCB requires 8556 bytes of ROM and 672 bytes of RAM to support both encryption and decryption, including key setup and mask generation. Figure 7 shows concrete execution time for the entire procedure, including key setup, mask generation, encryption, and authentication. For a 128-byte message and a 16-byte AD, AES-XOCB processes at 306 cpb, while an optimized AES-GCM implementation requires 11012 bytes of ROM and runs at 880 cpb [40].

[14] The source codes can be found via https://www.dropbox.com/sh/k0y8h1boah072mn/AAAYPUr0j4MU9F3-w1k7U52Ha?dl=0

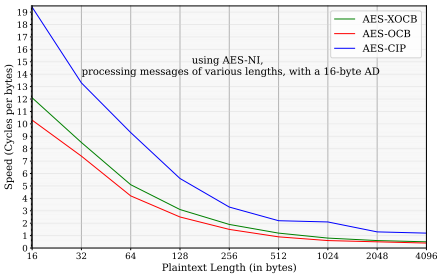

Fig. 6: Speeds on an x86-64 CPU

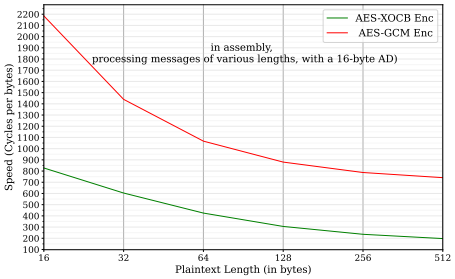

Fig. 7: Speeds on an 8-bit AVR

7 Conclusions

We have shown a new authenticated encryption mode XOCB. It has a quantitatively stronger security guarantee than the seminal OCB while inheriting most of the efficiency advantages. In particular, it is exactly rate-one and has beyond-birthday-bound security assuming SPRP for the underlying block cipher, if the maximum input length is sufficiently smaller than the birthday bound. The block cipher could be instantiated with an n-bit block cipher with a key of any length, allowing us to use AES-128 for a typical example. There are numerous works on BBB-secure AE modes, however, they rely on a stronger primitive (e.g. TBC) or stronger assumption (e.g. ideal cipher model), and XOCB is the first scheme that achieves the aforementioned goals without such a compromise. Several further research topics, such as optimizing the scheme to reduce computational overhead or reducing the length contribution to the bound, and a more comprehensive benchmark, would be interesting directions.

Acknowledgements

We thank the anonymous reviewers for their insightful comments that improved the presentation of our paper. Jooyoung Lee was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2021R1F1A1047146). Zhenzhen Bao was supported by the National Key R&D Program of China (Grant No. 2018YFA0704701), the Major Program of Guangdong Basic and Applied Research (Grant No. 2019B030302008), and the Shandong Province Key R&D Project (Nos. 2020ZLYS09 and 2019JZZY010133).

References

[1] Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D (2007), National Institute of Standards and Technology.

[2] Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: How to securely release unverified plaintext in authenticated encryption. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 105–125. Springer, Heidelberg (Dec 2014). https://doi.org/10.1007/978-3-662-45611-8_6

[3] Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 411–436. Springer, Heidelberg (Nov / Dec 2015). https://doi.org/10.1007/978-3-662-48800-3_17

[4] Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 321–345. Springer, Heidelberg (Sep 2017). https://doi.org/10.1007/978-3-319-66787-4_16

[5] Beierle, C., Biryukov, A., dos Santos, L.C., Großschädl, J., Perrin, L., Udovenko, A., Velichkov, V., Wang, Q.: Alzette: A 64-bit ARX-box - (feat. CRAX and TRAX). In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 419–448. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56877-1_15

[6] Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (Dec 2000). https://doi.org/10.1007/3-540-44448-3_41

[7] Bhargavan, K., Leurent, G.: On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and OpenVPN. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016. pp. 456–467. ACM Press (Oct 2016). https://doi.org/10.1145/2976749.2978423

[8] Bhattacharjee, A., Bhaumik, R., Nandi, M.: Offset-based bbb-secure tweakable block-ciphers with updatable caches. In: INDOCRYPT. Lecture Notes in Computer Science, vol. 13774, pp. 171–194. Springer (2022)

[9] Bhaumik, R., Nandi, M.: Improved security for OCB3. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 638–666. Springer, Heidelberg (Dec 2017). https://doi.org/10.1007/978-3-319-70697-9_22

[10] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (Sep 2007). https://doi.org/10.1007/978-3-540-74735-2_31

[11] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knežević, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (Dec 2012). https://doi.org/10.1007/978-3-642-34961-4_14

[12] Choi, W., Lee, B., Lee, Y., Lee, J.: Improved security analysis for nonce-based enhanced hash-then-mask macs. In: Moriai, S., Wang, H. (eds.) Advances in Cryptol-

ogy – ASIACRYPT 2020. pp. 697–723. Springer International Publishing, Cham (2020)

[13] Cogliati, B., Dutta, A., Nandi, M., Patarin, J., Saha, A.: Proof of Mirror Theory for any xi_max. Cryptology ePrint Archive, Paper 2022/686 (2022), https://eprint.iacr.org/2022/686, https://eprint.iacr.org/2022/686

[14] Cogliati, B., Patarin, J.: Mirror theory: A simple proof of the pi+pj theorem with xi_max=2. Cryptology ePrint Archive, Paper 2020/734 (2020), https://eprint.iacr.org/2020/734, https://eprint.iacr.org/2020/734

[15] Datta, N., Dutta, A., Nandi, M., Paul, G.: Double-block hash-then-sum: A paradigm for constructing BBB secure PRF. IACR Trans. Symm. Cryptol. **2018**(3), 36–92 (2018). https://doi.org/10.13154/tosc.v2018.i3.36-92

[16] Datta, N., Dutta, A., Nandi, M., Yasuda, K.: Encrypt or Decrypt? To Make a Single-Key Beyond Birthday Secure Nonce-Based MAC. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 (Proceedings, Part I). LNCS, vol. 10991, pp. 631–661. Springer (2018). https://doi.org/10.1007/978-3-319-96884-1_21

[17] Dutta, A., Nandi, M., Saha, A.: Proof of mirror theory for xi_max=2. IEEE Transactions on Information Theory **68**(9), 6218–6232 (2022). https://doi.org/10.1109/TIT.2022.3171178

[18] Dutta, A., Nandi, M., Talnikar, S.: Beyond Birthday Bound Secure MAC in Faulty Nonce Model. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2019 (Proceedings, Part I). LNCS, vol. 11476, pp. 437–466. Springer (2019). https://doi.org/10.1007/978-3-030-17653-2_15

[19] Hoang, V.T., Tessaro, S.: Key-Alternating Ciphers and Key-Length Extension: Exact Bounds and Multi-user Security. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 (Proceedings, Part I). LNCS, vol. 9814, pp. 3–32. Springer (2016). https://doi.org/10.1007/978-3-662-53018-4_1

[20] Inoue, A., Iwata, T., Minematsu, K., Poettering, B.: Cryptanalysis of OCB2: Attacks on authenticity and confidentiality. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 3–31. Springer, Heidelberg (Aug 2019). https://doi.org/10.1007/978-3-030-26948-7_1

[21] Iwata, T.: New blockcipher modes of operation with beyond the birthday bound security. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 310–327. Springer, Heidelberg (Mar 2006). https://doi.org/10.1007/11799313_20

[22] Iwata, T.: Authenticated encryption mode for beyond the birthday bound security. In: Vaudenay, S. (ed.) AFRICACRYPT 08. LNCS, vol. 5023, pp. 125–142. Springer, Heidelberg (Jun 2008)

[23] Iwata, T., Ohashi, K., Minematsu, K.: Breaking and repairing GCM security proofs. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 31–49. Springer, Heidelberg (Aug 2012). https://doi.org/10.1007/978-3-642-32009-5_3

[24] Jha, A., List, E., Minematsu, K., Mishra, S., Nandi, M.: XHX - A framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. In: Lange, T., Dunkelman, O. (eds.) LATINCRYPT 2017. LNCS, vol. 11368, pp. 207–227. Springer, Heidelberg (Sep 2017). https://doi.org/10.1007/978-3-030-25283-0_12

[25] Katz, J., Yung, M.: Unforgeable encryption and chosen ciphertext secure modes of operation. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 284–299. Springer, Heidelberg (Apr 2001). https://doi.org/10.1007/3-540-44706-7_20

[26] Kim, S., Lee, B., Lee, J.: Tight security bounds for double-block hash-then-sum MACs. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part I. LNCS, vol. 12105, pp. 435–465. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45721-1_16
[27] Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer, Heidelberg (Feb 2011). https://doi.org/10.1007/978-3-642-21702-9_18
[28] Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. Journal of Cryptology **24**(3), 588–613 (Jul 2011). https://doi.org/10.1007/s00145-010-9073-y
[29] Mennink, B.: Optimally secure tweakable blockciphers. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 428–448. Springer, Heidelberg (Mar 2015). https://doi.org/10.1007/978-3-662-48116-5_21
[30] Mennink, B.: Insuperability of the standard versus ideal model gap for tweakable blockcipher security. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 708–732. Springer, Heidelberg (Aug 2017). https://doi.org/10.1007/978-3-319-63715-0_24
[31] Naito, Y.: Improved XKX-based AEAD scheme: Removing the birthday terms. In: Lange, T., Dunkelman, O. (eds.) LATINCRYPT 2017. LNCS, vol. 11368, pp. 228–246. Springer, Heidelberg (Sep 2017). https://doi.org/10.1007/978-3-030-25283-0_13
[32] Naito, Y.: Tweakable blockciphers for efficient authenticated encryptions with beyond the birthday-bound security. IACR Trans. Symm. Cryptol. **2017**(2), 1–26 (2017). https://doi.org/10.13154/tosc.v2017.i2.1-26
[33] Niwa, Y., Ohashi, K., Minematsu, K., Iwata, T.: GCM security bounds reconsidered. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 385–407. Springer, Heidelberg (Mar 2015). https://doi.org/10.1007/978-3-662-48116-5_19
[34] Patarin, J.: Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. IACR Cryptology ePrint Archive, Report 2010/287 (2010), available at http://eprint.iacr.org/2010/287
[35] Patarin, J.: Mirror Theory and Cryptography. IACR Cryptology ePrint Archive, Report 2016/702 (2016), available at http://eprint.iacr.org/2016/702
[36] Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM CCS 2002. pp. 98–107. ACM Press (Nov 2002). https://doi.org/10.1145/586110.586125
[37] Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer, Heidelberg (Dec 2004). https://doi.org/10.1007/978-3-540-30539-2_2
[38] Rogaway, P., Bellare, M., Black, J., Krovetz, T.: OCB: A block-cipher mode of operation for efficient authenticated encryption. In: Reiter, M.K., Samarati, P. (eds.) ACM CCS 2001. pp. 196–205. ACM Press (Nov 2001). https://doi.org/10.1145/501983.502011
[39] Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_23
[40] Sovyn, Y., Khoma, V., Podpora, M.: Comparison of Three CPU-Core Families for IoT Applications in Terms of Security and Performance of AES-GCM. IEEE Internet of Things Journal **7**(1), 339–348 (2020). https://doi.org/10.1109/JIOT.2019.2953230