

Revisiting Full-PRF-Secure PMAC and Using It for Beyond-Birthday Authenticated Encryption

Eik List¹(✉) and Mridul Nandi²

¹ Bauhaus-Universität Weimar, Weimar, Germany
`eik.list@uni-weimar.de`

² Applied Statistics Unit, Indian Statistical Institute, Kolkata, India
`mridul.nandi@gmail.com`

Abstract. This paper proposes an authenticated encryption scheme, called SIVx, that preserves BBB security also in the case of unlimited nonce reuses. For this purpose, we propose a single-key BBB-secure message authentication code with $2n$ -bit outputs, called PMAC2x, based on a tweakable block cipher. PMAC2x is motivated by PMAC_TBC1k by Naito; we revisit its security proof and point out an invalid assumption. As a remedy, we provide an alternative proof for our construction, and derive a corrected bound for PMAC_TBC1k.

Keywords: Symmetric cryptography · Message authentication codes · Authenticated encryption · Provable security

1 Introduction

Nonce-Based Authenticated Encryption. Authenticated encryption (AE) schemes aim at protecting both the privacy and the integrity of submitted messages. Authenticated encryption schemes that allow to authenticate not only the encrypted message, but also associated data, are commonly known as AEAD schemes [22]. The common security notions for AE schemes concern the prevention of any leakage about the encrypted messages except for their lengths. Since stateless schemes would enable adversaries to detect a duplicate encryption of the same associated data and message under the current key, Rogaway proposed nonce-based encryption [24], where the user provides an additional nonce for every message she wants to process. In theory, the concept of nonces is simple. However, the practice has shown numerous examples of implementation failures, and settings that render it difficult to almost impossible to prevent nonce reuse (cf. [8]). Before the CAESAR competition, the majority of widely used AE schemes protected neither the confidentiality nor the integrity of messages in the case of nonce repetitions. As a consequence, a considerable number of CAESAR candidates aimed a certain level of security if nonces repeat (e.g., [1, 9, 10, 15]).

Parallelizable MACs in Authenticated Encryption. Block-cipher-based message authentication codes (MACs) are important components not only for

authentication, but also as part of AE schemes, where they are used to derive an initialization vector (IV) that is then used for encryption. In particular, parallelizable MACs like PMAC [6] allow to process multiple blocks in parallel, which is beneficial for software performance on current x64 processors. Since PMAC has several further desirable properties, e. g. being online and incremental, it is not a surprise that all the CAESAR candidates cited above essentially combine a variant of PMAC (or its underlying hash function) with a block-cipher-based mode of operation for efficiently processing associated data and message.

Beyond-Birthday-Bound AE. Besides performance, the quantitative security guarantees are important aspects for AE schemes. The privacy and authenticity guarantees of the AE schemes cited above are limited by the birthday bound of $O(\ell^2/2^n)$, where n denotes the state size of the underlying primitive, and ℓ the number of blocks processed over all queries. Since the schemes above possess an n -bit state, a state collision that leads to attacks has significant probability after approximately $2^{n/2}$ blocks have been processed under the same key.

To address this issue, Peyrin and Seurin presented Synthetic Counter in Tweak (SCT) [20], a beyond-birthday-bound (BBB) AE scheme based on a tweakable block cipher under a single key. Internally, SCT is a MAC-then-Encrypt composition: the MAC part is a PMAC-like construction, called EPWC. The encryption part is Counter-in-Tweak (CTR_T), an efficient mode of operation that takes an n -bit nonce and an n -bit IV. Both EPWC and CTR_T guarantee BBB security as long as nonces never repeat. However, the security of the nonce-IV-based CTR_T degrades to the birthday bound with an increasing number of nonce reuses; even worse, the security of EPWC (and consequently that of SCT) immediately reduces to the birthday bound if a single nonce repeats once. In [21, p. 7], the extended version of [20], the authors remarked therefore (among others) the following open problem: “[...] *to construct an AE scheme which remains BBB-secure even when nonces are arbitrarily repeated. The main difficulty is to build a deterministic, stateless, BBB-secure MAC, which is known to be notably hard*”. Intuitively, an efficient block-cipher-based BBB-secure MAC with $2n$ bit output length could allow to construct such a deterministic AE scheme.¹ Thus, this work will put a large focus on the construction of BBB-secure MACs.

Previous Work. Naito [17] proposed two MACs with full PRF security based on a tweakable block cipher: PMAC_TBC3K and PMAC_TBC1K. While the former requires three keys, the latter uses tweak-based domain separation to require only a single key. Extending the latter seemed a well-suited starting point for our work since such a MAC could be combined in straight-forward manner with a BBB-secure mode of encryption. Though, during our studies, we found that the analysis in [17] assumed internal values to be independent, which—as we will show—cannot always be guaranteed. Since the proof depended largely

¹ We stress that BBB-secure AE is not new if one considers schemes with multiple primitives and keys. For the sake of space limitations, a discussion can be found in the full version of this work [11].

Table 1. Previous parallel BBB-secure MACs. (T)BC = (tweakable) block cipher, $q = \max. \#queries$, $m = \max. \#blocks \text{ per query}$, $\ell = \max. \text{total } \#blocks$.

Primitive	Construction	Keys	Output size	Advantage	Reference
BC	PMAC ⁺	3	n	$O(q^3 m^3 / 2^{2n} + qm / 2^n)$	[25]
	1k_PMAC ⁺	1	n	$O(qm^2 / 2^n + q^3 m^4 / 2^{2n})$	[7]
TBC	PMAC_TBC3k	3	n	$O(q^2 / 2^{2n})$	[17]
	PMAC_TBC1k	1	n	$O(q / 2^n + q^2 / 2^{2n})$	[17]
	PMACx	1	n	$O(q^2 / 2^{2n} + q^3 / 2^{3n})$	Section 5
	PMAC2x	1	$2n$	$O(q^2 / 2^{2n} + q^3 / 2^{3n})$	Section 4

on this aspect, we developed an alternative analysis for our construction and derived a corrected bound for a PMAC_TBC1k-like variant with n -bit output. So, despite the assumption in the original proof, we confirm that Naito’s MAC is secure for close to 2^{n-2} blocks processed under the same key.

Contribution. Our contributions are threefold: first, we propose a BBB-secure parallelizable MAC, called PMAC2x, which produces $2n$ -bit outputs and bases on a tweakable block cipher. Figure 1 provides a schematic illustration. Our MAC differs from PMAC_TBC1k mainly in the fact of the extended output, and minorly in the point that we add support for inputs whose length is not a multiple of n . As our second contribution, we briefly revisit the analysis by Naito on PMAC_TBC1k and show that we can easily adapt our proof for PMAC2x and derive a secure variant that we call PMACx which XORs both its outputs and produces only n -bit tags. Table 1 compares our constructions to earlier parallelizable BBB-secure MACs. As our third contribution, we combine PMAC2x with the purely IV-based variant of Counter-in-Tweak to a single-primitive, single-key deterministic authenticated encryption scheme, which we call SIVx, and which provides BBB-security without assumptions about nonces.

Earlier Parallelizable MACs. A considerable amount of works considered parallel MACs; parallel XOR-MACs have already been introduced in 1995 by Bellare et al. [2]; their constructions fed the message blocks together with a counter into a primitive to obtain stateful and randomized MACs. Bernstein [5] published the Protected Counter Sum (PCS), which transformed an XOR-MAC with an independent PRF into a stateless deterministic MAC. PMAC was described by Black and Rogaway first in [6], and was slightly modified to PMAC1 in [23]. Since then, the security of PMAC has been rigorously studied in various works [12, 14, 16, 18, 19]. The first BBB-secure parallelizable MAC was proposed by Yasuda [25]; His PMAC⁺ construction is a three-key version of PMAC which possesses two n -bit state values, which are processed by two independently keyed PRPs, and are XORed to produce the tag. Datta et al. [7] derived a single-key version thereof, called 1k_PMAC⁺. While those are rate-1 designs with larger internal state, there also exist slightly less efficient proposals with smaller state. Yasuda [26] introduced PMAC with parity (PMAC/P), which processes each sequence of r consecutive message blocks in PMAC-like

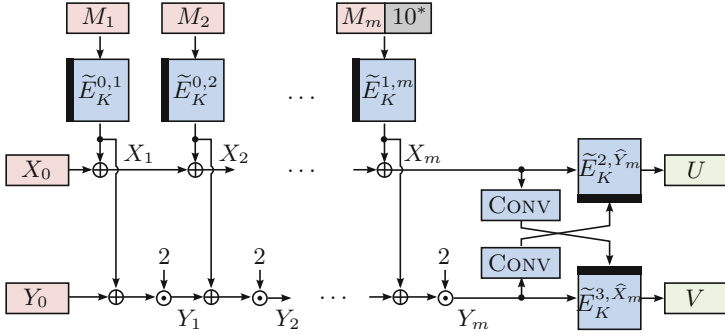


Fig. 1. Processing an m -block message with a partial final block in PMAC2x. $\tilde{E} : \{0, 1\}^k \times \{0, 1\}^d \times \{0, 1\}^t \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ denotes a tweakable block cipher, $\text{CONV} : \{0, 1\}^n \rightarrow \{0, 1\}^t$ a regular function, and \odot multiplication in Galois-Field $\mathbb{GF}(2^n)$.

manner, but inserts the XOR sum of those r blocks as an additional block. Zhang’s PMACX construction [27] generalized PMAC/P by multiplying the input with an MDS matrix before authentication. In a similar direction goes LIGHTMAC [13], a lightweight variant similar to Bernstein’s PCS. However, the security guarantees of all earlier parallelizable MACs in this paragraph are far from the optimal PRF bound.

2 Preliminaries

General Notation. We use lowercase letters x, y for indices and integers, uppercase letters X, Y for binary strings and functions, calligraphic uppercase letters \mathcal{X}, \mathcal{Y} for sets; $X \parallel Y$ for the concatenation of binary strings X and Y , and $X \oplus Y$ for their bitwise XOR. We indicate the length of X in bits by $|X|$, and write X_i for the i -th block, $X[i]$ for the i -th most-significant bit of X , and $X[i..j]$ for the bit sequence $X[i], \dots, X[j]$. We denote by $X \leftarrow \mathcal{X}$ that X is chosen uniformly at random from the set \mathcal{X} . We define $\text{Func}(\mathcal{X}, \mathcal{Y})$ for the set of all functions $F : \mathcal{X} \rightarrow \mathcal{Y}$, $\text{Perm}(\mathcal{X})$ for the set of all permutations $\pi : \mathcal{X} \rightarrow \mathcal{X}$, and $\widetilde{\text{Perm}}(\mathcal{T}, \mathcal{X})$ for the set of tweaked permutations over \mathcal{X} with tweak space \mathcal{T} . We define by $X_1, \dots, X_j \stackrel{x}{\leftarrow} X$ an injective splitting of a string X into blocks of x -bit such that $X = X_1 \parallel \dots \parallel X_j$, $|X_i| = x$ for $1 \leq i \leq j - 1$, and $|X_j| \leq x$. For two sets \mathcal{X} and \mathcal{Y} , let $\mathcal{X} \stackrel{\cup}{\leftarrow} \mathcal{Y}$ denote $\mathcal{X} \leftarrow \mathcal{X} \cup \mathcal{Y}$. A uniform random function $\rho : \mathcal{X} \rightarrow \mathcal{Y}$ is a random variable uniformly distributed over $\text{Func}(\mathcal{X}, \mathcal{Y})$. Given a function $F : \mathcal{X} \rightarrow \mathcal{Y}$, we write $\text{domain}(F)$ for the set of all inputs $X \in \mathcal{X}$ to F that occurred *before* (i.e., excluding) the current query; similarly, we write $\text{range}(F)$ for the set of all outputs $Y \in \mathcal{Y}$ that occurred before the current query. We borrow the notation for a restriction on a set from [8]: let $\mathcal{Q} \subseteq (\mathcal{X} \times \mathcal{Y} \times \mathcal{Z})^*$, then we denote by $\mathcal{Q}_{|Y, Z} = \{(Y, Z) \mid \exists X : (X, Y, Z) \in \mathcal{Q}\}$ the restriction of \mathcal{Q} to values $Y \in \mathcal{Y}$ and $Z \in \mathcal{Z}$. This generalizes in the obvious way.

For an event E , we denote by $\Pr[E]$ the probability of E . We write $\langle x \rangle_n$ for the binary representation of an integer x as an n -bit string, or short $\langle x \rangle$ if n is clear from the context, in big-endian manner, e. g., $\langle 1 \rangle_4$ would be encoded to (0001) .

Adversaries. An adversary \mathbf{A} is an efficient Turing machine that interacts with a given set of oracles that appear as black boxes to \mathbf{A} . We denote by $\mathbf{A}^{\mathcal{O}}$ the output of \mathbf{A} after interacting with some oracle \mathcal{O} . We write $\Delta_{\mathbf{A}}(\mathcal{O}^1; \mathcal{O}^2)$ for the advantage of \mathbf{A} to distinguish between oracles \mathcal{O}^1 and \mathcal{O}^2 . All probabilities are defined over the random coins of the oracles and those of the adversary, if any. We write $\text{Adv}_F^X(q, \ell, \theta) := \max_{\mathbf{A}} \{\text{Adv}_F^X(\mathbf{A})\}$ for the maximal advantage over all X -adversaries \mathbf{A} on F that run in time at most θ and pose at most q queries of at most ℓ blocks in total to its oracles. Wlog., we assume that \mathbf{A} never asks queries to which it already knows the answer.

We will provide pseudocode descriptions of the oracles, which will be referred to as games, according to the game-playing framework by Bellare and Rogaway [3]. Each game consists of a set of procedures. We define $\Pr[G(\mathbf{A}) \Rightarrow x]$ as the probability that the game G outputs x when given \mathbf{A} as input.

Definition 1 (TPRP Advantage). Let $\tilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{X} \rightarrow \mathcal{X}$ be a tweakable block cipher with non-empty key space \mathcal{K} and tweak space \mathcal{T} . Let \mathbf{A} a computationally bounded adversary with access to an oracle, where $K \leftarrow \mathcal{K}$ and $\tilde{\pi} \leftarrow \widetilde{\text{Perm}}(\mathcal{T}, \mathcal{X})$. Then, the TPRP advantage of \mathbf{A} on \tilde{E} is defined as $\text{Adv}_{\tilde{E}}^{\text{TPRP}}(\mathbf{A}) := \Delta_{\mathbf{A}}(\tilde{E}_K; \tilde{\pi})$.

A MAC is a tuple of functions $F : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ with non-empty key space \mathcal{K} , and a generic verification function $\text{VERIFY} : \mathcal{K} \times \mathcal{X} \times \mathcal{Y} \rightarrow \{1, \perp\}$, where for all $K \in \mathcal{K}$ and $X \in \mathcal{X}$, $\text{VERIFY}_K(X, Y)$ returns 1 iff $F_K(X) = Y$ and \perp otherwise. We use \perp , when in place of an oracle, to always return the invalid symbol \perp . It is well-known that if F is a secure PRF, it is also a secure MAC; however, the converse statement is not necessarily true.

Definition 2 (PRF Advantage). Let $F : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ be a function with non-empty key space \mathcal{K} , and \mathbf{A} a computationally bounded adversary with access to an oracle, where $K \leftarrow \mathcal{K}$ and $\rho \leftarrow \text{Func}(\mathcal{X}, \mathcal{Y})$. Then, the PRF advantage of \mathbf{A} on F is defined as $\text{Adv}_F^{\text{PRF}}(\mathbf{A}) := \Delta_{\mathbf{A}}(F_K; \rho)$.

3 Definition of PMAC2x and PHASHx

This section defines the generic PMAC2x construction and its underlying hash function PHASHx. Fix integers k, n, t, d , with $d \geq 2$. Let $\mathcal{K} = \{0, 1\}^k$ and $\mathcal{T} = \{0, 1\}^t$ be non-empty sets of keys and tweaks, respectively. Moreover, derive a set of domains $\mathcal{D} := \{0, 1, 2, 3\} = \{0, 1\}^d$ which are encoded as their respective d -bit values, and a domain-tweak set $\mathcal{T}' := \mathcal{D} \times \mathcal{T}$. Let $\mathcal{M} \subseteq (\{0, 1\}^n)^*$ denote an input space. Further, let $\tilde{E} : \mathcal{K} \times \mathcal{T}' \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ denote a tweakable block cipher. We will often write $\tilde{E}_{K, D, T}^{\mathcal{D}, \mathcal{T}}(\cdot)$ as short form of $\tilde{E}(K, D, T, \cdot)$. $K \in \mathcal{K}$, $D \in \mathcal{D}$, and

Algorithm 1. Definition of $\text{PMAC2x}[\tilde{E}]$ and its internal hash function $\text{HASHx}[\tilde{E}]$ with a tweakable block cipher $\tilde{E} : \mathcal{K} \times \{0, 1\}^d \times \{0, 1\}^t \times \{0, 1\}^n \rightarrow \{0, 1\}^n$. n , t , and d denote state, tweak and domain sizes, respectively.

11: function $\text{PMAC2x}[\tilde{E}_K](M)$ 12: $(X_m, Y_m) \leftarrow \text{HASHx}[\tilde{E}_K](M)$ 13: $\hat{X}_m \leftarrow \text{CONV}(X_m)$ 14: $\hat{Y}_m \leftarrow \text{CONV}(Y_m)$ 15: $U \leftarrow \tilde{E}_K^{2, \hat{Y}_m}(X_m)$ 16: $V \leftarrow \tilde{E}_K^{3, \hat{X}_m}(Y_m)$ 17: return $(U \parallel V)$ 21: function $\text{CONV}(X)$ 22: if $t \geq n$ then 23: return X 24: return $X[1..t]$ 31: function $\tilde{E}_K^{D, T}(X)$ 32: $\tilde{T} \leftarrow \langle D \rangle_d \parallel T[1..t]$ 33: return $\tilde{E}_K^{\tilde{T}}(X)$	41: function $\text{HASHx}[\tilde{E}_K](M)$ 42: $X_0 \leftarrow 0^n; Y_0 \leftarrow 0^n$ 43: $(M_1, \dots, M_m) \xleftarrow{n} M$ 44: for $i \leftarrow 1$ to $m - 1$ do 45: $Z_i \leftarrow \tilde{E}_K^{0, (i)}(M_i)$ 46: $X_i \leftarrow X_{i-1} \oplus Z_i$ 47: $Y_i \leftarrow 2 \cdot (Y_{i-1} \oplus Z_i)$ 48: if $ M_m = n$ then 49: $Z_m \leftarrow \tilde{E}_K^{0, (m)}(M_m)$ 50: else 51: $M_m^* \leftarrow M_m \parallel 10^{n- M_m -1}$ 52: $Z_m \leftarrow \tilde{E}_K^{1, (m)}(M_m^*)$ 53: $X_m \leftarrow X_{m-1} \oplus Z_m$ 54: $Y_m \leftarrow 2 \cdot (Y_{m-1} \oplus Z_m)$ 55: return (X_m, Y_m)
---	--

$T \in \mathcal{T}$ denote key, domain, and tweak, respectively. $\text{CONV} : \{0, 1\}^n \rightarrow \{0, 1\}^t$ be a regular function² which is used to convert the outputs of HASHx , X_m and Y_m , so they can be used as tweaks of \tilde{E} in the finalization step. We denote by $\text{PMAC2x}[\tilde{E}]$ and $\text{HASHx}[\tilde{E}]$ the instantiation of PMAC2x and HASHx with \tilde{E} . Both are defined, with a default instantiation of CONV , in Algorithm 1.

4 Security Analysis of PMAC2x

Theorem 1. Let \tilde{E} and $\text{PMAC2x}[\tilde{E}]$ be defined as in Sect. 3. Let $d + t = n$, and let $m < 2^t$ denote the maximum number of n -bit blocks of any query. Then

$$\begin{aligned} \text{Adv}_{\text{PMAC2x}[\tilde{E}]}^{\text{PRF}}(q, \ell, \theta) &\leq \frac{2^{2d}q^2}{2 \cdot (2^n - q)^2} + \frac{2^d q^3}{3 \cdot 2^{2n}(2^n - q)} + \frac{2^d q^2}{2^n(2^n - q)} \\ &\quad + \text{Adv}_{\tilde{E}}^{\text{TPRP}}(\ell + 2q, O(\theta + \ell + 2q)). \end{aligned}$$

The final term results from a standard argument after replacing the tweakable block cipher \tilde{E} by a random tweakable permutation $\tilde{\pi} \leftarrow \widetilde{\text{Perm}}(\mathcal{T}', \{0, 1\}^n)$.

Let \mathbf{A} be an adversary that makes at most q queries of at most m blocks each and of at most ℓ blocks in total. We assume, \mathbf{A} does not ask duplicate queries and has the goal to distinguish between a $\text{PMAC2x}[\tilde{\pi}]$ oracle with an internally sampled tweaked permutations $\tilde{\pi}$ and a random function $\rho : \{0, 1\}^* \rightarrow \{0, 1\}^{2n}$.

We consider the game described in Algorithm 2. The game *without* the boxed statements coincides with a random function ρ , whereas the game *with* them

² A function is called regular iff all outputs are produced by an equal number of inputs.

Algorithm 2. Main Game, initialization, finalization, and subroutines. Boxed statements belong exclusively to the real world.

<pre> 1: procedure INITIALIZE 2: $\text{bad}_U \leftarrow \text{false}; \text{bad}_V \leftarrow \text{false}; \mathcal{Q} \leftarrow \emptyset$ 3: $X_0 \leftarrow 0^n; Y_0 \leftarrow 0^n; b \leftarrow \{0, 1\}$ </pre>	<pre> 11: function FINALIZE(b') 12: $\text{bad} \leftarrow \text{bad}_U \vee \text{bad}_V$ 13: return $b' = b \vee \text{bad}$ </pre>
<pre> 21: function ORACLE(M) 22: $(X_m, Y_m) \leftarrow \text{PHASH}_X[\tilde{\pi}](M)$ 23: $\hat{X}_m \leftarrow \text{CONV}(X_m)$ 24: $\hat{Y}_m \leftarrow \text{CONV}(Y_m)$ 25: $U \leftarrow \{0, 1\}^n$ 26: $V \leftarrow \{0, 1\}^n$ 27: if $(\hat{X}_m, \hat{Y}_m) \in \mathcal{Q}_{ \hat{X}_m, \hat{Y}_m}$ then 28: $(U, V) \leftarrow \text{Case1}(X_m, Y_m, \hat{X}_m, \hat{Y}_m)$ 29: else if $U \in \text{range}(\tilde{\pi}^{2, \hat{Y}_m}) \wedge$ 30: $V \in \text{range}(\tilde{\pi}^{3, \hat{X}_m})$ then 31: $(U, V) \leftarrow \text{Case2}(X_m, Y_m, \hat{X}_m, \hat{Y}_m)$ 32: else if $U \in \text{range}(\tilde{\pi}^{2, \hat{Y}_m}) \wedge$ 33: $V \notin \text{range}(\tilde{\pi}^{3, \hat{X}_m})$ then 34: $(U, V) \leftarrow \text{Case3}(\hat{X}_m, \hat{Y}_m, U, V)$ 35: else if $U \notin \text{range}(\tilde{\pi}^{2, \hat{Y}_m}) \wedge$ 36: $V \in \text{range}(\tilde{\pi}^{3, \hat{X}_m})$ then 37: $(U, V) \leftarrow \text{Case4}(\hat{X}_m, \hat{Y}_m, U, V)$ 38: else if $U \notin \text{range}(\tilde{\pi}^{2, \hat{Y}_m}) \wedge$ 39: $V \notin \text{range}(\tilde{\pi}^{3, \hat{X}_m})$ then 40: $(U, V) \leftarrow \text{Case5}(\hat{X}_m, \hat{Y}_m, U, V)$ 41: $\mathcal{Q} \leftarrow \bigcup \{(\hat{X}_m, \hat{Y}_m, U, V)\}$ 42: $\tilde{\pi}^{2, \hat{Y}_m}[X_m] \leftarrow U$ 43: $\tilde{\pi}^{3, \hat{X}_m}[Y_m] \leftarrow V$ 44: return $(U \parallel V)$ </pre>	<pre> 51: function CASE1($X_m, Y_m, \hat{X}_m, \hat{Y}_m$) 52: if $X_m \in \text{domain}(\tilde{\pi}^{2, \hat{Y}_m})$ then 53: $U \leftarrow \tilde{\pi}^{2, \hat{Y}_m}[X_m]$ 54: else 55: $U \leftarrow \{0, 1\}^n \setminus \text{range}(\tilde{\pi}^{2, \hat{Y}_m})$ 56: if $Y_m \in \text{domain}(\tilde{\pi}^{3, \hat{X}_m})$ then 57: $V \leftarrow \tilde{\pi}^{3, \hat{X}_m}[Y_m]$ 58: else 59: $V \leftarrow \{0, 1\}^n \setminus \text{range}(\tilde{\pi}^{3, \hat{X}_m})$ 60: $\text{bad}_U \leftarrow \text{bad}_V \leftarrow \text{true}$ 61: return (U, V) </pre>
<pre> 95: function CASE5($\hat{X}_m, \hat{Y}_m, U, V$) 96: return (U, V) </pre>	<pre> 71: function CASE2($X_m, Y_m, \hat{X}_m, \hat{Y}_m$) 72: $U \leftarrow \{0, 1\}^n \setminus \text{range}(\tilde{\pi}^{2, \hat{Y}_m})$ 73: $V \leftarrow \{0, 1\}^n \setminus \text{range}(\tilde{\pi}^{3, \hat{X}_m})$ 74: $\text{bad}_U \leftarrow \text{bad}_V \leftarrow \text{true}$ 75: return (U, V) </pre>
<pre> </pre>	<pre> 81: function CASE3($\hat{X}_m, \hat{Y}_m, U, V$) 82: $U \leftarrow \{0, 1\}^n \setminus \text{range}(\tilde{\pi}^{2, \hat{Y}_m})$ 83: $\text{bad}_U \leftarrow \text{true}$ 84: return (U, V) </pre>
<pre> </pre>	<pre> 91: function CASE4($\hat{X}_m, \hat{Y}_m, U, V$) 92: $V \leftarrow \{0, 1\}^n \setminus \text{range}(\tilde{\pi}^{3, \hat{X}_m})$ 93: $\text{bad}_V \leftarrow \text{true}$ 94: return (U, V) </pre>

exactly represents $\text{PMAC2x}[\tilde{\pi}]$, performing lazy sampling for the permutations $\tilde{\pi}^{2, \hat{Y}_m}(\cdot)$ and $\tilde{\pi}^{3, \hat{X}_m}(\cdot)$, for all $\hat{X}_m, \hat{Y}_m \in \{0, 1\}^t$. Both algorithms differ only when bad events occur. Hence, by the fundamental lemma of game playing [4], it holds

$$\Pr[\mathbf{A}^{\text{PMAC2x}[\tilde{\pi}](\cdot)} \Rightarrow 1] - \Pr[\mathbf{A}^{\rho(\cdot)} \Rightarrow 1] \leq \Pr[\mathbf{A} \text{ sets bad}].$$

In the remainder, we consider five cases which cover all possibilities:

- Case1: $(\hat{X}_m, \hat{Y}_m) \in \mathcal{Q}_{|\hat{X}_m, \hat{Y}_m}$.

- Case2: $(\widehat{X}_m, \widehat{Y}_m) \notin \mathcal{Q}_{|\widehat{X}_m, \widehat{Y}_m} \wedge U \in \text{range}(\widetilde{\pi}^2, \widehat{Y}_m) \wedge V \in \text{range}(\widetilde{\pi}^3, \widehat{X}_m)$.
- Case3: $(\widehat{X}_m, \widehat{Y}_m) \notin \mathcal{Q}_{|\widehat{X}_m, \widehat{Y}_m} \wedge U \in \text{range}(\widetilde{\pi}^2, \widehat{Y}_m) \wedge V \notin \text{range}(\widetilde{\pi}^3, \widehat{X}_m)$.
- Case4: $(\widehat{X}_m, \widehat{Y}_m) \notin \mathcal{Q}_{|\widehat{X}_m, \widehat{Y}_m} \wedge U \notin \text{range}(\widetilde{\pi}^2, \widehat{Y}_m) \wedge V \in \text{range}(\widetilde{\pi}^3, \widehat{X}_m)$.
- Case5: $(\widehat{X}_m, \widehat{Y}_m) \notin \mathcal{Q}_{|\widehat{X}_m, \widehat{Y}_m} \wedge U \notin \text{range}(\widetilde{\pi}^2, \widehat{Y}_m) \wedge V \notin \text{range}(\widetilde{\pi}^3, \widehat{X}_m)$.

We list Case 5 only for the sake of completeness. It is easy to see that in Case 5, the output of the game is indistinguishable between the worlds. We use M^i , \widehat{X}_m^i , \widehat{Y}_m^i , U^i , V^i to refer to the respective values of the i -th query, where $i \in [1, q]$, and assume it is the current query of the adversary. Additionally, we will use indices j and k , where $j, k \in [1, i - 1]$, to refer to previous queries.

Case1: For this case, we revisit the fact that for two fixed disjoint queries M^i and M^j , the values X_m and Y_m are results of two random variables. A variant of the proof is given e.g. in [17, Sect. 3.3], and revisited in the following only for the sake of completeness. Fix query indices $i \in [1, q]$ and $j \in [1, i - 1]$. Let m and m' denote the number of blocks of the i -th and j -th query, respectively; moreover, let X_m^i, Y_m^i denote the values X_m and Y_m of the i -th query and $X_{m'}^j, Y_{m'}^j$ those of the j -th query, respectively. $X_m^i, Y_m^i, X_{m'}^j$, and $Y_{m'}^j$ result from:

$$\begin{aligned} X_m^i &= C_1^i \oplus C_2^i \oplus \dots \oplus C_m^i & Y_m^i &= 2^m C_1^i \oplus 2^{m-1} C_2^i \oplus \dots \oplus 2 \cdot C_m^i, \\ X_{m'}^j &= C_1^j \oplus C_2^j \oplus \dots \oplus C_{m'}^j & Y_{m'}^j &= 2^{m'} C_1^j \oplus 2^{m'-1} C_2^j \oplus \dots \oplus 2 \cdot C_{m'}^j. \end{aligned}$$

So, we want to bound the probability for the following equational system:

$$\begin{aligned} C_1^i \oplus C_2^i \oplus \dots \oplus C_m^i &= C_1^j \oplus C_2^j \oplus \dots \oplus C_{m'}^j, \\ 2^m C_1^i \oplus 2^{m-1} C_2^i \oplus \dots \oplus 2 \cdot C_m^i &= 2^{m'} C_1^j \oplus 2^{m'-1} C_2^j \oplus \dots \oplus 2 \cdot C_{m'}^j. \end{aligned}$$

There exist three distinct subcases which cover all possibilities:

- **Subcase 1: $m \neq m'$.** In this case, the equations above provide a unique solution set for two random variables; thus, the probability that the equations hold for two fixed queries is upper bounded by $1/(2^n - (i - 1))^2$.
- **Subcase 2: $m = m'$ and there exists $\alpha \in [1, m]$ s.t. $C_\alpha^i \neq C_\alpha^j$ and for all $\beta \in [1, m]$ with $\beta \neq \alpha$: $C_\beta^i = C_\beta^j$.** In this case, both messages share only a single different output block. Thus, the equations above never hold.
- **Subcase 3: $m = m'$ and there exist distinct $\beta \in [1, m]$ with $\beta \neq \alpha$: $C_\beta^i = C_\beta^j$.** In this case, both messages share only a single different output block. Thus, the equations above can never hold.

So, the probability for two fixed disjoint queries M^i and M^j that $(X_m^i, Y_m^i) = (X_{m'}^j, Y_{m'}^j)$ holds, is bounded by $1/(2^n - q)^2$. Since \widehat{X}_m^i and \widehat{Y}_m^i are derived from X_m^i and Y_m^i , respectively by a regular function (and so are $\widehat{X}_{m'}^j$ and $\widehat{Y}_{m'}^j$ derived from $X_{m'}^j$ and $Y_{m'}^j$, respectively), it follows that the probability of $(\widehat{X}_m^i, \widehat{Y}_m^i) = (\widehat{X}_{m'}^j, \widehat{Y}_{m'}^j)$ to hold for the i -th and j -th query, with $j \in [1, i - 1]$, is at most

$$(i - 1) \cdot \frac{2^d}{(2^n - q)} \cdot \frac{2^d}{(2^n - q)} = \frac{2^{2d}(i - 1)}{(2^n - q)^2}.$$

Case2: In this case, there exists some previous query $(\widehat{X}_{m'}, \widehat{Y}_{m'}, U^j, V^j)$ s.t. $U = U^j \wedge \widehat{Y}_m = \widehat{Y}_{m'}$, and a distinct previous query $(\widehat{X}_{m''}, \widehat{Y}_{m''}, U^k, V^k)$ s.t. $V = V^k \wedge \widehat{X}_m = \widehat{X}_{m''}$. From our assumption $(\widehat{X}_m, \widehat{Y}_m) \notin \mathcal{Q}_{|\widehat{X}_m, \widehat{Y}_m}$, it follows that $j \neq k$; otherwise, the current query would have stepped into Case1 instead. We can bound the probability by

$$\begin{aligned} & \Pr \left[(U = U^j \wedge \widehat{Y}_m = \widehat{Y}_{m'}) \wedge (V = V^k \wedge \widehat{X}_m = \widehat{X}_{m''}) \right] \\ & \leq \Pr \left[U = U^j \wedge \widehat{Y}_m = \widehat{Y}_{m'} \wedge V = V^k \mid \widehat{X}_m = \widehat{X}_{m''} \right]. \end{aligned}$$

U and V are chosen independently and uniformly at random from $\{0, 1\}^n$ each, and can collide with at most $i - 1$ previous values U^j and at most $i - 1$ previous values V^k , respectively. For fixed j and k , the probability for U to collide with U^j is upper bounded by $1/2^n$, and independently, the probability for V to collide with V^k is also $1/2^n$. Since the game chooses U and V independently from Y_m , the probability that \widehat{Y}_m collides with $\widehat{Y}_{m'}$ is at most $2^d/(2^n - q)$ since we assumed that the adversary poses no duplicate queries, and therefore, Y_m and $Y_{m'}$ are results of two random variables. Since the collision of $U = U^j$ already fixes the colliding query pair, there is no additional factor $(i - 1)$ for the choice of which pairs of \widehat{Y}_m and $\widehat{Y}_{m'}$ to collide. It follows that the probability for this case to occur at the i -th query is upper bounded by

$$\frac{i - 1}{2^n} \cdot \frac{i - 2}{2^n} \cdot \frac{2^d}{2^n - q} \leq \frac{2^d(i - 1)^2}{2^{2n}(2^n - q)}.$$

Case3: In this case, there exists some previous query $(\widehat{X}_{m'}, \widehat{Y}_{m'}, U^j, V^j)$ s.t. $U = U^j \wedge \widehat{Y}_m = \widehat{Y}_{m'}$. From our assumption $(\widehat{X}_m, \widehat{Y}_m) \notin \mathcal{Q}_{|\widehat{X}_m, \widehat{Y}_m}$ and $\widehat{Y}_m = \widehat{Y}_{m'}$, follows that $\widehat{X}_m \neq \widehat{X}_{m'}$, holds, like in Case2. We can bound the probability by

$$\begin{aligned} & \Pr \left[U = U^j \wedge \widehat{Y}_m = \widehat{Y}_{m'} \wedge V \notin \text{range}(\widetilde{\pi}^3, \widehat{X}_m) \right] \\ & \leq \Pr \left[U = U^j \wedge \widehat{Y}_m = \widehat{Y}_{m'} \mid V \notin \text{range}(\widetilde{\pi}^3, \widehat{X}_m) \right]. \end{aligned}$$

For a fixed j -th query, the probability that \widehat{Y}_m collides with $\widehat{Y}_{m'}$ is at most $2^d/(2^n - q)$. Since U is chosen uniformly at random from $\{0, 1\}^n$ and independently from Y_m , U can collide with U^j with probability $1/2^n$. So, the probability of this case to occur for the i -th query can be upper bounded by

$$\frac{2^d}{2^n - q} \cdot \frac{i - 1}{2^n} = \frac{2^d(i - 1)}{2^n(2^n - q)}.$$

Case4: In this case, it holds that $V = V^j \wedge \widehat{X}_m = \widehat{X}_{m'}$. From $(\widehat{X}_m, \widehat{Y}_m) \notin \mathcal{Q}_{|\widehat{X}_m, \widehat{Y}_m}$ and $\widehat{X}_m = \widehat{X}_{m'}$ follows here that $\widehat{Y}_m \neq \widehat{Y}_{m'}$, holds, analogously to

Case2 and Case3. We can bound the probability by

$$\begin{aligned} & \Pr \left[V = V^j \wedge \widehat{X}_m = \widehat{X}_{m'}^j, \wedge U \notin \text{range}(\widetilde{\pi}^2, \widehat{Y}_m) \right] \\ & \leq \Pr \left[V = V^j \wedge \widehat{X}_m = \widehat{X}_{m'}^j \mid U \notin \text{range}(\widetilde{\pi}^2, \widehat{Y}_m) \right]. \end{aligned}$$

Obviously, this case can be handled similarly as Case3. For a fixed j -th query, the probability that \widehat{X}_m collides with $\widehat{X}_{m'}^j$ is at most $2^d/(2^n - q)$. Since V is chosen uniformly at random from $\{0, 1\}^n$ and independently from X_m , V can collide with V^j with probability $1/2^n$. So, the probability of this case to occur for the i -th query can also be upper bounded by

$$\frac{2^d(i-1)}{2^n(2^n - q)}.$$

Taking the terms from all cases and the union bound over at most q queries gives

$$\begin{aligned} \Pr [\mathbf{A} \text{ sets bad}] & \leq \sum_{i=1}^q \left(\frac{2^{2d}(i-1)}{(2^n - q)^2} + \frac{2^d(i-1)^2}{2^{2n}(2^n - q)} + 2 \cdot \frac{2^d(i-1)}{2^n(2^n - q)} \right) \\ & \leq \frac{2^{2d}q^2}{2 \cdot (2^n - q)^2} + \frac{2^d q^3}{3 \cdot 2^{2n}(2^n - q)} + \frac{2^d q^2}{2^n(2^n - q)}. \end{aligned}$$

5 Security Analysis of PMACx

This section considers a variant of PMAC2x, PMACx, that adds a final XOR to produce only an n -bit tag, following the design of PMAC_TBC1K. A schematic illustration is given in Fig. 2. We revisit the assumption by Naito, and show that our proof of PMAC2x needs only a slight adaption for PMACx.

Previous Analysis. Theorem 2 in [17] proves the security of PMAC_TBC1K with the help of an analysis of multi-collisions of the final chaining values (X_m and Y_m in our notation). Note that our notation differs from [17] to be consistent to our previous section. Define two monotone events mcoll_1 and mcoll_2 . Let ρ and ξ denote positive integers and define three sets \mathcal{X} , \mathcal{Y} , and \mathcal{Q} which store the values \widehat{X}_m^i , \widehat{Y}_m^i , and the tuples (X_m^i, \widehat{Y}_m^i) , respectively, of the queries $1 \leq i \leq q$.

$$\begin{aligned} \text{mcoll}_1 & := (\exists \widehat{X}_m^1, \dots, \widehat{X}_m^\rho \in \mathcal{X} \text{ s.t. } \widehat{X}_m^1 = \dots = \widehat{X}_m^\rho) \\ & \quad \vee (\exists \widehat{Y}_m^1, \dots, \widehat{Y}_m^\rho \in \mathcal{Y} \text{ s.t. } \widehat{Y}_m^1 = \dots = \widehat{Y}_m^\rho), \\ \text{mcoll}_2 & := \exists (X_m^1, \widehat{Y}_m^1), \dots, (X_m^\xi, \widehat{Y}_m^\xi) \in \mathcal{Q} \text{ s.t. } (X_m^1, \widehat{Y}_m^1) = \dots = (X_m^\xi, \widehat{Y}_m^\xi). \end{aligned}$$

The original proof further defined a monotone compound event $\text{mcoll} := \text{mcoll}_1 \vee \text{mcoll}_2$ and used the fact that

$$\begin{aligned} \Pr [\mathbf{A} \text{ sets bad}] & = \Pr [\mathbf{A} \text{ sets bad} \wedge \text{mcoll}] + \Pr [\mathbf{A} \text{ sets bad} \wedge \neg \text{mcoll}] \\ & \leq \Pr [\text{mcoll}_1] + \Pr [\text{mcoll}_2] + \Pr [\mathbf{A} \text{ sets bad} \mid \neg \text{mcoll}]. \end{aligned}$$

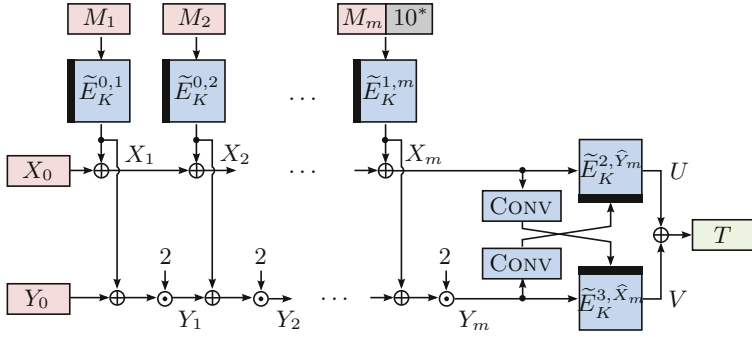


Fig. 2. PMACx, the variant of PMAC2x with n -bit output, following the design of PMAC_TBC1k.

The analysis in [17] bounds $\Pr[\text{mcoll}_1]$ as

$$\Pr[\text{mcoll}_1] \leq 2 \cdot 2^t \cdot \binom{q}{\rho} \cdot \left(\frac{2^{n-t}}{2^n - q}\right)^\rho \leq 2^{t+1} \cdot \left(\frac{2^{n-t} \cdot eq}{\rho(2^n - q)}\right)^\rho,$$

using Stirling’s approximation $x! \geq (x/e)^x$ for any x . Note, in PMAC_TBC1k, the domain size in PMAC2x is fixed to $d = 2$ bits. The bound above consists of the probability that ρ values are all equal, $(2^{n-t}/(2^n - q))^\rho$; the fact that there are 2^t tweak values; and the $\binom{q}{\rho}$ possible ways to choose ρ out of q values. However, the bound holds *only if* the ρ colliding tweaks stem from ρ linearly independent random variables, which is *not* necessarily the case. Imagine a sequence of 2^m queries which combine pair-wise distinct blocks $\{M_i, M'_i\}$ with $M_i \neq M'_i$, for $1 \leq i \leq m$ position-wise, i. e., we have 2^m queries of m blocks each: $Q^0 = (M_1, M_2, M_3, \dots, M_m)$, $Q^1 = (M'_1, M_2, M_3, \dots, M_m)$, $Q^2 = (M_1, M'_2, M_3, \dots, M_m)$, \dots , $Q^{2^m-1} = (M'_1, M'_2, M'_3, \dots, M'_m)$. When used as queries to PMAC_TBC1k, the 2^m resulting values X_m^i , for $0 \leq i \leq 2^m - 1$, depend on the linear combination of only $2m$ random variables. A similar argument holds for the values Y_m^i , as well as for the similarly treated bound of mcoll_2 . Thus, the multi-collision bound demands a significantly more detailed analysis.

Fixing the Analysis. From our proof for PMAC2x, we can now derive a corollary for a similar security bound for PMACx, which again can be easily transformed into a bound for PMAC_TBC1k.

Corollary 1. Let \tilde{E} and PMAC2x $[\tilde{E}]$ be defined as in Sect. 3. Let $d + t = n$, and let $m < 2^t$ denote the maximum number of n -bit blocks of any query. Then, it holds that $\text{Adv}_{\text{PMACx}[\tilde{E}]}^{\text{PRF}}(q, \ell, \theta) \leq \text{Adv}_{\text{PMAC2x}[\tilde{E}]}^{\text{PRF}}(q, \ell, \theta)$.

The proof can use a game almost identical to that in Algorithm 2, where we only modify Line 44 to return the XOR of U and V . This is shown in Algorithm 3. All further procedures and functions remain identical to those in Algorithm 2.

If $(U \parallel V)$ is indistinguishable from outputs of a $2n$ -bit random function ρ , then each of the n -bit outputs U and V can be considered random. It follows,

Algorithm 3. The updated game for the security proof of PMACx. Only the double-boxed statement changes compared with the game in Algorithm 2.

```

21: function ORACLE( $M$ )
22:   ...
42:    $\tilde{\pi}^{2, \hat{Y}_m}[X_m] \leftarrow U$ 
43:    $\tilde{\pi}^{3, \hat{X}_m}[Y_m] \leftarrow V$ 
44:   return  $T \leftarrow U \oplus V$ 
    
```

if U is indistinguishable from n -bit values, then the XOR sum of $U \oplus V$ is also indistinguishable from a random n -bit value. Hence, the PRF advantage of \mathbf{A} on PMACx is upper bounded by that of an adversary \mathbf{A}' on PMAC2x with an equal amount of resources as \mathbf{A} ; hence, the corollary follows.

PMACx and PMAC_TBC1k differ in three aspects: (1) PMACx allows messages whose length is not a multiple of n bits by padding the final block with 10^* and using a distinct tweak for it; (2) PMACx defines a generic d -bit domain encoding and defines a conversion function CONV for deriving the inputs for the finalization; and (3) PMACx adds a final doubling for a simpler and consistent description. Clearly, none of the differences affects the distribution of final chaining values \hat{X}_m and \hat{Y}_m . Hence, when fixing $d = 2$, a security result for PMACx can be easily carried over to a bound for PMAC_TBC1k.

6 Definition and Security Analysis of SIVx

Next, we define the deterministic AE scheme SIVx, which combines PMAC2x and the IV-based Counter-in-Tweak mode IVCTRT. We recall the definitions of IV-based encryption and Deterministic AE security in the full version of this work [11]. Note, that it is straight-forward to derive a nonce-based AE scheme by fixing the nonce length and appending the nonce to the associated data.

IVCTRT. IVCTRT denotes the version of Counter in Tweak [21, Appendix C], which takes a $2n$ -bit random IV plus the message for each encryption. Let $\mathcal{T} = \{0, 1\}^t$, and $\mathcal{T}' = \{0, 1\} \times \mathcal{T}$. The mode first splits $(U, V) \stackrel{n}{\leftarrow} IV$, and uses a given tweakable block cipher $\tilde{E} : \mathcal{K} \times \mathcal{T}' \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ in counter mode, with V as cipher input. Next, it derives $T \leftarrow \text{CONV}'(U)$ from U with a regular function $\text{CONV}' : \{0, 1\}^n \rightarrow \mathcal{T}$ and increments T for every call to \tilde{E} using addition modulo 2^t . We denote by $\text{IVCTRT}[\tilde{E}]$ the instantiation of IVCTRT with \tilde{E} ; from Theorem 1 and Appendix C in [21], we recall the following theorem:

Theorem 2 (ivE Security of IVCTRT). *Let $\tilde{\pi} \leftarrow \widetilde{\text{Perm}}(\mathcal{T}', \{0, 1\}^n)$ be an ideal tweakable block cipher. Let \mathbf{A} be an adversary which asks at most q queries of at most $8 \leq \ell \leq |\mathcal{T}|$ blocks in total. Then*

$$\text{Adv}_{\text{IVCTRT}[\tilde{\pi}]}^{\text{ivE}}(\mathbf{A}) \leq \frac{1}{2^n} + \frac{1}{|\mathcal{T}|} + \frac{4\ell \log q}{|\mathcal{T}|} + \frac{\ell \log^2(\ell)}{2^n}.$$

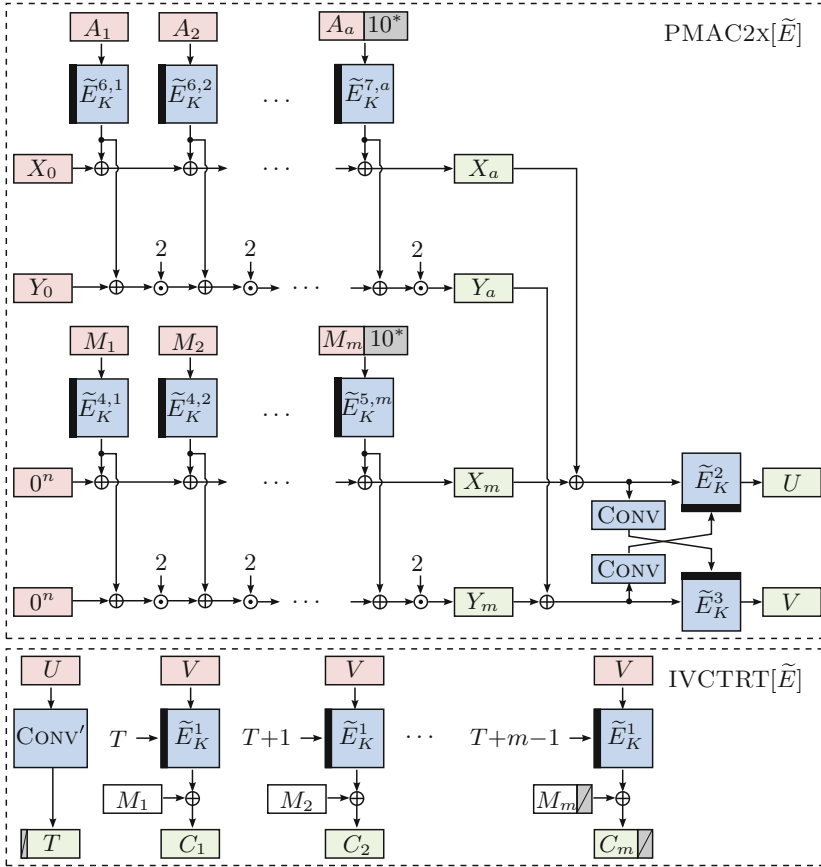


Fig. 3. The deterministic AE scheme SIVx from the composition of $\text{PMAC2x}[\tilde{E}]$ (top), and the $\text{IVCTRT}[\tilde{E}]$ mode of encryption (bottom) [20]. The figure starts the message processing in PMAC2x from 0^n only to prevent that X_0 and Y_0 cancel out.

Definition of SIVx. We define the deterministic AE scheme $\text{SIVx}[\tilde{E}]$ as the composition of $\text{PMAC2x}[\tilde{E}]$ and $\text{IVCTRT}[\tilde{E}]$, as given in Algorithm 4. A schematic illustration of the encryption process is depicted in Fig. 3. In general, we denote by $\text{SIVx}[F, \Pi]$ the instantiation of SIVx with a function F and an IV-based encryption scheme Π in SIVx. To use the same key in all calls to \tilde{E} , we parametrize PHASHx to separate the domains. We use the domains $2 = (0010)_2$ and $3 = (0011)_2$ for the finalization steps, $4 = (0100)_2$ and $5 = (0101)_2$ for processing the associated data, as well as $6 = (0110)_2$ and $7 = (0111)_2$ for processing the message in PMAC2x. We encode them into the $d = 4$ most significant bits of the tweak. Inside IVCTRT, however, we use the single-bit domain 1 in all calls to \tilde{E} for we lose only a single bit from the IV. For concreteness, we define the initial values in PMAC2x as $X_0 = Y_0 = 0^n$.

Algorithm 4. Definition of $\text{SIVx}[\tilde{E}]$. Note that $\text{IVCTRT}[\tilde{E}]$ and its inverse $\text{IVCTRT}^{-1}[\tilde{E}]$ are identical operations. Moreover, we define $\text{HASHx}[\tilde{E}]^{D_1, D_2}$ to denote $\text{HASHx}[\tilde{E}]$ using D_1 as domain for processing full blocks, and using D_2 for a potential partial final block of the input to HASHx .

11: function $\text{SIVx}[\tilde{E}_K](A, M)$	41: function $\text{SIVx}^{-1}[\tilde{E}_K](A, C, \text{TAG})$
12: $\text{TAG} \leftarrow \text{PMAC2x}[\tilde{E}_K](A, M)$	42: $(U, V) \xleftarrow{n} \text{TAG}$
13: $(U, V) \xleftarrow{n} \text{TAG}$	43: $IV \leftarrow \text{CONV}'(U) \parallel V$
14: $IV \leftarrow \text{CONV}'(U) \parallel V$	44: $M \leftarrow \text{IVCTRT}^{-1}[\tilde{E}_K](IV, C)$
15: $C \leftarrow \text{IVCTRT}[\tilde{E}_K](IV, M)$	45: $\text{TAG}' \leftarrow \text{PMAC2x}[\tilde{E}_K](A, M)$
16: return (C, TAG)	46: if $\text{TAG} = \text{TAG}'$ then
21: function $\text{CONV}'(U)$	47: return M
22: return $T \leftarrow U[1..t]$	48: return \perp
31: function $\text{IVCTRT}[\tilde{E}_K](IV, M)$	51: function $\text{PMAC2x}[\tilde{E}_K](A, M)$
32: $(T, V) \leftarrow IV$	52: $(X_a, Y_a) \leftarrow \text{HASHx}[\tilde{E}_K]^{4,5}(A)$
33: $(M_1, \dots, M_m) \xleftarrow{n} M$	53: $(X_m, Y_m) \leftarrow \text{HASHx}[\tilde{E}_K]^{6,7}(M)$
34: for $i \leftarrow 1$ to $m - 1$ do	54: $\hat{X}_m \leftarrow \text{CONV}(X_m \oplus X_a \oplus X_0)$
35: $C_i \leftarrow \tilde{E}_K^{1, T+(i-1)}(V) \oplus M_i$	55: $\hat{Y}_m \leftarrow \text{CONV}(Y_m \oplus Y_a \oplus Y_0)$
36: $S_m \leftarrow \tilde{E}_K^{1, T+(m-1)}(V)[1.. M_m]$	56: $U \leftarrow \tilde{E}_K^{2, \hat{Y}_m}(X_m)$
37: $C_m \leftarrow S_m \oplus M_m$	57: $V \leftarrow \tilde{E}_K^{3, \hat{X}_m}(Y_m)$
38: return $C \leftarrow (C_1 \parallel \dots \parallel C_m)$	58: return $IV \leftarrow (U \parallel V)$

Theorem 3 (DAE Security of SIVx). Let $F : \mathcal{K}_1 \times \mathcal{A} \times \mathcal{M} \rightarrow \{0, 1\}^{2n}$, and let $\Pi = (\tilde{\mathcal{E}}, \tilde{\mathcal{D}})$ be an IV-based encryption scheme with key space \mathcal{K}_2 and IV space \mathcal{IV} . Let $K_1 \leftarrow \mathcal{K}_1$ and $K_2 \leftarrow \mathcal{K}_2$ be independent. Let $\text{CONV}' : \{0, 1\}^n \rightarrow \mathcal{IV}$ be a regular function. Let \mathbf{A} be a DAE adversary running in time at most θ , asking at most q queries of at most $8 \leq \ell < 2^t$ blocks in total. Then, it holds that

$$\text{Adv}_{\text{SIVx}[F, \Pi]}^{\text{DAE}}(\mathbf{A}) \leq \text{Adv}_{\Pi}^{\text{IVe}}(\theta + O(\ell), q, \ell) + \text{Adv}_F^{\text{PRF}}(\theta + O(\ell), q, \ell) + \frac{q}{2^n}.$$

We defer the proof of Theorem 3 to the full version of this work [11]. Inserting the bounds from Theorems 1 and 2, we obtain the corollary below, where F denotes $\text{PMAC2x}[\tilde{E}]$ and Π represents $\text{IVCTRT}[\tilde{E}]$.

Corollary 2. Fix positive integers k, n, t and $d = 4$. Define $d + t = n$ and let $\mathcal{T} = \{0, 1\}^t$ and $\mathcal{T}' = \{0, 1\}^d \times \{0, 1\}^t$, and $\mathcal{IV} = \{0, 1\}^{n-1}$. Let $\tilde{E} : \mathcal{K} \times \mathcal{T}' \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, and $\text{CONV}' : \{0, 1\}^n \rightarrow \mathcal{IV}$ be a regular function. Let $K \leftarrow \mathcal{K}$ and \mathbf{A} be a DAE adversary that runs in time at most θ , and asks at most q queries of at most $8 \leq \ell < 2^t$ blocks in total. Then

$$\text{Adv}_{\text{SIVx}[\tilde{E}]}^{\text{DAE}}(\mathbf{A}) \leq \frac{2^{2d}q^2}{2 \cdot (2^n - q)^2} + \frac{2^d q^3}{3 \cdot 2^{2n}(2^n - q)} + \frac{2^d q^2}{2^n(2^n - q)} + \frac{4\ell \log q + 1}{2^{n-1}} + \frac{q + 1 + \ell \log^2(\ell)}{2^n} + \text{Adv}_E^{\text{TPRP}}(\theta + O(2\ell + 2q), 2\ell + 2q).$$

7 Conclusion

This work revisited the PMAC_TBC1K construction by Naito for constructing a MAC with beyond-birthday-bound (BBB) security and $2n$ -bit outputs, called PMAC2x. We identified a critical assumption in the previous analysis of PMAC_TBC1K and circumvented it by a new proof for PMAC2x; moreover, we could easily derive a proof for PMACx, a variant of our PMAC2x construction with n -bit outputs. So, we also provided a corrected bound for Naito's construction. We obtained the positive result that all three constructions provide PRF security for up to $O(q^2/2^{2n} + q^3/2^{3n})$ queries. With the help of PMAC2x, we constructed a BBB-secure AE scheme from a tweakable block cipher whose security is independent of nonces and which depends on a single primitive under a single key. We are aware that the $2n$ -bit tag of SIVx requires still as many bits to be transmitted as for the $2n$ -bit nonce-IV in SCT; future work could study how an appropriate truncation could reduce the transmission overhead while retaining BBB security.

Acknowledgments. The authors would like to thank Yusuke Naito and the anonymous reviewers for fruitful comments that helped improve our work.

References

1. Andreeva, E., Bogdanov, A., Datta, N., Luykx, A., Mennink, B., Nandi, M., Tischhauser, E., Yasuda, K.: COLM v1 (2016). Submission to the CAESAR competition. <http://competitions.cr.yp.to/caesar-submissions.html>
2. Bellare, M., Guérin, R., Rogaway, P.: XOR MACs: new methods for message authentication using finite pseudorandom functions. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 15–28. Springer, Heidelberg (1995). doi:10.1007/3-540-44750-4_2
3. Bellare, M., Rogaway, P.: Code-based game-playing proofs and the security of triple encryption. IACR Cryptology ePrint Archive, 2004:331 (2004)
4. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006). doi:10.1007/11761679_25
5. Bernstein, D.J.: How to stretch random functions: the security of protected counter sums. J. Crypt. **12**(3), 185–192 (1999)
6. Black, J., Rogaway, P.: A block-cipher mode of operation for parallelizable message authentication. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 384–397. Springer, Heidelberg (2002). doi:10.1007/3-540-46035-7_25
7. Datta, N., Dutta, A., Nandi, M., Paul, G., Zhang, L.: Building single-key beyond birthday bound message authentication code. IACR Cryptology ePrint Archive, 2015/958 (2015)
8. Fleischmann, E., Forler, C., Lucks, S.: McOE: a family of almost foolproof on-line authenticated encryption schemes. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 196–215. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34047-5_12

9. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 15–44. Springer, Heidelberg (2015). doi:[10.1007/978-3-662-46800-5_2](https://doi.org/10.1007/978-3-662-46800-5_2)
10. Jean, J., Nikolić, I., Peyrin, T.: Deoxys v1.4 (2016). Third-round submission to the CAESAR competition. <http://competitions.cr.yt.to/caesar-submissions.html>
11. List, E., Nandi, M.: Revisiting Full-PRF-Secure PMAC and using it for beyond-birthday authenticated encryption. Cryptology ePrint Archive (2016, to appear)
12. Luykx, A., Preneel, B., Szepieniec, A., Yasuda, K.: On the influence of message length in PMAC’s security bounds. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 596–621. Springer, Heidelberg (2016). doi:[10.1007/978-3-662-49890-3_23](https://doi.org/10.1007/978-3-662-49890-3_23)
13. Luykx, A., Preneel, B., Tischhauser, E., Yasuda, K.: A MAC mode for lightweight block ciphers. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 43–59. Springer, Heidelberg (2016). doi:[10.1007/978-3-662-52993-5_3](https://doi.org/10.1007/978-3-662-52993-5_3)
14. Mandal, A., Nandi, M.: An improved collision probability for CBC-MAC and PMAC. IACR Cryptology ePrint Archive, 2007:32 (2007)
15. Minematsu, K.: Parallelizable rate-1 authenticated encryption from pseudorandom functions. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 275–292. Springer, Berlin (2014). doi:[10.1007/978-3-642-55220-5_16](https://doi.org/10.1007/978-3-642-55220-5_16)
16. Minematsu, K., Matsushima, T.: New bounds for PMAC, TMAC, and XCBC. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 434–451. Springer, Berlin (2007). doi:[10.1007/978-3-540-74619-5_27](https://doi.org/10.1007/978-3-540-74619-5_27)
17. Naito, Y.: Full PRF-secure message authentication code based on tweakable block cipher. In: Au, M.-H., Miyaji, A. (eds.) ProvSec 2015. LNCS, vol. 9451, pp. 167–182. Springer, Cham (2015). doi:[10.1007/978-3-319-26059-4_9](https://doi.org/10.1007/978-3-319-26059-4_9)
18. Nandi, M.: A unified method for improving PRF bounds for a class of blockcipher based MACs. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 212–229. Springer, Berlin (2010). doi:[10.1007/978-3-642-13858-4_12](https://doi.org/10.1007/978-3-642-13858-4_12)
19. Nandi, M., Mandal, A.: Improved security analysis of PMAC. J. Math. Crypt. **2**(2), 149–162 (2008)
20. Peyrin, T., Seurin, Y.: Counter-in-tweak: authenticated encryption modes for tweakable block ciphers. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 33–63. Springer, Berlin (2016). doi:[10.1007/978-3-662-53018-4_2](https://doi.org/10.1007/978-3-662-53018-4_2)
21. Peyrin, T., Seurin, Y.: Counter-in-tweak: authenticated encryption modes for tweakable block ciphers. IACR Cryptology ePrint Archive, 2015:1049, Version, 27 May 2016
22. Rogaway, P.: Authenticated-encryption with associated-data. In: ACM Conference on Computer and Communications Security, pp. 98–107 (2002)
23. 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, Berlin (2004). doi:[10.1007/978-3-540-30539-2_2](https://doi.org/10.1007/978-3-540-30539-2_2)
24. Rogaway, P.: Nonce-based symmetric encryption. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 348–358. Springer, Berlin (2004). doi:[10.1007/978-3-540-25937-4_22](https://doi.org/10.1007/978-3-540-25937-4_22)
25. Yasuda, K.: A new variant of PMAC: beyond the birthday bound. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 596–609. Springer, Berlin (2011). doi:[10.1007/978-3-642-22792-9_34](https://doi.org/10.1007/978-3-642-22792-9_34)

26. Yasuda, K.: PMAC with parity: minimizing the query-length influence. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 203–214. Springer, Berlin (2012). doi:[10.1007/978-3-642-27954-6_13](https://doi.org/10.1007/978-3-642-27954-6_13)
27. Zhang, Y.: Using an error-correction code for fast, beyond-birthday-bound authentication. In: Nyberg, K. (ed.) CT-RSA 2015. LNCS, vol. 9048, pp. 291–307. Springer, Cham (2015). doi:[10.1007/978-3-319-16715-2_16](https://doi.org/10.1007/978-3-319-16715-2_16)