Cryptogr. Commun. (2018) 10:731–753
https://doi.org/10.1007/s12095-017-0275-0

On rate-1 and beyond-the-birthday bound secure online ciphers using tweakable block ciphers

Ashwin Jha[1] · **Mridul Nandi**[1]

Received: 30 July 2017 / Accepted: 10 December 2017 / Published online: 6 January 2018

Abstract Recently, Andreeva et al. showed that online ciphers are actually equivalent to arbitrary tweak length (ATL) tweakable block ciphers (TBCs). Within this result they gave a security preserving generic conversion from ATL TBCs to online ciphers. XTX by Minematsu and Iwata is a nice way of extending the tweak space of any fixed tweak length (FTL) TBC using a pAXU hash function. By combining the previous two methods one can get a FTL TBC based online cipher with security in the order of $\sigma^2\varepsilon$ where σ is the total number of blocks in all queries, and ε is the pAXU bound of the underlying hash function. In this paper we show that there are genuine practical issues which render it almost impossible to get full security using this approach. We then observe that a recent online enciphering scheme called POEx by Forler et al. is actually an implicit example of this approach. We show a flaw in the analysis of POEx which results in a birthday bound attack and invalidates the beyond-the-birthday bound OSPRP security claim. We take a slightly different approach then the one just mentioned and propose XTC which achieves OSPRP security of $O(\max(n\sigma 2^{-n}, \sigma^2 2^{-(n+t)}))$ where t is the tweak size and n is the block size. While doing so we present an impossibility result for $t > n$ which can be of independent interest.

Keywords Provable security · Online cipher · Tweakable block cipher · POEx · XTX · XTC

This article is part of the Topical Collection on *Special Issue on Statistics in Design and Analysis of Symmetric Ciphers*

✉ Ashwin Jha
ashwin.jha1991@gmail.com

Mridul Nandi
mridul.nandi@gmail.com

[1] Applied Statistics Unit, Indian Statistical Institute, Kolkata, India

Mathematics Subject Classification (2010) 94A60

1 Introduction

Online Ciphers In low-memory devices and continuous data streaming platforms, it is often desirable to produce encrypted data in online fashion, i.e. the encryption of the current data block should only depend on the previous data blocks. In such scenarios the online property is desirable. Paraphrasing the informal definition by Rogaway and Zhang [32], an encryption scheme is said to satisfy the *online property* if (1) it can be realized by an algorithm that, for any input, read its input blocks one at a time in order and computes the corresponding output blocks one at a time in order, and (2) it uses only a constant-bound amount of memory and/or latency. The introductory definition by Bellare et al. [6] satisfied (1), which was later strengthened by Boldyreva and Taesombut [12] to satisfy (2).

Symmetric-key schemes are inherently probabilistic in the sense that the security relies on the randomness of the secret key. The language of provable security itself relies heavily on the tools and concepts from the field of probability theory. For instance the usual notion of security for a general enciphering scheme, as defined by Luby and Rackoff [25], requires it to be *indistinguishable* (have a negligible computational distance, or statistical distance in information-theoretic settings) from a uniform random permutation. In other words, it should be a good pseudorandom permutation or PRP. It is trivial to see that online ciphers are not PRPs. For an ideally secure cipher every bit of the ciphertext should depend on every bit of the plaintext and vice-versa - a requirement which invalidates the online property. The appropriate security notions of online ciphers are online PRP (OPRP) and online SPRP (OSPRP) [6], which require indistinguishability from a uniform online random permutation. But even with this relaxed notion of online random permutations, there are scenarios where an ideal online (S)PRP can be completely vulnerable. Hoang et al. [22] demonstrated such a vulnerability through their chosen-prefix secret-suffix (CPSS) attack. Such attacks are impossible for an ideal PRP.

Although online ciphers satisfy a relaxed security notion which does not serve all scenarios, they are highly valuable in practice, for they allow single-pass and on the fly encryption of plaintexts. This is of particular interest in applications with high-throughput demands and low memory buffer such as the OpenSSL `EVP_DecryptUpdate` interface as noticed in [18]. Most prominently online ciphers are used in authenticated encryption to dilute the effect of nonce misuses [1, 4, 13, 16]. Amanatidis et al. [2] used online ciphers to solve a database-security problem. Recently, Andreeva et al. [3] showed yet another application for online ciphers, as a primitive to build offline (usual) cipher.

(Tweakable) Block Cipher Based Online Ciphers. In their foundational paper Bellare et al. [6] proposed CBC like constructions, viz. HCBC1 and HCBC2, both of which employed one call to block cipher and one call to an almost XOR universal (AXU) hash function. HCBC1 and HCBC2 were shown to have birthday bound OPRP security, and birthday bound OSPRP security, respectively. HPCBC, proposed by Boldyreva and Taesombut [12], was a variant of HCBC2 that prepends the encryption of a random IV in order to fit a stronger notion. Nandi [28, 29] proposed simpler proofs for HCBC1 and HCBC2, and gave two improved schemes called MHCBC and MCBC. Among these MCBC has the feature of replacing the call to a hash function by a second call to the block cipher.

In [32] Rogaway and Zhang proposed three schemes, namely TC1, TC2, and TC3, based on tweakable block ciphers (or TBCs) [24]. These schemes exploited the additional tweak input of TBCs to eliminate the additional calls to hash function/block cipher. In an independent

work Fleischmann et al. [17] presented a scheme called McOE-G which is similar to TC3, albeit with a more practical handling of arbitrary length inputs.

For a variant of online property called diblock-online, Bhaumik and Nandi [10] gave an inverse-free construction called OleF, that achieved diblock-online SPRP security. Apart from these, several online ciphers were also proposed within full fledged authenticated encryption schemes [1, 4, 5, 13, 16]. Recently, Forler et al. [18] proposed an online cipher POEx that claims beyond-the-birthday bound (BBB) security.

1.1 Motivation

A recent work of Andreeva et al. [3] shows that online ciphers are equivalent to arbitrary tweak length (ATL) TBCs. Although this result is more of theoretical interests, the TBC to online cipher converter can be coupled with XTX [27] to get an efficient online cipher (see Section 3). As we see later, a naive approach in using this converter will result in a loss in factor of ℓ (maximum permissible length of any message). Forler et al. [18] implicitly used this converter in POEx. But we show a critical flaw in their analysis which invalidates the security claim of POEx. Apparently avoiding some loss in the security is a non-trivial task.

We would like to note here that getting an optimally secure and efficient online cipher based on normal block ciphers is almost impossible in standard model. A recent result by Mennink [26] shows that it is impossible to get optimally secure TBCs via block ciphers in the standard model. This can be easily extended over to online ciphers using the equivalence result by Andreeva et al. [3]. In light of these recent works, it would be interesting to explore the possibility of an (almost) optimally secure and efficient (in some sense) online cipher based on TBCs.

1.2 Our contribution

In the following discussion, t and n denote the tweak and block size, respectively. Our contributions are threefold. First, as a prolog to our later analysis, in Section 3 we derive a practical instantiation of Andreeva et al.'s [3] TBC to online cipher converter using XTX [27], which is implicitly used in [18]. We discuss its limitation in achieving efficiency and optimal security at the same time. Second, in Section 4 we show a flaw in the security analysis of POEx (see Section 4). We exploit this flaw to get a birthday attack on POEx which invalidates the security claim. Third, in Section 5 we explore the possibility of a more efficient way of realizing the generic TBC to online cipher converter [3]. We show an impossibility result which implies that the task is hard for $t > n$. We then propose XTC, an update over POEx that achieves $\min(2^n/n, 2^{(n+t)/2})$ block-queries security using a fixed tweak length (FTL) TBC (see Section 5). Table 1 compares XTC with existing online ciphers (both standalone and encryption phase of online misuse resistant AE schemes).

Our attack on POEx and the impossibility result use certain special features of the probability distributions of random mappings over ranked nodes and online ciphers, respectively. In case of the attack on POEx we show that random mappings over ranked nodes converge to a single fixed value when iterated. In case of the impossibility result we show that it is much more probable to get collisions of special types on online ciphers as compared to a general cipher. These results are examples in a long list of other such instances in literature which use statistical and probabilistic tools in symmetric-key provable security.

Table 1 Comparison of existing O(S)PRP-secure online ciphers and online nonce misuse-resistant AE schemes with our proposal

Feature	Online ciphers												Online AE					
	XTC (Section 5)	**POEx** [18]	POE [1]	COPE [4]	**TC3** [32]	**TC2** [32]	**TC1** [32]	MCBC [29]	MHCBC [29]	HPCBC [12]	HCBC2 [6]	HCBC1 [6]	OleF [10]	ElMD [13]	ElME [16]	COBRA [5]	**McOE-G** [17]	**McOE-X** [17]
#(T)BC	ℓ	ℓ	ℓ	2ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	$\ell+1$	ℓ	ℓ	2ℓ	2ℓ	2ℓ	ℓ	ℓ	ℓ
#Hashing	2ℓ	2ℓ	2ℓ	–	–	–	–	ℓ	2ℓ	$2\ell+1$	2ℓ	ℓ	–	–	–	ℓ	ℓ	–
Key Length	$2n+k$	$n+t+k$	$n+k$	k	k	k	k	k	$n+k$	$2n+k$	$2n+k$	$n+k$	k	k	k	k	$n+k$	k
OSPRP	✓	✓	✓	–	✓	✓	–	✓	✓	✓	✓	–	✓	✓	✓	–	–	–
BBB	✓	$*$	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Schemes based on tweakable block ciphers are in boldface. The data is given for encrypting an ℓ-blocks message. k, n and t denote the key size, block size and tweak size (wherever applicable) of the underlying (T)BC. #Hashing denotes the number of $\mathbb{F}_{2^n}$ multiplications per block assuming $t = n$. ✓/– means the given scheme provides/lacks the respective feature. $*$ means the claimed security of POEx is BBB. See Section 4 for the actual security of POEx

2 Preliminaries

2.1 Basic notations and conventions

Throughout this paper we fix n and t as positive integers denoting block and tweak size. We sometime use the adjective *fresh* for a variable which is distinct from all the previous variables. We write $x \dashv y$ to denote "x such that y". For integers $a \leq b$, we let $[a..b] = \{a, \ldots, b\}$ or simply $[b]$ when $a = 1$.

If x is a vector (or a sequence or a string) then $|x|$ denotes its length, and x_i denotes its i-th coordinate. We let $\perp$ denote the empty vector, which has length 0. If $0 \leq i \leq |x|$, then we let $x_{\leq i} = (x_1, \ldots, x_i)$, this being $\perp$ when $i = 0$. For any two strings x and y over a set $\mathscr{S}$, $x \| y$ denotes the concatenation of x and y. If $z = x \| y$, then x and y are called the prefix and suffix, respectively of z.

Within oracle interactions, we write the i-th query instance of some variable X by X^i. Accordingly the j-th coordinate of the i-th query instance of X is written as X^i_j.

For any set $\mathscr{S}$, we let $\mathscr{S}^i$ denote the set of all i length vectors over $\mathscr{S}$. We let $\mathscr{S}^{\leq \ell} = \cup_{i=1}^{\ell} \mathscr{S}^i$, $\mathscr{S}^+ = \cup_{i=1}^{\infty} \mathscr{S}^i$, and $\mathscr{S}^* = \cup_{i=0}^{\infty} \mathscr{S}^i$. We let $\mathscr{B} := \{0,1\}^n$ and $\mathscr{T} := \{0,1\}^t$, where the elements of $\{0,1\}$, $\mathscr{B}$, and $\mathscr{T}$ are called *bits*, *blocks*, and *tweaks* respectively. Some times we also view blocks as integers in $\left[0..2^n - 1\right]$. For a set $\mathscr{S}$, $X \leftarrow_{\$} \mathscr{S}$ denotes the uniform random sampling of X from $\mathscr{S}$.

2.2 Universal hashing

A hash function family H is a $(\mathscr{K}, \mathscr{D}, \mathscr{R})$-family of functions $\{H_k := H(k, \cdot) : \mathscr{D} \to \mathscr{R}\}_{k \in \mathscr{K}}$ defined over its *domain or message space* $\mathscr{D}$, *digest or hash space* $\mathscr{R}$ and indexed by the *key space* $\mathscr{K}$.

Definition 1 (ε-almost-XOR-universal hash function) A $(\mathscr{K}, \mathscr{D}, \mathscr{B})$-hash family H is called *ε-almost-XOR Universal* (or AXU), if for any two distinct x and x' in $\mathscr{D}$ and a $\delta \in \mathscr{B}$, the *differential probability* is at most ε. In other words, we have

$$\mathsf{diff}_H := \max_{x \neq x', \delta} \Pr_K[H_K(x) \oplus H_K(x') = \delta] \leq \varepsilon,$$

where the random variable K is uniformly distributed over the set $\mathscr{K}$.

Multi-linear hash [21, 33] and PDP hash [11, 21, 23, 37] are some examples of AXU hash functions.

When $\delta = 0$, the 0-differential event is equivalent to collision. So we rewrite diff_H as coll_H, and we call it the maximum *collision probability*.

Definition 2 (ε-almost-universal hash function) A $(\mathscr{K}, \mathscr{D}, \mathscr{B})$-hash family H is called *ε-almost universal* (or ε-AU) if

$$\mathsf{coll}_H := \max_{x \neq x'} \mathsf{Pr}_K[H_K(x) = H_K(x')] \leq \varepsilon.$$

Almost (XOR) universal hash functions were described by Wegman and Carter [14, 36] and Gilbert et al. [20], followed by investigations by Stinson [34, 35].

Minematsu and Iwata [27] introduced the notion of ε-partial-almost-XOR-universal hash functions to capture simultaneous collision and differential events on two distinct parts of any hash digest. This will be useful for our later analysis of POEx and XTC.

Definition 3 (ε-partial-almost-XOR-universal hash function) A $(\mathscr{K}, \mathscr{D}, \mathscr{T} \times \mathscr{B})$-hash family H is called *ε-partial-almost-XOR-universal* (or ε-pAXU) if

$$\mathsf{coll\text{-}diff}_H := \max_{x \neq x', \delta} \mathsf{Pr}_K[H_K(x) \oplus H_K(x') = (0, \delta)] \leq \varepsilon.$$

We accumulate some easily verifiable properties of AXU hash functions in Proposition 1. These properties will aid our later work.

Proposition 1 *For A(X)U hash functions we have*

1. *Any ε-AXU hash function is an ε-AU hash function.*
2. *The concatenation $H_1 \| H_2$ of an ε_1-AU hash H_1, and an ε_2-AXU hash H_2 is an $\varepsilon_1\varepsilon_2$-pAXU hash function when the hash keys for H_1 and H_2 are sampled independently.*
3. *The m-bit truncation of an ε-AXU hash function is $(2^{n-m} \cdot \varepsilon)$-AU hash function.*

2.3 Adversaries and advantage

An adversary $\mathbf{A}$ is an efficient Turing machine that interacts with a given set of oracles in black box fashion. For an oracle $\mathscr{O}$, $\mathbf{A}^{\mathscr{O}}$ denotes $\mathbf{A}$'s interaction with $\mathscr{O}$. For an oracle $\mathscr{O}$, $\mathscr{O}^{\pm}$ denotes the bidirectional access to the underlying function and its inverse. In this work we always see $\mathbf{A}$ as a distinguisher which tries to detect the output distribution of some oracle and outputs a single bit after its interaction with the given oracle(s). Without loss of generality, we assume that $\mathbf{A}$ never asks queries to which it already knows the answer. We write $\mathbf{Exp}^{\mathsf{dist}}_{\mathbf{A},\mathscr{O}}$ for the random experiment (a Bernoulli trial) that runs a DIST-adversary $\mathbf{A}$ with the oracle $\mathscr{O}$, where DIST denotes a distinguishing model like PRP, SPRP, PRF etc. We write

$$\mathbf{Adv}^{\mathsf{dist}}_{\mathscr{O}_1;\mathscr{O}_2}(\mathbf{A}) := \left|\Pr\left[\mathbf{Exp}^{\mathsf{dist}}_{\mathbf{A},\mathscr{O}_1} \Rightarrow 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{dist}}_{\mathbf{A},\mathscr{O}_2} \Rightarrow 1\right]\right|$$

to denote the DIST advantage of $\mathbf{A}$ in distinguishing $\mathscr{O}_1$ from $\mathscr{O}_2$. All probabilities are defined over the random coins of the oracles and those of the adversary, if any. In general $\mathscr{O}_2$ will be clear from DIST's context, hence we will drop it from the subscript. We will provide code-based descriptions of the oracles, called games, according to the game-playing framework by Bellare and Rogaway [7, 8].

2.4 Tweakable block cipher

A *tweakable block cipher* $\widetilde{E}$ with associated key space $\mathscr{K}$, tweak space $\mathscr{T}$, and message space $\mathscr{B}$ is a mapping $\widetilde{E} : \mathscr{K} \times \mathscr{T} \times \mathscr{B} \to \mathscr{B}$ such that for every key $K \in \mathscr{K}$ and tweak $T \in \mathscr{T}$, it holds that $\widetilde{E}^T_K(\cdot) := \widetilde{E}_K(T, \cdot) := \widetilde{E}(K, T, \cdot)$ is a permutation over $\mathscr{B}$. We let $\mathsf{Tperm}(\mathscr{T}, \mathscr{B})$ to be the set of all tweakable permutations over $\mathscr{B}$ with tweak space $\mathscr{T}$, and $\widetilde{\Pi} \leftarrow_{\$} \mathsf{Tperm}(\mathscr{T}, \mathscr{B})$. Let $\mathbf{Exp}^{\mathsf{tsprp}}_{\mathbf{A},\mathscr{O}}$ be an experiment which outcomes 1 if $\mathbf{A}^{\mathscr{O}}$ outputs 1, and 0 otherwise, where $\mathbf{A}$ is a *Tweakable Strong Pseudorandom Permutation* (TSPRP)-adversary. The TSPRP advantage of $\mathbf{A}$ against $\widetilde{E}$ is defined as

$$\mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}}(\mathbf{A}) := \left|\Pr_{K \leftarrow_{\$} \mathscr{K}}\left[\mathbf{Exp}^{\mathsf{tsprp}}_{\mathbf{A},\widetilde{E}^{\pm}_K} \Rightarrow 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{tsprp}}_{\mathbf{A},\widetilde{\Pi}^{\pm}} \Rightarrow 1\right]\right|.$$

Let $\mathbb{A}(q, \tau)$ be the class of all adversaries with runtime at most τ, and number of queries at most q. For $\varepsilon \in [0, 1]$, $\widetilde{E}$ is called a (q, τ, ε)-TSPRP if and only if,

$$\mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}}(q, \tau) := \max_{\mathbf{A} \in \mathbb{A}(q,\tau)} \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}}(\mathbf{A}) \leq \varepsilon.$$

2.5 Online cipher

An *online cipher* $\mathbf{O}$ with associated key space $\mathscr{K}$ and message space $\mathscr{B}^*$ is a mapping $\mathbf{O} : \mathscr{K} \times \mathscr{B}^* \to \mathscr{B}^*$ such that for every key $K \in \mathscr{K}$, $\mathbf{O}_K(\cdot) := \mathbf{O}(K, \cdot)$ is a length-preserving permutation over $\mathscr{B}^*$ and $\mathbf{O}_K(X)$ is a prefix of $\mathbf{O}_K(Y)$ if and only if X is a prefix of Y. We let $\mathsf{Operm}(\mathscr{B}^*)$ to be the set of all online permutations over $\mathscr{B}^*$, and $\overrightarrow{\Pi} \leftarrow_{\$} \mathsf{Operm}(\mathscr{B}^*)$. Let $\mathbf{Exp}^{\mathsf{osprp}}_{\mathbf{A},\mathscr{O}}$ be an experiment which outcomes 1 if $\mathbf{A}^{\mathscr{O}}$ outputs 1, and 0 otherwise, where $\mathbf{A}$ is an *Online Strong Pseudorandom Permutation* (OSPRP)-adversary. The OSPRP advantage of $\mathbf{A}$ against $\mathbf{O}$ is defined as

$$\mathbf{Adv}^{\mathsf{osprp}}_{\mathbf{O}}(\mathbf{A}) := \left| \Pr_{K \leftarrow_{\$} \mathscr{K}} \left[\mathbf{Exp}^{\mathsf{osprp}}_{\mathbf{A},\mathbf{O}^{\pm}_K} \Rightarrow 1 \right] - \Pr \left[\mathbf{Exp}^{\mathsf{osprp}}_{\mathbf{A},\overrightarrow{\Pi}^{\pm}} \Rightarrow 1 \right] \right|.$$

Let $\mathbb{A}(q, \ell, \sigma, \tau)$ be the class of all adversaries with runtime at most τ, number of queries at most q, maximum query length at most ℓ, and the total length over all queries at most σ (also referred as block-queries). For $\varepsilon \in [0, 1]$, $\mathbf{O}$ is called a $(q, \ell, \sigma, \tau, \varepsilon)$-OSPRP if and only if,

$$\mathbf{Adv}^{\mathsf{osprp}}_{\mathbf{O}}(q, \ell, \sigma, \tau) := \max_{\mathbf{A} \in \mathbb{A}(q,\ell,\sigma,\tau)} \mathbf{Adv}^{\mathsf{osprp}}_{\mathbf{O}}(\mathbf{A}) \leq \varepsilon.$$

In this work we will generally consider computationally unbounded adversaries. Without loss of generality, it suffices to only focus on deterministic adversaries, as for any probabilistic adversary there exists a deterministic one with at least the same advantage, and we will assume so henceforth.

3 Toward a general design strategy for BBB online ciphers using TBCs

Rate of any cryptographic scheme is defined as the ratio of the number of blocks in any input to the number of primitive (TBC in this case) calls for that input. In most of the cases rate is independent of the input. A rate ≥ 1 scheme will, in general, be more efficient then a rate $\frac{1}{2}$ scheme when identical primitives are employed. From now onwards, we use rate and number of field multiplications as our parameters for efficiency.

This section serves as a prolog for later analysis of POEx and XTC. In this section we discuss the equivalence between online ciphers and TBCs shown by Andreeva et al. [3]. We also demonstrate how a simple practical instantiation can be given by using XTX. We then observe how a combination of practical (read efficiency) issues render it non-trivial to convert the theoretical construction of [3] in practice.

3.1 The iterated TBC view of online ciphers

In a recent work [3] on a generic construction of offline ciphers using online ciphers as primitives, Andreeva et al. made the following useful observation:

an ideal online cipher is equivalent to an ideal ATL TBC.

We reproduce their result in Theorem 1.

Theorem 1 ([3, Theorem 1]) *There is a security preserving one-to-one correspondence between online ciphers on $\mathscr{B}^+$ and tweakable block ciphers on $\mathscr{B}$ with tweak space $\mathscr{B}^*$.*

While a more detailed proof is available in [3], we describe their generic construction of online cipher based on a tweakable block cipher with arbitrary tweak space. Let $\widetilde{\mathbf{E}}_K$ be a TBC with tweak space $\mathscr{B}^*$ and block space $\mathscr{B}$. We define an online cipher $\mathbf{O}[\widetilde{\mathbf{E}}]$ over $\mathscr{B}^+$ as $\forall \ell \geq 1, \forall x := (x_1, \ldots, x_\ell) \in \mathscr{B}^\ell$,

$$\mathbf{O}[\widetilde{\mathbf{E}}](x) := \widetilde{\mathbf{E}}_K^{\mu}(x_1) \| \widetilde{\mathbf{E}}_K^{x_1}(x_2) \| \cdots \| \widetilde{\mathbf{E}}_K^{x_{<i}}(x_i) \| \cdots \| \widetilde{\mathbf{E}}_K^{x_{<\ell}}(x_\ell).$$

One can easily verify that,

$$\mathbf{Adv}^{\mathsf{osprp}}_{\mathbf{O}[\widetilde{\mathbf{E}}]}(q, \sigma, \tau) \leq \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{\mathbf{E}}}(\sigma, \tau') \tag{1}$$

where $\tau' = \tau + O(\sigma)$. Specifically when we replace $\widetilde{\mathbf{E}}$ with an ideal ATL tweakable random permutation $\widetilde{\boldsymbol{\Pi}}$ we get $\mathbf{Adv}^{\mathsf{osprp}}_{\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]}(q, \sigma, \tau) = 0$. A minor variant of the above mentioned construction may also consider the previous ciphertext blocks along with the plaintext blocks, i.e., tweak for the i-th block is $(x_{<i}, \mathbf{O}[\widetilde{\mathbf{E}}](x_{<i}))$, and the modified definition is,

$$\mathbf{O}[\widetilde{\mathbf{E}}](x) := \widetilde{\mathbf{E}}_K^{\mu}(x_1) \| \widetilde{\mathbf{E}}_K^{(x_1, \mathbf{O}[\widetilde{\mathbf{E}}](x_1))}(x_2) \| \cdots \| \widetilde{\mathbf{E}}_K^{(x_{<i}, \mathbf{O}[\widetilde{\mathbf{E}}](x_{<i}))}(x_i) \| \cdots \| \widetilde{\mathbf{E}}_K^{(x_{<\ell}, \mathbf{O}[\widetilde{\mathbf{E}}](x_{<\ell}))}(x_\ell).$$

It is not hard to see that this does not give any added security advantage. But the utility of this modification will become more apparent as we proceed. Thus any online cipher can be viewed as a chain of iterated TBC. We call this equivalent view of online ciphers, *the iterated TBC view of online cipher*.

3.2 O[XTX]: Moving from Theory to Practice

Given the strong security guarantee of (1) and rate 1 construction, it is only natural to look for practical instantiations. The next immediate question is: how can we instantiate an ATL TBC based on an FTL TBC.

XTX [27] by Minematsu and Iwata is an elegant way of extending the tweak length of an FTL TBC. At the highest level, XTX employs a pAXU hash which takes a tweak value of arbitrary length as input and computes a fixed length tweak and input/output masking for the underlying FTL TBC. Formally, let $H : \mathscr{B}^* \to \mathscr{T} \times \mathscr{B}$ be an ε-pAXU. The hash output $H(T)$ is parsed into $H_{\mathrm{twk}}(T) \| H_{\mathrm{msk}}(T)$. Using this pAXU and a TBC $\widetilde{E}$ over the tweak space $\mathscr{T}$, XTX is defined as

$$\mathsf{XTX}^T_K(x) = \widetilde{E}_K^{H_{\mathrm{twk}}(T)}(x \oplus H_{\mathrm{msk}}(T)) \oplus H_{\mathrm{msk}}(T).$$

In [27], Minematsu and Iwata have shown the following upper bound on the TSPRP advantage of XTX.

Theorem 2 $\mathbf{Adv}^{\mathsf{tsprp}}_{XTX}(\sigma, \tau') \leq \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}}(\sigma, \tau'') + \varepsilon\sigma^2$, *where* $\tau'' = \tau' + \mathsf{Time}_H \times O(\sigma)$.

By using (1) and Theorem 2, we obtain a construction **O**[XTX] with OSPRP advantage at most $\mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}}(\sigma, \tau'') + \varepsilon\sigma^2$. At this point we would like to emphasize that POEx is an implicit example of **O**[XTX]. Although **O**[XTX] is simple (both in description and security proof), it requires very strong bound on ε, close to $2^{-(n+t)}$. Now we describe how this becomes an issue when coupled with the need for efficiency.

Security degradation and Hash Key. Note that $H : \mathscr{B}^* \to \{0,1\}^{n+t}$ is required to be ε-pAXU for some ε. If we keep hash key size to be $n+t$ then $\varepsilon' \geq \ell'/2^{n+t}$ where $\ell' = \ell n/(n+t)$ (the number of $(n+t)$-bits present in ℓ block inputs). We can achieve this bound by considering polyhash defined over $(n+t)$-bit binary field. If we plug this ε we obtain a bound of the form $\ell\sigma^2/2^{n+t}$. To get rid of the ℓ factor we need to apply hash functions with larger hash key such as Pseudo dot product [11, 21, 37], multi-linear hash [14, 20, 33], and EHC [30], which might be practically infeasible. This raises the following ambitious question.

(a) *Can we devise some method to reduce the hash key size to a constant factor of $n+t$, while avoiding the ℓ factor in the security bound?*

3.2.1 Toward a possible remedy

One possible way of solving (a) could be to use a fixed number of previous blocks information instead of the entire history of previous blocks. This will certainly replace the ℓ factor in case of poly-hash. But now we have to use both plaintext and ciphertext pairs, otherwise the overall design will no longer be secure (the adversary can always keep the required plaintext blocks fixed). Further a direct security reduction like **O**[XTX] will not be possible any more and an independent security analysis is required. In Section 5 we use this design strategy to give an (almost) affirmative solution for (a) in shape of a new construction that achieves a security in the order of $\min(2^n/n, 2^{(n+t)/2})$ block-queries.

POEx, though an implicit instance of **O**[XTX], tries to use this strategy to resolve (a). It fixes the input of the hash function to just the immediate previous input-output of the underlying TBC to avoid the ℓ factor while maintaining a key size of $n+t$ bits. Unfortunately, there is a flaw in their analysis (see Section 4). The actual bound for POEx must have an ℓ factor.

4 Revisiting POEx

In this section we review the security of POEx, a rate-1 online cipher based on tweakable block ciphers, which claims Beyond Birthday Security [18].

4.1 Description of POEx

POEx is an extended version of POE, the online cipher used under POET [1]. It constructs an online cipher by iterated usage of a fixed tweak length TBC and a pAXU hash function on two block input/output. The algorithmic description of POEx is given in Algorithm 1 and a schematic illustration of the encryption/decryption process is shown in Fig. 1. On a macro level the construction looks neat and the security claim looks correct. But in the following subsections we describe a birthday bound attack on POEx that invalidates the security claim.

Algorithm 1 Definition of $\mathsf{POEx}[\widetilde{E}, H]$. $\mu, \nu \in \mathbb{F}_{2^n}$, are two application specific constants

function $\mathsf{POEx}[\widetilde{E}_K, H_L].\mathsf{Enc}(P)$	**function** $\mathsf{POEx}[\widetilde{E}_K, H_L].\mathsf{Dec}(C)$
$(X_0, Y_0) \leftarrow (\mu, \nu)$	$(X_0, Y_0) \leftarrow (\mu, \nu)$
$(P_1, \ldots, P_\ell) \xleftarrow{n} P$	$(C_1, \ldots, C_\ell) \xleftarrow{n} C$
for $i \leftarrow 1$ **to** ℓ **do**	**for** $i \leftarrow 1$ **to** ℓ **do**
$(T_i, U_i) \leftarrow H_L(X_{i-1}, Y_{i-1})$	$(T_i, U_i) \leftarrow H_L(X_{i-1}, Y_{i-1})$
$X_i \leftarrow P_i \oplus U_i$	$Y_i \leftarrow C_i \oplus U_i$
$Y_i \leftarrow \widetilde{E}_K^{T_i}.\mathsf{Enc}(X_i)$	$X_i \leftarrow \widetilde{E}_K^{T_i}.\mathsf{Dec}(Y_i)$
$C_i \leftarrow Y_i \oplus U_i$	$P_i \leftarrow X_i \oplus U_i$
end for	**end for**
return $C := (C_1 \| \cdots \| C_\ell)$	**return** $P := (P_1 \| \cdots \| P_\ell)$
end function	**end function**

The security of POEx is claimed to be $2^{\frac{n+t}{2}}$ block-queries. More specifically we have the exact security claim in Theorem 3.

Theorem 3 (Theorem 3 in [18])

$$\mathbf{Adv}^{\mathsf{osprp}}_{\mathsf{POEx}[\widetilde{E},H]} \leq 2\mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}^{\pm}}(\sigma, O(\tau)) + 2(\sigma+1)^2\varepsilon \cdot \left(2 + \frac{2^t}{2^n - (\sigma+1)}\right)$$

4.1.1 Flaw in POEx security analysis

The security of POEx mainly rely on the pAXU bound (say ε) of the underlying hash function. For example one of the bad events in the security proof of POEx is (T_j^i, X_j^i) (tweak-input) collision. This case has been bounded to $\varepsilon \cdot \sigma^2/2$. The argument being: there can be at most $\binom{\sigma}{2}$ pairs and for each pair the pAXU bound is ε. But this argument only holds when all inputs to the hash function are independent of the hash key. In case of POEx this is not true as the input to the hash function is dependent on the previous hash values

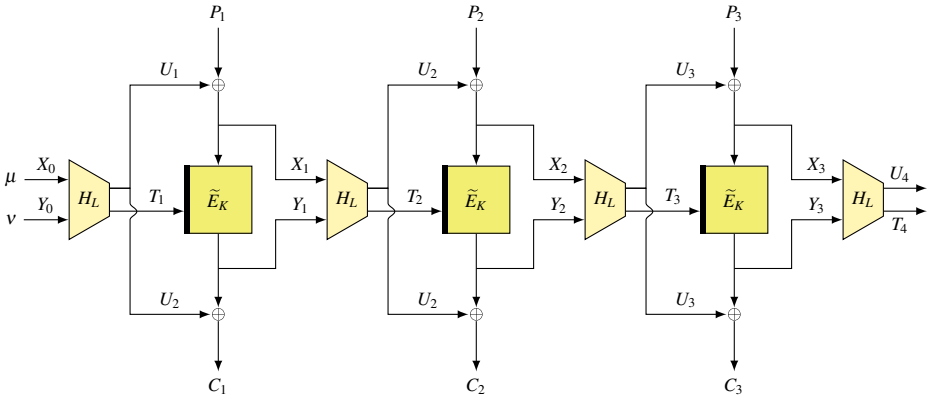

Fig. 1 Schematic of the encryption/decryption process for the first three blocks of plaintext/ciphertext using POEx construction

(linear combination of previous hash value and current message block). For example if we consider the candidate hash function given in [18], the hash-function inputs are dependent and the polynomial degree will accumulate through subsequent inputs to poly-hash. So the security claim is invalid. It seems that POEx requires a stronger assumption, i.e. it needs pAXU assumption on the iterated hash function. This may not be easy to achieve with algebraic hash functions.

4.2 Birthday bound attack on POEx

We substantiate the flaw discovered in the previous subsection by constructing an artificial example of a pAXU hash function that is not secure when used in iterated fashion. For the sake of simplicity we take $t = n$. Similar attacks can be constructed for any arbitrary t. The main idea is to construct an AXU hash, say SciFi, that on any given input converges to a fixed value when iterated a moderate number of times. By looking at the masking and tweak generator it can be observed that the adversary can use SciFi to fix the masking to an unknown but constant value. But the same is not possible for the tweak value as the Y_i value is not in the control of the adversary. Once the masking is fixed, the adversary can expect tweak collisions in birthday bound.

4.2.1 AXU hash using random mapping over ranked nodes

Let $\mathsf{rank} : \mathscr{B} \to [0..n]$ be a surjective function which is defined as follows:

$$\forall x \in \mathscr{B},\ \mathsf{rank}(x) := \begin{cases} 0 & x = 0, \\ i & 2^{i-1} \le x \le 2^i - 1. \end{cases}$$

Let $\mathscr{B}_i = \{x \in \mathscr{B} : \mathsf{rank}(x) = i\}$ and $\mathscr{B}_{<i} = \cup_{j<i}\mathscr{B}_j$. Clearly $\mathscr{B} = \sqcup_{i\in[n]}\mathscr{B}_i$. Let

$$\Psi \leftarrow_{\$} \mathsf{Perm}(\mathscr{B})$$
$$\Phi_0 \leftarrow_{\$} \mathsf{Func}(\mathscr{B}_0 \times \mathscr{B}, \mathscr{B}_0)$$
$$\forall i \in [1..n],\ \Phi_i \leftarrow_{\$} \mathsf{Func}(\mathscr{B}_i \times \mathscr{B}, \mathscr{B}_{<i})$$

We write Φ to denote the random vector $(\Phi_0, \ldots, \Phi_n)$. We define a keyed hash function $\mathsf{SciFi}_{\Psi,\Phi} : \mathscr{B} \times \mathscr{B} \to \mathscr{B}$ as follows:

$$\forall (x, y) \in \mathscr{B} \times \mathscr{B},\ \mathsf{SciFi}_{\Psi,\Phi}(x, y) := \Psi^{-1} \circ \Phi_{\mathsf{rank}(\Psi(x))}(\Psi(x), y).$$

For the sake of simplicity, we drop the subscript Ψ, Φ from $\mathsf{SciFi}_{\Psi,\Phi}$. It is easy to verify that after at most n iterations SciFi returns a fixed value, i.e. for any x and $\dot{y} := (y_1, \ldots, y_n) \in \mathscr{B}^n$ we have

$$\mathsf{SciFi}^n(x, \dot{y}) := \mathsf{SciFi}(\cdots(\mathsf{SciFi}(\mathsf{SciFi}(x, y_1), y_2)\cdots, y_n) = c,$$

where c is some unknown but constant value. This can be argued as follows: after each iteration the rank of the output reduces by at least 1. Since there are finite number of ranks, $n + 1$ to be precise, we must be at rank 0 in n iterations. Once we reach rank 0 the output becomes fixed to $\Psi^{-1}(0)$. We summarize this property in Lemma 1.

Lemma 1 *For all $m \ge n$, and $(x, \dot{y}) \in \mathscr{B} \times \mathscr{B}^m$, we have $\mathsf{SciFi}^m(x, \dot{y}) = c$, where $c = \Psi^{-1}(0)$.*

In Lemma 2 below, we show that SciFi is an $O\left(\frac{n}{2^n}\right)$-AXU hash function. Thus, security wise it can be a good candidate for AXU hash.

Lemma 2 *For distinct* $(x_1, y_1), (x_2, y_2) \in \mathscr{B} \times \mathscr{B}$ *and* $\delta \in \mathscr{B}$*, we have*

$$\Pr[\mathsf{SciFi}(x_1, y_1) \oplus \mathsf{SciFi}(x_2, y_2) = \delta] \leq \frac{n+2}{2^n}.$$

In other words, SciFi *is an* $(n+2)2^{-n}$*-AXU hash function.*

Proof Let R_1 and R_2 denote $\mathsf{rank}(\Psi(x_1))$ and $\mathsf{rank}(\Psi(x_2))$, respectively. Let

$$\mathtt{NZ} := [n] \times [n] \text{ and } \mathtt{Z} := [0..n] \times [0..n] \setminus \mathtt{NZ}.$$

Now we have,

$$\begin{aligned} &\Pr[\mathsf{SciFi}(x_1, y_1) \oplus \mathsf{SciFi}(x_2, y_2) = \delta] \\ &= \overbrace{\Pr[\mathsf{SciFi}(x_1, y_1) \oplus \mathsf{SciFi}(x_2, y_2) = \delta, (R_1, R_2) \in \mathtt{NZ}]}^{\varepsilon_1} \\ &\quad + \overbrace{\Pr[\mathsf{SciFi}(x_1, y_1) \oplus \mathsf{SciFi}(x_2, y_2) = \delta, (R_1, R_2) \in \mathtt{Z}]}^{\varepsilon_2} \end{aligned} \tag{2}$$

We bound ε_1 and ε_2 below:

Bound on ε_1. For a fixed $\psi \in \mathsf{Perm}(\mathscr{B})$ if we consider inputs (x_1, y_1) and (x_2, y_2) such that $\psi(x_1) \neq 0$ and $\psi(x_2) \neq 0$, then the bound effectively reduces to bounding the probability that the sum of outputs of two (possibly) independent random functions on distinct inputs equals to δ. This can be bounded by $2^{-\max(r_1, r_2)}$ (by conditioning on the output of the function with smaller range). Using the preceding argument and conditioning on Ψ, we have

$$\begin{aligned} \varepsilon_1 &\leq \sum_{(r_1,r_2)\in \mathtt{NZ}} \Pr[\Psi^{-1}(\Phi_{r_1}(\Psi(x_1), y_1)) \oplus \Psi^{-1}(\Phi_{r_2}(\Psi(x_2), y_1)) = \delta] \cdot \frac{2^{r_1+r_2-2}}{2^{2n}} \\ &\leq \sum_{(r_1,r_2)\in \mathtt{NZ}} \frac{1}{2^{\max(r_1,r_2)-1}} \cdot \frac{2^{r_1+r_2-2}}{2^{2n}} \\ &\leq \sum_{(r_1,r_2)\in \mathtt{NZ}} \frac{2^{r_1-1}}{2^{2n}} = \frac{n}{2^n} \end{aligned} \tag{3}$$

Bound on ε_2. In this case at least one of x_1 or x_2 is mapped to 0 by Ψ. Further $R_1 = R_2 = 0$ is possible if and only if $x_1 = x_2$, in which case we can simply bound the probability to at most 2^{-n}. So without loss of generality we assume that at least one of them is non-zero, say R_1. Now we can proceed as earlier and we have

$$\begin{aligned} \varepsilon_2 &\leq \frac{1}{2^n} + \sum_{r_1\in[n]} \Pr[\Psi^{-1}(\Phi_{r_1}(\Psi(x_1), y_1)) = x_2 \oplus \delta] \cdot \frac{2^{r_1-1}}{2^{2n}} \\ &\leq \frac{1}{2^n} + \sum_{r_1\in[n]} \frac{1}{2^{r_1-1}} \cdot \frac{2^{r_1-1}}{2^{2n}} \\ &\leq \frac{2}{2^n} \end{aligned} \tag{4}$$

The result follows from (2)–(4). □

4.2.2 Attack description

Lemma 2 shows that SciFi is a good AXU hash function, whereas Lemma 1 shows that the iterated version $\mathsf{SciFi}^{\geq n}$ is a pathetic AXU hash. We use this later fact to construct a birthday bound attack on POEx. Suppose we instantiate the POEx construction using a tuple of AXU hash functions $H := (\mathscr{F}, \mathsf{SciFi})$ where $\mathscr{F} \leftarrow_{\$} \mathsf{Func}(\mathscr{B} \times \mathscr{B}, \mathscr{B})$ is chosen independently of SciFi. Further the transition is defined as follows:

$$H(X_i, Y_i) = (\mathscr{F}(X_i, Y_i), \mathsf{SciFi}(X_i, Y_i))$$

Since $\mathscr{F}$ is a uniform random function over $\mathsf{Func}(\mathscr{B} \times \mathscr{B}, \mathscr{B})$ and hence a 2^{-n}-AXU hash function, using Proposition 1 we can conclude that H is an $\frac{n+2}{2^{2n}}$-pAXU hash function. Note that the choice of universal hash is not rigid. We can take any good candidate of universal hash provided it is keyed independently of SciFi. Now consider an adversary $\mathbf{A}$ that works as follows:

1. $\mathbf{A}$ makes q queries of the form $(P_1^i \| 0^{\ell-1})$, such that for distinct $i, i' \in [q]$ $P_1^i \neq P_1^{i'}$ and $\ell > n$, and observes the outputs $(C_1^i \| \cdots \| C_{\ell-1}^i)$.
2. If $(C_j^i, C_{j+1}^i) = (C_{j'}^{i'}, C_{j'+1}^{i'})$ for two distinct block indices (i, j) and (i', j'), then $\mathbf{A}$ returns 1.

For an ideal online cipher this should require roughly 2^n many blocks. But for $\mathsf{POEx}[\widetilde{\Pi}, H]$ this would require roughly $2^{n/2}$ many blocks. This can be argued using Lemma 1 which, in this case, imply that for each query beyond block index $n-1$, the input becomes a constant value. So all that is required is a tweak collision which can be achieved if we have roughly $2^{n/2}$ blocks. *Hence* POEx *is only birthday bound secure online cipher.*

Remark 1 We would like to remark here that the hash function notion (pAXU) used in the original POEx design [18] has been modified in a later journal version [19] with due acknowledgments to our observations. The authors have defined a new and stronger notion of hash function, called Chained-Partial-AXU Hash Function (an extension of the pAXU notion for iterated use), and based their security analysis on this notion. Note that the authors have not given any practical instantiations for such hash function and we speculate that given the stringent conditions it will be difficult to construct an efficient candidate with close to $2^{-(n+t)}$ bound. We refer the readers to [19] for a more detailed exposition. In this paper we follow the original POEx design from [18].

5 XTC: rate-1 (almost) optimally secure online cipher

In the preceding section we showed a subtle flaw in the security claim (and proof) of POEx. Here we explore the possibility of another approach toward a practical instantiation of Andreeva et al. [3] TBC to online cipher converter. As discussed in Section 3, $\mathbf{O}$[XTX] cannot achieve both rate-1 and BBB security simultaneously. A possible remedy is the idea of using just a fixed number of previous input-output block-pairs information. Based on this idea, we propose XTC, that achieves min $(2^n/n, 2^{\frac{n+t}{2}})$ block-queries security. The XTC construction is similar to POEx in the sense that it follows POEx's idea of generating the mask and tweak using a fixed number of previous blocks information. Both POEx and XTC can be viewed as possible candidates of TBC to online cipher converter [3], although based on slightly different strategies. POEx is an $\mathbf{O}$[XTX]-like scheme which uses pAXU property

on hash function with at most ℓ blocks input, whereas XTC uses pAXU property on hash function with at most three blocks input.

5.1 An impossibility result

In POEx [18] as well as XTC (see Section 5.2), only a small number of previous blocks are used for tweak and mask computations. POEx uses the immediate previous TBC input and output blocks, where as XTC employs the previous two plaintext and ciphertext blocks. This is done to avoid the loss of ℓ factor in security. In Theorem 4 we show an impossibility result that would imply that this approach will be futile when the security goal is more than 2^n block-queries, which is possible for $t > n$.

Definition 4 A pair of distinct vectors $(a_1, \ldots, a_m)$ and $(b_1, \ldots, b_m)$ is called an m-block *all-but-first collision pair* if $a_i = b_i$ for all $i \in [2..m]$.

Theorem 4 *Given $m \geq 3$, for an ideal online cipher an m-block all-but-first collision pair can be constructed in $O(m^2 \cdot 2^n)$ block-queries.*

Proof Consider an adversary **A** that works as follows:

1. Makes roughly $2^{n/2}$ distinct encryption queries (P_1^i, x) and stores the result (C_1^i, C_2^i) in $\mathscr{L}$. At least one collision on the second ciphertext block is expected, say (P_1^i, x) and (P_1^j, x) with the corresponding ciphertext (C_1^i, c_2) and (C_1^j, c_2).
2. Renames P_1^i, P_1^j, C_1^i, and C_1^j as p_1, p_1', c_1 and c_1'. Fixes $p_2 = p_2' := x$ and $c_2' = c_2$.
3. For $i \in [2..m]$
   (a) For all $a \in \mathscr{B}$ makes 2 queries $(p_1, p_2, \ldots, a)$ and $(p_1', p_2, \ldots, a)$, and checks if $(c_1, c_2, \ldots, c_i) = (c_1', c_2, \ldots, c_i')$. If the equality holds, then fixes $p_i = p_i' := a$ and moves to next i.
4. Sets $p := (p_1, \ldots, p_m)$, $c := (c_1, \ldots, c_m)$, $p' := (p_1', \ldots, p_m')$, and $c' = (c_1', \ldots, c_m')$.
5. Returns (p, c) and (p', c') as the m-blocks all-but-first collision pair.

As the first block is distinct for all queries in step 1 above, there is very high chance of getting a collision pair. In step 3(a) at the i-th iteration $(p_1, p_2, \ldots, p_{i-1})$ is distinct from $(p_1', p_2, \ldots, p_{i-1})$ so the outputs c_i and c_i' are uniformly and independently drawn from $\mathscr{B}$, whence we expect one collision in 2^n tries. Hence the total block-query complexity is bounded by $O(m^2 \cdot 2^n)$. □

Step 1 of Theorem 4 proof gives a simple corollary that lower bounds the number of previous blocks information required for n bits security.

Corollary 1 *To achieve n bits security at least 2 previous plaintext-ciphertext blocks information is required.*

Proof The proof follows from step 1 of the proof of Theorem 4. □

Remark 2 Note that, although we have constructed an all-but-first collision pair with start index as the first index of the queries, similar strategy can be applied if the collision pair has to be shifted to a later start index.

Using Theorem 4 and Remark 2 it is clear that to achieve more than 2^n block-queries security one has to process all the previous blocks.

5.2 Description of XTC

For the sake of simplicity we assume tweak size $t = n$ while describing the scheme and its security. Later we give ways to extend the result for all t. Using Corollary 1 we know that 2 previous plaintext-ciphertext block-pairs can be sufficient for the desired security goal. The algorithmic description of XTC is given in Algorithm 2, while Fig. 2 gives a pictorial illustration of the encryption/decryption process. XTC can be viewed as an iteration of XTX, much like **O**[XTX], albeit with tweak length fixed to three blocks. We call this equivalent view **O**′[XTX]. In Fig. 2, this equivalent view is represented by dashed rectangles which denote the underlying XTX component.

Although **O**[XTX] and XTC are similar in their use of XTX for tweak length extension, yet XTC is much more efficient while maintaining a satisfactory level of security. For a plaintext-ciphertext pair (P, C) we only use $(S_{i-2} = P_{i-2} \oplus C_{i-2}, P_{i-1}, C_{i-1})$ as the tweak input to the i-th block XTX. While this enables the application of efficient algebraic hash functions within XTX, the security analysis of the overall scheme becomes a bit more involved.

5.2.1 Design choices and rationale

We choose the pair $(S_{i-2}, P_{i-1}, C_{i-1})$ as it reduces the state size by one block, and is the simplest such pair. Further it can be easily verified that we cannot reduce this to 2 blocks without compromising on security. As far as the choice of hash function is concerned, we need universal property for the tweak part and XOR universal for the masking part. In other words we need H to be a pAXU hash over 3 blocks input. Since we are considering only rate-1 constructions we recommend algebraic hash functions for H.

BRW-based pAXU candidate BRW hash function is an efficient candidate that requires just one multiplication when the input is restricted to three blocks. It was proposed by Bernstein [9] based on previous works by Rabin and Winograd [31]. For a 3-blocks input (a_1, a_2, a_3), $\mathsf{BRW}_x(a_1, a_2, a_3)$ is defined as:

$$\mathsf{BRW}_x(a_1, a_2, a_3) := (a_1 \oplus x) \odot (a_2 \oplus x^2) \oplus a_3,$$

where $\odot$ and $\oplus$ denote field multiplication and addition, respectively, over $\mathbb{F}_{2^n}$ generated by some predefined primitive element. It is well-known that BRW hash with three blocks input is $3 \cdot 2^{-n}$-universal [9, 15].

For $L \in \mathscr{B}^2$, let L_1 and L_2 be the most and least, respectively, significant n bits of L. We define the keyed hash function, $H_L : \mathscr{B} \times \mathscr{B} \times \mathscr{B} \to \mathscr{B} \times \mathscr{B}$ as follows

$$\forall S, P, C \in \mathscr{B}, \quad (T, U) := H_L(S, P, C) = (\mathsf{BRW}_{L_1}(C, S, P), \mathsf{BRW}_{L_2}(C, S, 0)).$$

Note that the hash function definition is not arbitrary over the three block inputs. For example the security reduces to birthday bound if we just swap the values of T and U as after the swapping, U does not follow AXU. In Lemma 3 we bound the pAXU probability of H.

Lemma 3 *For distinct* $(S, P, C), (S', P', C') \in \mathscr{B}^3$, *and* $\delta \in \mathscr{B}$ *we have,*

$$\Pr_L[H_L(S, P, C) \oplus H_L(S', P', C') = (0, \delta)] \leq \frac{9}{2^{2n}}.$$

Proof To compute the output (T, U), H_L employs two calls to BRW hash with independent keys L_1 and L_2, which enables a universal bound of $9 \cdot 2^{-n}$. But we need pAXU property, which in this case means that the second output should have AXU property. Note that the last input block for the second call of BRW is always zero. This enables us to consider the difference block as part of the input and reduce the AXU bound to universal bound.

$$\begin{aligned}
\Pr_L[H_L(S, P, C) \oplus H_L(S', P', C') = (0, \delta)] &\leq \Pr_{L_1}[\mathsf{BRW}_{L_1}(C, S, P) = \mathsf{BRW}_{L_1}(C', S', P')] \\
&\quad \cdot \Pr_{L_2}[\mathsf{BRW}_{L_2}(C, S, 0) \oplus \mathsf{BRW}_{L_2}(C', S', 0) = \delta] \\
&= \Pr_{L_1}[\mathsf{BRW}_{L_1}(C, S, P) = \mathsf{BRW}_{L_1}(C', S', P')] \\
&\quad \cdot \Pr_{L_2}[\mathsf{BRW}_{L_2}(C, S, \delta) = \mathsf{BRW}_{L_2}(C', S', 0)] \\
&\leq \frac{9}{2^{2n}}
\end{aligned}$$

□

We emphasize here that the above definition is not the only possibility. Indeed, one can use polyhash to further reduce the state size at the cost of two more multiplication. In general any good pAXU hash function can be employed. In this work we mainly focus on saving on the number of multiplications.

Algorithm 2 Definition of $\mathsf{XTC}[\widetilde{E}, H]$. We take $\mu = \eta = 0$ and $\nu = 1$

function $\mathsf{XTC}[\widetilde{E}_K, H_L].\mathsf{Enc}(P)$	**function** $\mathsf{XTC}[\widetilde{E}_K, H_L].\mathsf{Dec}(C)$
$S_{-1} \leftarrow \eta$	$S_{-1} \leftarrow \eta$
$P_0 \leftarrow \mu$	$P_0 \leftarrow \mu$
$C_0 \leftarrow \nu$	$C_0 \leftarrow \nu$
$(P_1, \ldots, P_\ell) \xleftarrow{n} P$	$(C_1, \ldots, C_\ell) \xleftarrow{n} C$
for $i \leftarrow 1$ **to** ℓ **do**	**for** $i \leftarrow 1$ **to** ℓ **do**
$(T_i, U_i) \leftarrow H_L(S_{i-2}, P_{i-1}, C_{i-1})$	$(T_i, U_i) \leftarrow H_L(S_{i-2}, P_{i-1}, C_{i-1})$
$X_i \leftarrow P_i \oplus U_i$	$Y_i \leftarrow C_i \oplus U_i$
$Y_i \leftarrow \widetilde{E}_K^{T_i}.\mathsf{Enc}(X_i)$	$X_i \leftarrow \widetilde{E}_K^{T_i}.\mathsf{Dec}(Y_i)$
$C_i \leftarrow Y_i \oplus U_i$	$P_i \leftarrow X_i \oplus U_i$
$S_i \leftarrow P_i \oplus C_i$	$S_i \leftarrow P_i \oplus C_i$
end for	**end for**
return $C := (C_1 \| \cdots \| C_\ell)$	**return** $P := (P_1 \| \cdots \| P_\ell)$
end function	**end function**

Remark 3 Deriving hash keys via TBC calls. XTC requires two hash keys. However it can be easily reduced to one by reserving one tweak bit for key generation. For example one can use $(L_1, L_2) = (\widetilde{E}_K^{1\|0}(0), \widetilde{E}_K^{1\|1}(1))$ as the hash key and fix the most significant bit (MSB)

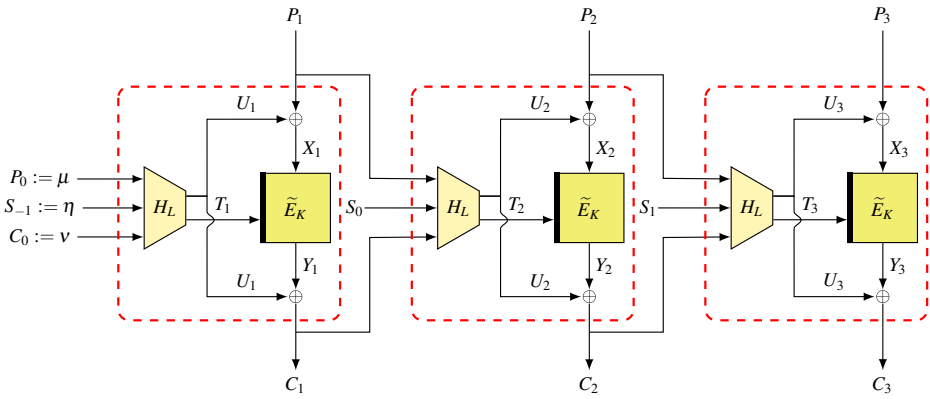

Fig. 2 Schematic of the encryption/decryption process for the first three blocks of plaintext/ciphertext using XTC construction. For each $i \geq 0$, $S_i := P_i \oplus C_i$. Dashed rectangles denote the XTX components of XTC and can be used to view XTC as $\mathbf{O}'[\mathsf{XTX}]$

of the tweaks for each TBC in block processing to 0. This will lead to a nominal loss of 1 bit security.

5.3 Security of XTC

We show that $\mathsf{XTC}[\widetilde{\Pi}, H]$ construction is secure upto $2^n/n$ block-queries given the hash function H is $O(2^{-2n})$-pAXU hash and $\widetilde{\Pi}$ is a uniform tweakable random permutation. More generally, we prove the upper bound result on the OSPRP advantage of $\mathsf{XTC}[\widetilde{E}, H]$ in Theorem 5.

Theorem 5 *If* XTC *is defined as above, H is an ε-pAXU, and $\sigma \leq 2^{n-3}$, then we have*

$$\mathbf{Adv}^{\mathsf{osprp}}_{XTC}[\widetilde{E}, H](q, \ell, \sigma, \tau) \leq \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}}(\sigma, \tau'') + \sigma^2\varepsilon + \frac{2(n+2)\sigma}{2^n}.$$

Before proceeding with the proof it would be better to discuss the main crux of the proof briefly. We employ the iterated TBC view (see Section 3) of an online cipher. The main steps of the proof are shown in Fig. 3. Basically we reduce the original XTC construction based on $\widetilde{E}$ and H to a variant of $\mathbf{O}[\widetilde{\mathbf{E}}]$, that we call $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$. $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$ behaves exactly as $\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]$, except that it restricts the i-th block tweak to $(S_{i-2}, P_{i-1}, C_{i-1})$. Finally we bound the distance between $\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]$ and $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$. Intuitively the two oracles will behave identically until there is a tweak collision in one of them which is not reciprocated by the other one. Now a tweak collision in $\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]$ always implies a tweak collision in $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$. But the converse is not true and we bound the probability of this event to complete the proof.

Proof We will follow the series of steps shown in Fig. 3. Step (1) simply transforms XTC to $\mathbf{O}'[\mathsf{XTX}]$. Steps (2) and (3) are used to replace the underlying $\mathsf{XTX}[\widetilde{E}, H]$ with an ATL TRP, $\widetilde{\boldsymbol{\Pi}}$. Using Theorem 2, steps (2) and (3) are bounded by $\mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}}(\sigma, \tau'') + \varepsilon\sigma^2$. In step (4) we upper bound the distance between $\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]$ and $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$ using the game-playing technique [7, 8].

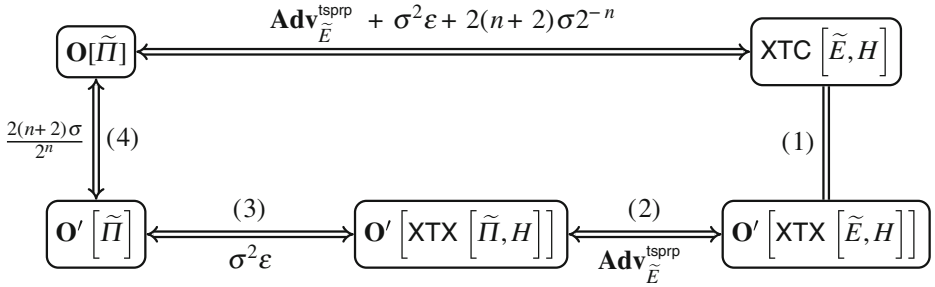

Fig. 3 Main steps of the proof are given in (1)–(4). $\Longleftrightarrow$ with overlying text denotes the distance between the corresponding schemes and $=$ denotes the equivalence between the corresponding schemes. Sequence of steps: (1) XTX based view of XTC; (2) and (3) TSPRP security of XTX; (4) Distance between $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$ and $\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]$

Let $\mathbf{A} \in \mathbb{A}(q, \ell, \sigma)$ be a computationally unbounded deterministic adversary that makes q distinct queries adaptively to either (1) the real oracle $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$, or (2) the ideal oracle, $\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]$. Refer to games G1 and G2 as shown in Fig. 4. For each tweak input x, sets $\mathsf{domain}(\widetilde{\boldsymbol{\Pi}}^x)$ and $\mathsf{range}(\widetilde{\boldsymbol{\Pi}}^x)$ are initialized as empty sets and automatically grow as points are added to the domain and range of the partial function $\widetilde{\boldsymbol{\Pi}}^x$. At any instant, sets $\overline{\mathsf{domain}}(\widetilde{\boldsymbol{\Pi}}^x)$ and $\overline{\mathsf{range}}(\widetilde{\boldsymbol{\Pi}}^x)$ represent the complements of these sets relative to $\mathscr{B}$. The internal variables arising in one call to XTC are analogously as given in Algorithm 2 and Figs. 2 and 3.

We now briefly describe the working of the two games. Game G1 and G2 faithfully simulate $\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]$ and $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$, respectively. Consider the encryption/decryption process at the (i, j)-th block (i-th query and j-th block) for $i \in [q]$ and $j \in [\ell^i]$. Without loss of generality, we assume that i is an encryption query. It is easy to see that both $\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]$ and $\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}]$ have identical output distributions until the `bad` flag is not set by G2. Obviously the output distribution is identical when there is no tweak collisions in either of the game. A tweak collision in G1, i.e., $(P^i_{<j}, C^i_{<j}) = (P^{i'}_{<j}, C^{i'}_{<j})$, would imply a tweak collision in G2, i.e., $\Gamma^i_j := (S^i_{j-2}, P^i_{j-1}, C^i_{j-1}) = \Gamma^{i'}_{j'} := (S^{i'}_{j-2}, P^{i'}_{j-1}, C^{i'}_{j-1})$, whence the output distribution is identical. But a tweak collision in G2 not necessarily mean a tweak collision in G1. This is captured by the setting of `bad` flag in game G2. Let `BadInput` denote the event that $\mathbf{A}$ sets `bad`, i.e.,
`BadInput` : $\exists (i, j), (i', j') \in \mathscr{I}$,

$$\dashv (P^i_{<j} \neq P^{i'}_{<j'}) \wedge (\Gamma^i_j = \Gamma^{i'}_{j'}).$$

In this case, the output of G1 is uniform and random over $\mathscr{B}$, whereas the output of G2 is either completely determined (when $P^i_j = P^{i'}_{j'}$) or uniform and random over $\overline{\mathsf{range}}(\widetilde{\boldsymbol{\Pi}}^{\Gamma^i_j}) \subseteq \mathscr{B}$, whence the two output distributions differ. So the fundamental lemma of game-playing [8] says us that,

$$\mathbf{Adv}^{\mathsf{osprp}}_{\mathbf{O}'[\widetilde{\boldsymbol{\Pi}}];\mathbf{O}[\widetilde{\boldsymbol{\Pi}}]}(\mathbf{A}) \leq \Pr[\mathtt{BadInput}].$$

Bound on Pr[`BadInput`]. In the following discussion we assume that all the queries are of encryption type. This will not hamper the correctness of our analysis due to the online

Game G1

On the i-th encryption query:

for $j \leftarrow 1$ **to** $|P^i|$ **do**
  if $\exists i' < i \dashv P^i_{<j} = P^{i'}_{<j}$ **then**
    if $P^i_j = P^{i'}_j$ **then**
      $C^i_j \leftarrow C^{i'}_j$
    else
      $C^i_j \leftarrow\!\$\, \overline{\mathsf{range}}(\widetilde{\Pi}^{(P^i_{<j}, C^i_{<j})})$
      $\widetilde{\Pi}^{(P^i_{<j}, C^i_{<j})}(P^i_j) \leftarrow C^i_j$
    end if
  else
    $C^i_j \leftarrow\!\$\, \mathscr{B}$
    $\widetilde{\Pi}^{(P^i_{<j}, C^i_{<j})}(P^i_j) \leftarrow C^i_j$
  end if
end for
return $C^i := (C^i_1, \ldots, C^i_{|P^i|})$

Game G2

On the i-th encryption query:

for $j \leftarrow 1$ **to** $|P^i|$ **do**
  if $\exists (i' < i) \wedge \exists (j' < j) \dashv \Gamma^i_j = \Gamma^{i'}_{j'}$ **then**
    if $P^i_{<j} \neq P^{i''}_{<j'}$ **then**
      set `bad`
    end if
    if $P^i_j = P^{i'}_{j'}$ **then**
      $C^i_j \leftarrow C^{i'}_{j'}$
    else
      $C^i_j \leftarrow\!\$\, \overline{\mathsf{range}}(\widetilde{\Pi}^{\Gamma^i_j})$
      $\widetilde{\Pi}^{\Gamma^i_j}(P^i_j) \leftarrow C^i_j$
    end if
  else
    $C^i_j \leftarrow\!\$\, \mathscr{B}$
    $\widetilde{\Pi}^{\Gamma^i_j}(P^i_j) \leftarrow C^i_j$
  end if
end for
return $C^i := (C^i_1, \ldots, C^i_{|P^i|})$

Fig. 4 Game G1 and G2 used in the proof of Theorem 5. G1 corresponds to the iterated view of an online cipher $\mathbf{O}[\widetilde{\Pi}]$. G2 corresponds to $\mathbf{O}'[\widetilde{\Pi}]$. In G2, $\Gamma^i_j := (S^i_{j-2}, P^i_{j-1}, C^i_{j-1})$. We skip the decryption procedures for both the game as they are similar to the respective encryption procedure and can be described analogously. Note that the output distribution of the two games are identical until the `bad` flag is not set

property, rather it will greatly simplify the analysis. The basic idea is to first bound the number of multicollisions on (S^i_{j-2}, P^i_{j-1}), and then for each (i, j) bound the probability of C^i_{j-1} collisions over the multicollision set that contains (S^i_{j-2}, P^i_{j-1}). Note that $(P^i_{<j} \neq P^{i'}_{<j'}) \wedge (P^i_{j-1} = P^{i'}_{j'-1}) \implies (P^i_{\leq j-2} \neq P^{i'}_{\leq j'-2})$, otherwise the probability of `BadInput` is zero. Let

$$\mathscr{I}_{\mathsf{pre}} := \left\{(i, j) \in \mathscr{I} \ : \ \forall (i', j') < (i, j) \text{ we have } P^i_{\leq j} \neq P^{i'}_{\leq j'}\right\}$$

We define the multicollision relation $\sim$ on $\mathscr{I}_{\mathsf{pre}}$ as follows:

$$\forall (i, j), (i', j') \in \mathscr{I}_{\mathsf{pre}},\ (i, j) \sim (i', j') \iff (S^i_{j-1}, P^i_j) = (S^{i'}_{j'-1}, P^{i'}_{j'}).$$

Clearly $\sim$ is an equivalence relation. Let $\mathscr{P}_\alpha$ denote the equivalence class containing $\alpha \in \mathscr{I}_{\mathsf{pre}}$ and $\#\mathtt{mc} := \max_{\alpha \in \mathscr{I}_{\mathsf{pre}}} \left|\mathscr{P}_\alpha\right|$. Let $\mathtt{MC}_n$ denote the event $\#\mathtt{mc} > n$. We make the following claim on $\#\mathtt{mc}$.

Claim 1 For $\sigma \leq 2^{n-3}$, we have

$$\Pr[\mathtt{MC}_n] \leq \frac{4\sigma}{2^n}.$$

We are interested in the conditional probability of $\texttt{BadInput}$ given $\neg\texttt{MC}_n$. It is clear that for each $(i, j-1) \in \mathscr{I}_{\mathsf{pre}}$ we have $\left|\mathscr{P}_{(i,j-1)}\right| \leq n$, and we have to bound the probability of C^i_{j-1} collisions for at most these many pairs.

$$\begin{aligned}
\Pr[\texttt{BadInput}] &\leq \Pr[\texttt{MC}_n] + \Pr[\texttt{BadInput} \mid \neg\texttt{MC}_n] \\
&\leq \Pr[\texttt{MC}_n] + \sum_{(i,j-1)\in\mathscr{I}_{\mathsf{pre}}} \sum_{(i',j')\in\mathscr{P}_{(i,j-1)}} \Pr\left[C^i_{j-1} = C^{i'}_{j'}\right] && (5)\\
&\leq \Pr[\texttt{MC}_n] + \sum_{(i,j-1)\in\mathscr{I}_{\mathsf{pre}}} \sum_{(i',j')\in\mathscr{P}_{(i,j-1)}} \frac{1}{2^n - s_{(i,j-2)}} && (6)\\
&\leq \Pr[\texttt{MC}_n] + \sum_{(i,j-1)\in\mathscr{I}_{\mathsf{pre}}} \sum_{(i',j')\in\mathscr{P}_{(i,j-1)}} \frac{1}{2^n - \sigma} && (7)\\
&\leq \Pr[\texttt{MC}_n] + \sum_{(i',j')\in\mathscr{P}_{(i,j-1)}} \frac{n}{2^n - \sigma} && (8)\\
&\leq \frac{4\sigma}{2^n} + \frac{2n\sigma}{2^n} && (9)
\end{aligned}$$

Note that $P^i_{\leq j-2} \neq P^{i'}_{\leq j'-1}$, so the tweaks are distinct, whence the transition from (5) to (6), where $s_{(i,j-2)}$ denotes the number of (i_1, j_1) such that $P^{i_1}_{\leq j_1} = P^i_{\leq j-2}$. Using $s_{(i,j-2)} \leq \sigma$ we have (6) to (7). Using $\#\texttt{mc} \leq n$, $|\mathscr{I}_{\mathsf{pre}}| \leq \sigma$ and Claim 1 we have (7) to (8)–(9). □

Proof of Claim 1 Let $\#\texttt{mc} = m$ and for some $\alpha \in \mathscr{I}_{\mathsf{pre}}$, let $|\mathscr{P}_\alpha| = m$. For each $(i_a, j_a) \in \mathscr{P}_\alpha$, let $s_{(i_a,j_a)}$ denote the number of $(i, j) \in \mathscr{I}$ such that $P^i_{\leq j} = P^{i_a}_{\leq j_a}$. Let us fix one index, say lexicographically first one, (i_1, j_1) as our reference index. Then we get a system of $(m-1)$ equations of the form $(S^{i_1}_{j_1-1} = S^{i_a}_{j_a-1})_{a\in[2..m]}$. We can argue that all these equations are independent as $P^{i_a}_{\leq j_a-1} \neq P^{i_b}_{\leq j_b-1}$ where $a, b \in [m]$. So we have

$$\Pr[\#\texttt{mc} = m] \leq \frac{\binom{\sigma}{m}}{(2^n - s_{(i_2,j_2)})\cdots(2^n - s_{(i_m,j_m)})} \leq \frac{\binom{\sigma}{m}}{(2^n - \sigma)^{m-1}}.$$

Summing over all $m \geq n+1$, we have

$$\begin{aligned}
\Pr[\texttt{MC}_n] &= \sum_{m=n+1}^{\infty} \Pr[\#\texttt{mc} = m] \\
&\leq \sum_{m=n+1}^{\infty} \frac{\binom{\sigma}{m}}{(2^n - \sigma)^{m-1}} && (10)\\
&\leq \frac{1}{2} \sum_{m=n+1}^{\infty} \left(\frac{4\sigma}{2^n}\right)^m && (11)\\
&\leq \frac{4\sigma}{2^n} && (12)
\end{aligned}$$

From (10) to (11) we use the fact that $\sigma < 2^{n-1}$, and $n(m-1) > (n-1)m$ when $m > n$. Using the convergence result on infinite geometric series and assuming $\sigma \leq 2^{n-3}$, we get the result from (11) to (12). □

As a direct consequence of Theorem 5 and Lemma 3 we get the following corollary for the BRW-based instantiation of H.

Corollary 2 *If H is defined as in* Section 5.2.1*, and $\sigma \leq 2^{n-3}$, then we have*

$$\mathbf{Adv}^{\mathsf{osprp}}_{XTC}[\widetilde{E}, H](q, \ell, \sigma, \tau) \leq \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{E}}(\sigma, \tau'') + \frac{9\sigma^2}{2^{2n}} + \frac{2(n+2)\sigma}{2^n}.$$

5.4 Extending XTC for arbitrary tweak size

We extend the initial XTC scheme with BRW based hash function for any arbitrary tweak as follows:

- For $t < n$, we replace the tweak generating part of H by $\mathsf{chop}_t(\mathsf{BRW}_{L_1}(C, S, 0) \cdot (P \oplus L_1^3))$ where chop_t is the standard chop function that extracts the t-MSBs from an n-bits input. It can be easily shown that $\mathsf{BRW}_{L_1}(C, S, 0) \cdot (P \oplus L_1^3)$ is a $6 \cdot 2^{-n}$-AXU hash. So using Proposition 1 we can establish that $\mathsf{chop}_t(\mathsf{BRW}_{L_1}(C, S, 0) \cdot (P \oplus L_1^3))$ is a $6 \cdot 2^{-t}$-AU hash. Finally this gives a bound of $\min(2^n/n, 2^{\frac{n+t}{2}})$ block-queries.
- For $t > n$, we already know a 2^n block-queries attack (using Theorem 4) on XTC. While we cannot improve over it without incorporating all the previous blocks we can still get a sub-optimal upper bound of $2^n/n$ block-queries. This is achieved by padding the n-bits tweak generated by the hash function with zeros to make it t-bits. Clearly similar bounds as earlier will apply.

Combining the two cases we conclude that XTC is secure while $\sigma \ll \min(2^n/n, 2^{\frac{n+t}{2}})$.

6 Conclusion

In this paper we first discussed the practical instantiation of TBC based online cipher of [3] using XTX [27], and its limitations. We then showed a flaw in the security analysis of POEx which invalidates the BBB-security claim. Further we propose a rate-1 and $\min(2^n/n, 2^{\frac{(n+t)}{2}})$ block-queries secure online cipher called XTC. Both POEx and XTC can be viewed as practical instantiations of the generic converter by Andreeva et al. [3], albeit following different approaches. Apart from these we also show an impossibility result for online ciphers based on TBCs.

While we largely solved the problem for constructions with tweak size at most the block size, a rate-1 construction with more than 2^n block-queries security for tweak size more than the block size is still an open problem. One thing is clear that such constructions must use all the previous blocks to generate tweak and mask, otherwise the security reduces to $\tilde{O}(2^n)$ block-queries (see Section 3).

In this work we solely focus on standalone online ciphers. It might be an interesting future work to construct a BBB-secure online AE scheme based on XTC.

References

1. Abed, F., Fluhrer, S.R., Forler, C., List, E., Lucks, S., McGrew, D.A., Wenzel, J.: Pipelineable on-line encryption. In: Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers, pp. 205–223 (2014)

2. Amanatidis, G., Boldyreva, A., O'Neill, A.: Provably-secure schemes for basic query support in outsourced databases. In: Data and Applications Security XXI, Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security - DBSec 2007, pp. 14–30 (2007)
3. Andreeva, E., Barwell, G., Bhaumik, R., Nandi, M., Page, D., Stam, M.: Turning online ciphers off. IACR Transactions on Symmetric Cryptology 2017(2) (2017)
4. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: Proceedings of the Advances in Cryptology - ASIACRYPT 2013, Part I, pp. 424–443 (2013)
5. Andreeva, E., Luykx, A., Mennink, B., Yasuda, K.: COBRA: a parallelizable authenticated online cipher without block cipher inverse. In: 21st International Workshop on Fast Software Encryption - FSE 2014. Revised Selected Papers, pp. 187–204 (2014)
6. Bellare, M., Boldyreva, A., Knudsen, L.R., Namprempre, C.: On-line ciphers and the hash-cbc constructions. J. Cryptol. **25**(4), 640–679 (2012)
7. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. J. Comput. Syst. Sci. **61**(3), 362–399 (2000)
8. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Advances in Cryptology - EUROCRYPT 2006, Proceedings of the 25Th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, pp. 409–426 (2006)
9. Bernstein, D.J.: Polynomial evaluation and message authentication. http://cr.yp.to/antiforgery/pema-20071022.pdf. Access date is 27 July (2007)
10. Bhaumik, R., Nandi, M.: Olef: an inverse-free online cipher. an online SPRP with an optimal inverse-free construction. IACR Trans. Symmetric Cryptol. **2016**(2), 30–51 (2016)
11. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: fast and secure message authentication. In: Proceedings of the Advances in Cryptology - CRYPTO '99, pp. 216–233 (1999)
12. Boldyreva, A., Taesombut, N.: Online encryption schemes: new security notions and constructions. In: Topics in Cryptology - CT-RSA 2004, the Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23–27, 2004, Proceedings, pp. 1–14 (2004)
13. Bossuet, L., Datta, N., Mancillas-López, C., Nandi, M.: Elmd: a pipelineable authenticated encryption and its hardware implementation. IEEE Trans. Comput. **65**(11), 3318–3331 (2016)
14. Carter, L., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. **18**(2), 143–154 (1979)
15. Chakraborty, D., Ghosh, S., Sarkar, P.: A fast single-key two-level universal hash function. IACR Trans. Symmetric Cryptol. **2017**(1), 106–128 (2017)
16. Datta, N., Nandi, M.: Elme: a misuse resistant parallel authenticated encryption. In: Proceedings of the 19th Australasian Conference on Information Security and Privacy - ACISP 2014, pp. 306–321 (2014)
17. Fleischmann, E., Forler, C., Lucks, S.: Mcoe: a family of almost foolproof on-line authenticated encryption schemes. In: Fast Software Encryption - 19th International Workshop, FSE 2012. Revised Selected Papers, pp. 196–215 (2012)
18. Forler, C., List, E., Lucks, S., Wenzel, J.: POEX: a beyond-birthday-bound-secure on-line cipher. ArcticCrypt 2016 (2016). https://www.researchgate.net/publication/299565944_POEx_A_Beyond-Birthday-Bound-Secure_On-Line_Cipher. Access date is 27 July 2017
19. Forler, C., List, E., Lucks, S., Wenzel, J.: POEX: a beyond-birthday-bound-secure on-line cipher. Cryptogr. Commun. (2017). https://doi.org/10.1007/s12095-017-0250-9
20. Gilbert, E.N., MacWilliams, F.J., Sloane, N.J.A.: Codes which detect deception. Bell Syst. Tech. J. **53**, 405–424 (1974)
21. Halevi, S., Krawczyk, H.: MMH: software message authentication in the gbit/second rates. In: Proceedings of the 4th International Workshop on Fast Software Encryption, FSE '97, pp. 172–189 (1997)
22. Hoang, V.T., Reyhanitabar, R., Rogaway, P., Vizár, D.: Online authenticated-encryption and its nonce-reuse misuse-resistance. In: Proceedings of the 35Th Annual Cryptology Conference on Advances in Cryptology - CRYPTO 2015, Part I, pp. 493–517 (2015)
23. Krovetz, T.: Message authentication on 64-bit architectures. In: Selected Areas in Cryptography, 13Th International Workshop, SAC 2006, Revised Selected Papers, pp. 327–341 (2006)
24. Liskov, M., Rivest, R.L., Wagner, D.A.: Tweakable block ciphers. J. Cryptol. **24**(3), 588–613 (2011)
25. Luby, M., Rackoff, C.: How to construct pseudo-random permutations from pseudo-random functions (abstract). In: Proceedings of the Advances in Cryptology - CRYPTO '85, p. 447 (1985)
26. Mennink, B.: Insuperability of the standard versus ideal model gap for tweakable blockcipher security. Cryptology ePrint Archive Report 2017/474 (2017). http://eprint.iacr.org/2017/474
27. Minematsu, K., Iwata, T.: Tweak-length extension for tweakable blockciphers. In: Proceedings of the 15th IMA International Conference on Cryptography and Coding - IMACC 2015, Oxford, UK, December 1517, 2015, pp. 77–93 (2015)

28. Nandi, M.: A simple security analysis of hash-cbc and a new efficient one-key online cipher. IACR Cryptology ePrint Archive **2007**, 158 (2007)
29. Nandi, M.: Two new efficient cca-secure online ciphers: MHCBC and MCBC. In: Progress in Cryptology - INDOCRYPT 2008, Proceedings of the 9th International Conference on Cryptology in India, Kharagpur, India, December 14–17, 2008, pp. 350–362 (2008)
30. Nandi, M.: On the minimum number of multiplications necessary for universal hash functions. In: Fast Software Encryption - 21st International Workshop, FSE 2014. Revised Selected Papers, pp. 489–508 (2014)
31. Rabin, M.O., Winograd, S.: Fast evaluation of polynomials by rational preparation. Commun. Pure Appl. Math. **25**(4), 433–458 (1972)
32. Rogaway, P., Zhang, H.: Online ciphers from tweakable blockciphers. In: Topics in Cryptology - CT-RSA 2011 - the Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14–18, 2011. Proceedings, pp. 237–249 (2011)
33. Sarkar, P.: A new multi-linear universal hash family. Des. Codes Crypt. **69**(3), 351–367 (2013)
34. Stinson, D.R.: Combinatorial techniques for universal hashing. J. Comput. Syst. Sci. **48**(2), 337–346 (1994)
35. Stinson, D.R.: Universal hashing and authentication codes. Des. Codes Crypt. **4**(4), 369–380 (1994)
36. Wegman, M.N., Carter, L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. **22**(3), 265–279 (1981)
37. Winograd, S.: A new algorithm for inner product. IEEE Trans. Comput. **17**(7), 693–694 (1968)