

# Optimally Secure Tweakable Block Ciphers with a Large Tweak from $n$ -bit Block Ciphers

Yaobin Shen and François-Xavier Standaert

UCLouvain, ICTEAM, Crypto Group, Louvain-la-Neuve, Belgium  
firstname.lastname@uclouvain.be

**Abstract.** We consider the design of a tweakable block cipher from a block cipher whose inputs and outputs are of size  $n$  bits. The main goal is to achieve  $2^n$  security with a large tweak (i.e., more than  $n$  bits). Previously, Mennink at FSE'15 and Wang et al. at Asiacrypt'16 proposed constructions that can achieve  $2^n$  security. Yet, these constructions can have a tweak size up to  $n$ -bit only. As evident from recent research, a tweakable block cipher with a large tweak is generally helpful as a building block for modes of operation, typical applications including MACs, authenticated encryption, leakage-resistant cryptography and full-disk encryption.

We begin with how to design a tweakable block cipher with  $2n$ -bit tweak and  $n$ -bit security from two block cipher calls. For this purpose, we do an exhaustive search for tweakable block ciphers with  $2n$ -bit tweaks from two block cipher calls, and show that all of them suffer from birthday-bound attacks. Next, we investigate the possibility to design a tweakable block cipher with  $2n$ -bit tweak and  $n$ -bit security from three block cipher calls. We start with some conditions to build such a tweakable block cipher and propose a natural construction, called  $\tilde{G}1$ , that likely meets them. After inspection, we find a weakness in  $\tilde{G}1$  which leads to a birthday-bound attack. Based on  $\tilde{G}1$ , we then propose another construction, called  $\tilde{G}2$ , that can avoid this weakness. We finally prove that  $\tilde{G}2$  can achieve  $n$ -bit security with  $2n$ -bit tweak.

**Keywords:** Tweakable Block Cipher · Optimal ( $n$ -bit) Security · Large Tweak

## 1 Introduction

A block cipher  $E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{M}$  is a family of permutations over  $\mathcal{M}$  indexed by a key  $k \in \mathcal{K}$ . Tweakable block ciphers, formalized by Liskov et al. [LRW02], add another parameter called tweak to the classical block cipher. More formally, a tweakable block cipher  $\tilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$  is a family of permutations over  $\mathcal{M}$  indexed by a key  $k \in \mathcal{K}$  and a tweak  $t \in \mathcal{T}$ . Here the key  $k$  is secret and used to provide security, while the tweak  $t$  is public and used to provide variability.

As fundamental primitives, tweakable block ciphers have found a wide spectrum of applications, including tweakable encryption schemes [CS08, Dwo10, HR04, HR03, MM07, WFW05, Sar09], message authentication codes [IMPS17, Nai15], and authenticated encryption schemes [KR11, JNPS21].

A tweakable block cipher can be designed from scratch: examples date back to as early as the AES competition [Cro00, SO98]. This approach became more popular after Jean et al.'s TWEAKKEY framework [JNP14], which treats the key and the tweak in a similar manner. Following this framework, a number of tweakable block ciphers were proposed, such as Skinny [BJK<sup>+</sup>16] and Deoxys-BC [JNPS21].

A more generic approach is to design a tweakable block cipher from a classical block cipher such as the AES Rijndael in a block-box manner. Popular examples include

LRW1 [LRW02], LRW2 [LRW02], variants and extensions of LRW1 and LRW2 such as CLRW1 and CLRW2 [ZQG22, CS06, LST12, LS13, BGGS20, Rog04], Minematsu’s design [Min09], Mennink’s constructions [Men15], Wang et al.’s generalized constructions [WGZ<sup>+</sup>16], XHX [JLM<sup>+</sup>17] and XHX2 [LL18]. Early proposals like LRW1, LRW2 and their variants [LRW02, CS06, Rog04] are limited to the birthday-bound security so that their security guarantees vanish after approximately  $2^{n/2}$  queries where  $n$  is the block size. By cascading either LRW1 or LRW2, we can achieve beyond-birthday-bound security [JN20, BGGS20], and eventually the security bound can asymptotically approach full  $2^n$  security when the number of block cipher calls and the number of universal function calls increase [LS13, ZQG22]. Alternatively, by using a tweak-dependent key (i.e., the key of a block cipher call is generated depending on the tweak), Minematsu’s design [Min09] can achieve beyond-birthday-bound security  $\max\{2^{n/2}, 2^{n-|t|}\}$  in the standard model when the tweak size is shorter than  $n/2$  bits, while Mennink’s constructions [Men15], Wang et al.’s generalized constructions [WGZ<sup>+</sup>16], XHX [JLM<sup>+</sup>17], and XHX2 [LL18] can achieve at least  $2^n$  security in the ideal-cipher model.

Besides, from the design and potential applications perspective, tweakable block ciphers with flexible tweak sizes are in general interesting. This is for example achieved by dedicated designs like Skinny [BJK<sup>+</sup>16] and Deoxys-TBC [JNPS21], which allow tweak sizes up to  $2n$  bits when the block size and key size are both  $n$ -bit. A recent trend allows a even larger tweaks and several variants are proposed, e.g., Skinnye-64-256 [NSS20] and SKIN-NYee [NSS22] for up to  $3n$ -bit tweak and  $(5n+3)$ -bit tweak respectively, Deoxys-TBC-512 and Deoxys-TBC-640 [CJPS22] for up to  $3n$ -bit tweak and  $4n$ -bit tweak respectively. In general, the tweak of a TBC can be used to contain additional information associated with a plaintext block [MI15, Ava17]. Hence, it can be desirable to make the tweak longer than the block length for more flexible designs. As evident from recent research, a tweakable block cipher with a large tweak is in particular helpful as a building block for modes of operation. Typical applications include MACs [IMPS17] where a large tweak can lead to designs with improved efficiency, authenticated encryption [NSS20, NSS22, CJPS22, HC23] where a large tweak can lead to designs with improved security, leakage-resistant cryptography [BGPS21, SPS<sup>+</sup>22] where a large tweak can help to design schemes that are both more efficient and rely on an weaker physical assumptions, and full-disk encryption [ST13] where a large tweak can support more modular designs. Yet, when it comes to generic designs from classical block ciphers provably enjoying  $2^n$  security with large tweaks, the state-of-the-art is scarcer and to the best of our knowledge, all existing candidates require an additional primitive called universal hash to compress the tweak.

**OUR CONTRIBUTIONS.** In this paper, we focus on building a large-tweak (more than  $n$ -bit) tweakable block cipher with  $n$ -bit security from merely a block cipher. These constructions are interesting in the sense that (i) they only require block cipher calls without invoking other primitives and thus can be efficient, e.g., when AES-NI instructions or AES coprocessors are available; (ii) they are helpful as a building block for (possibly leakage-resistant) modes of operation that demand a tweakable block cipher with a tweak size of more than  $n$  bits; (iii) they may be useful against side-channel attacks, since they are merely based on block ciphers and the side-channel countermeasures for block ciphers like the AES are well studied, in software and hardware [GR17, MCS22].

We begin with a  $2n$ -bit tweak and study how to design a tweakable block cipher with  $2n$ -bit tweak and  $n$ -bit security from two block cipher calls, as the constructions from one block cipher call are at most birthday-bound secure as shown by Mennink [Men15]. We perform an exhaustive search on tweakable block ciphers from two block cipher calls that is partially based on the framework by Wang et al. [WGZ<sup>+</sup>16]. Our results show that for any tweakable block cipher with  $2n$ -bit tweak from two block cipher calls, there is always a birthday-bound attack, and thus invalidate the possibility to build a  $2n$ -bit tweakable block cipher with  $n$ -bit security from two block cipher calls (without other primitives).

Table 1: Comparison of  $\tilde{G}2$  with previous tweakable block ciphers. The column key size states the size of key required by the design. The column tweak size states the size of tweak supported by the design. The column #AXU states the number of universal hash functions required by the design. The column #E states the number of block cipher calls required by the design. The column tdk states if the design relies on tweak-dependent key. The column security states the security of the design proved in bits.

	key size	tweak size	#AXU	#E	tdk	security ( $\log_2$ )
LRW1	$n$	$n$	0	2	no	$n/2$ [LRW02]
LRW2	$2n$	arbitrary	1	1	no	$n/2$ [LRW02]
XEX	$n$	$n$	0	1	no	$n/2$ [Rog04]
CLRW1	$3n$	$n$	0	3	no	$2n/3$ [BGG20]
CLRW2	$4n$	arbitrary	2	2	no	$3n/4$ [JN20]
CLRW1 <sub><math>r</math></sub>	$rn$	$n$	0	$r$	no	$(r-1)n/(r+1)$ [ZQG22]
CLRW2 <sub><math>r</math></sub>	$2rn$	arbitrary	$r$	$r$	no	$rn/(r+2)$ [LS13]
Min	$n$	$t$	0	2	yes	$\max\{n/2, n-t\}$ [Min09]
XHX	$2n$	arbitrary	2	1	yes	$n$ [JLM <sup>+</sup> 17]
XHX2	$4n$	arbitrary	4	2	yes	$4n/3$ [LL18]
$\tilde{F}[1]$	$n$	$n$	0	1	yes	$2n/3$ [Men15]
$\tilde{F}[2]$	$n$	$n$	0	2	yes	$n$ [Men15]
$\tilde{E}1, \dots, \tilde{E}32$	$n$	$n$	0	2	yes	$n$ [WGZ <sup>+</sup> 16]
$\tilde{G}2$	$n$	$2n$	0	3	yes	$n$

We then investigate the possibility to design a tweakable block cipher with  $2n$ -bit tweak and  $n$ -bit security from three block cipher calls. We do not rely on the exhaustive search method by Wang et al. [WGZ<sup>+</sup>16] to build an  $n$ -bit tweak constructions from two block cipher calls since the number of possible constructions grows exponentially with the number of block cipher calls. Instead, we start with some desirable conditions to build a tweakable block cipher with  $2n$ -bit tweak from three block cipher calls and propose a construction called  $\tilde{G}1$  (illustrated in Figure 4) that likely meets them. Interestingly, after a closer study on this construction, we find some weakness in  $\tilde{G}1$  that consequently leads to a birthday-bound attack. Based on  $\tilde{G}1$ , we then propose another construction called  $\tilde{G}2$  (illustrated in Figure 5) that can avoid this issue. We prove that  $\tilde{G}2$  indeed achieves  $n$ -bit security with  $2n$ -bit tweaks. Note that our focus is to design a tweakable block cipher from a block cipher of  $n$ -bit size and an  $n$ -bit key. Hence the optimal security is expected to  $n$  bits since both key size and wire size are  $n$  bits, as in cases of [Men15] and [WGZ<sup>+</sup>16]. So the term optimal we use for our construction is limited to its security. As for its efficiency, there may be other constructions with  $2n$ -bit tweaks and  $n$ -bit security from three block cipher calls and we therefore do not claim optimality. Yet,  $\tilde{G}2$  is arguably a very efficient construction among the possible ones because (i) the first two block cipher calls can be computed in parallel; (ii) it only requires one tweak-dependent key. A comparison of  $\tilde{G}2$  with previous tweakable block ciphers is given in Table 1.

ORGANIZATION. We present notations and security notions in Section 2. We do an exhaustive search on tweakable block ciphers with  $2n$ -bit tweak and show birthday-bound attacks on them in Section 3. We propose two tweakable block ciphers  $\tilde{G}1$  and  $\tilde{G}2$  with  $2n$ -bit tweak from three block cipher calls in Section 4:  $\tilde{G}1$  is at most birthday-bound secure (as shown by an attack) while  $\tilde{G}2$  can achieve  $n$ -bit security supported by a proof. We conclude the paper in Section 5.

## 2 Preliminaries

NOTATION. Let  $\varepsilon$  denote the empty string. Let  $\{0, 1\}^*$  be the set of all finite bit strings including the empty string  $\varepsilon$ . For a finite set  $S$ , let  $x \xleftarrow{\$} S$  denote the uniform sampling from  $S$  assigning a value to  $x$ . Let  $|x|$  denote the length of the string  $x$ . Let  $x[i : j]$  denote the substring from the  $i$ -th bit to the  $j$ -th bit (inclusive) of  $x$ . Concatenation of strings  $x$  and  $y$  is written as  $x \parallel y$  or simply  $xy$ . If  $A$  is an algorithm, let  $y \leftarrow A(x_1, \dots; r)$  denote running  $A$  with randomness  $r$  on inputs  $x_1, \dots$  and assigning the output to  $y$ . Let  $y \xleftarrow{\$} A(x_1, \dots)$  be the result of picking  $r$  at random and letting  $y \leftarrow A(x_1, \dots; r)$ . Let  $\text{Perm}(n)$  denote the set of all permutations over  $\{0, 1\}^n$ , and let  $\text{Func}(*, n)$  denote the set of all functions from  $\{0, 1\}^*$  to  $\{0, 1\}^n$ .

BLOCK CIPHER AND TWEAKABLE BLOCK CIPHER. A block cipher  $E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{M}$  is a family of permutations, where  $E_k(\cdot) = E(k, \cdot)$  is a permutation over  $\mathcal{M}$  for each key  $k \in \mathcal{K}$ . We denote by  $E_k^{-1}(\cdot)$  its inverse for a fixed key  $k$ . We denote by  $\text{BC}(\mathcal{K}, \mathcal{M})$  the set of all such block ciphers.

A tweakable block cipher  $\tilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$  is a family of permutations, where  $\tilde{E}_k(t, \cdot) = \tilde{E}(k, t, \cdot)$  is a permutation over  $\mathcal{M}$  that is indexed by two functionally distinct parameters: a key  $k \in \mathcal{K}$  that is secret and used to provide the security, and a tweak  $t \in \mathcal{T}$  that is public and used to provide variability. Similarly, we denote by  $\tilde{E}_k^{-1}(t, \cdot)$  its inverse for a fixed key  $k$  and a tweak  $t$ . In the rest of this paper, we focus on a tweakable block cipher  $\tilde{E}$  that is built from a block cipher  $E$ . We denote by  $\widetilde{\text{Perm}}(\mathcal{T}, \mathcal{M})$  the set of all functions  $\tilde{\pi} : \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$  such that  $\tilde{\pi}(t, \cdot)$  is a permutation over  $\mathcal{M}$  for any  $t \in \mathcal{T}$ .

SECURITY DEFINITION. An adversary  $\mathcal{A}$  is an algorithm that always outputs a bit. We write  $\mathcal{A}^O = 1$  to denote the event that  $\mathcal{A}$  outputs 1 when given access to oracle  $O$ . Let  $\tilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$  be a tweakable block cipher that uses a block cipher  $E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{M}$  as the underlying primitive. Let  $\tilde{\pi} \xleftarrow{\$} \widetilde{\text{Perm}}(\mathcal{T}, \mathcal{M})$  be a tweakable random permutation. The advantage of  $\mathcal{A}$  in distinguishing  $\tilde{E}$  and  $\tilde{\pi}$  is defined as:

$$\text{Adv}_E^{\text{SPRP}}(\mathcal{A}) = \left| \Pr \left[ \mathcal{A}^{\tilde{E}_k^{\pm}, E^{\pm}} = 1 \right] - \Pr \left[ \mathcal{A}^{\tilde{\pi}^{\pm}, E^{\pm}} = 1 \right] \right| ,$$

where the probabilities are taken over the random choices of  $k \xleftarrow{\$} \mathcal{K}$ ,  $E \xleftarrow{\$} \text{BC}(\mathcal{K}, \mathcal{M})$ , and  $\tilde{\pi} \xleftarrow{\$} \widetilde{\text{Perm}}(\mathcal{T}, \mathcal{M})$ . We say the queries to  $\tilde{E}_k^{\pm}$  or  $\tilde{\pi}^{\pm}$  as construction queries, and the queries to  $E^{\pm}$  as ideal-cipher queries.

THE H-COEFFICIENT TECHNIQUE. Following Hoang and Tessaro [HT16], we consider the interaction between an adversary  $\mathcal{A}$  and an abstract system  $\mathbf{S}$  which answers  $\mathcal{A}$ 's queries. The resulting interaction can then be recorded with a transcript  $\tau$ . Let  $\text{ps}(\tau)$  denote the probability that  $\mathbf{S}$  produces  $\tau$ . Note that  $\text{ps}(\tau)$  is the description of  $\mathbf{S}$  and independent of the adversary  $\mathcal{A}$ . We say a transcript is attainable for the system  $\mathbf{S}$  if  $\text{ps}(\tau) > 0$ .

We now describe the H-coefficient technique of Patarin [Pat08, CS14]. Generically, it considers an adversary that aims at distinguishing a “real” system  $\mathbf{S}_1$  from an “ideal” system  $\mathbf{S}_0$ . The interactions of the adversary with those systems induce two transcript distributions  $X_1$  and  $X_0$  respectively. The upper bound on the distinguishing advantage of  $\mathcal{A}$  can be given by the statistical distance  $\text{SD}(X_1, X_0)$ .

**Lemma 1.** [Pat08, CS14] *Suppose that the set of attainable transcripts for the ideal system can be partitioned into good and bad ones. If there exists  $\epsilon \geq 0$  such that  $\frac{\text{ps}_1(\tau)}{\text{ps}_0(\tau)} \geq 1 - \epsilon$  for any good transcript  $\tau$ , then*

$$\text{SD}(X_1, X_0) \leq \epsilon + \Pr[X_0 \text{ is bad}] .$$

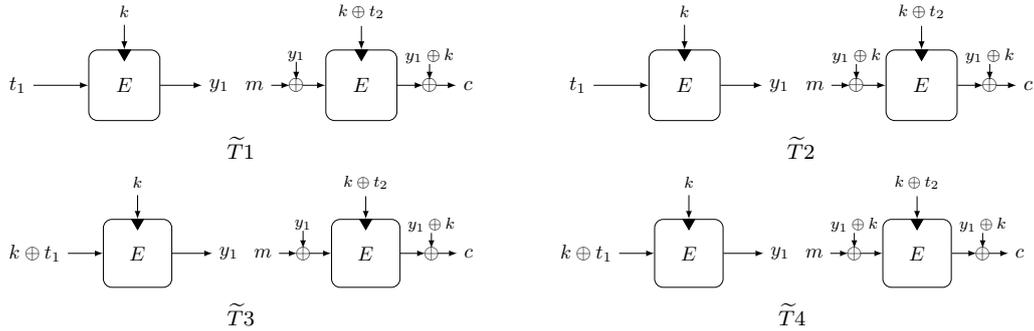


Figure 1: The four TBCs with two positions for tweak. Wang et al. [WGZ<sup>+</sup>16] showed that these TBCs has  $n$ -bit security when  $t_1 = t_2 = t$ . Here we consider the case when both  $t_1$  and  $t_2$  can be arbitrarily chosen by the adversary.

### 3 Attacks on Two Block Cipher Calls

In this section, we investigate the possibility to build a  $2n$ -bit tweak tweakable block cipher (TBC) with  $n$ -bit security from two block cipher calls. We do an exhaustive search on TBCs with a  $2n$ -bit tweak from two block cipher calls. Our results show that there is always a birthday-bound attack on these constructions. Here we focus on TBCs consisting of two block cipher calls and a linear transformation that is limited to XOR operation, as considered in [WGZ<sup>+</sup>16] for efficiency reason. A more generic construction of TBCs from block ciphers can be found in [Men15].

OVERVIEW OF OUR METHOD. We give a brief overview of how the exhaustive search proceeds. Our exhaustive search is partially based on Wang et al.’s framework [WGZ<sup>+</sup>16]. We discuss how some part of the exhaustive search can be reduced to results by Wang et al., and why the remaining parts require a new security analysis.

Following the framework by Wang et al. [WGZ<sup>+</sup>16], the constructions of TBC from two block cipher calls can be classified into two categories: either one block cipher call uses a tweak-dependent key and the other one uses a fixed key, or both two block cipher calls use a tweak-dependent key. The first category for an  $n$ -bit tweak has been exhaustively studied by Wang et al. and among them 56 TBCs with an  $n$ -bit tweak can achieve  $n$ -bit security. By contrast, the second category remains open even for  $n$ -bit tweak cases.<sup>1</sup> In Subsection 3.1, we show that the security of the first category for a  $2n$ -bit tweak can be reduced to that of  $n$ -bit tweak cases, Hence, we only need to consider whether we can turn these 56 TBCs to using a  $2n$ -bit tweak. We then find that only 4 out of 56 TBCs are possible candidates for a  $2n$ -bit tweak. However, as illustrated by our attacks, these four TBCs can achieve at most birthday-bound security. For the second category of a  $2n$ -bit tweak, we perform an exhaustive search in Subsection 3.2 and eventually find there is always a birthday-bound attack against them.

#### 3.1 On the Category with One Tweak-Dependent Key

For this category, the security analysis is partially based on [WGZ<sup>+</sup>16] that is first recalled as follows. Wang et al. [WGZ<sup>+</sup>16] performed an exhaustive search on the first category and found that 32 TBCs with  $n$ -bit tweak can achieve  $n$ -bit security where the first block cipher call can be pre-computed. They also found 24 TBCs [WGZ<sup>+</sup>16, Wan] with  $n$ -bit tweak and  $n$ -bit security for which the first block cipher cannot pre-computed. Here we

<sup>1</sup>In this paper, we don’t study the security of these TBCs with  $n$ -bit tweak since our main focus is to design a TBC with  $2n$ -bit tweak.

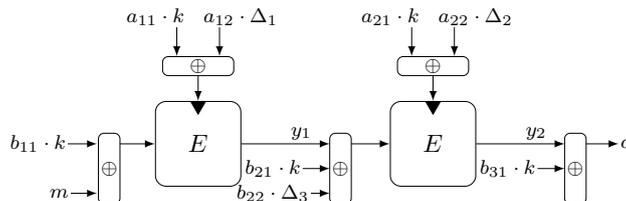


Figure 2: Type I constructions. Here  $\Delta_1, \Delta_2$  and  $\Delta_3$  represent three positions for the tweak.

can merely focus on these 56 constructions and check whether we can turn them into  $2n$ -bit tweak constructions while maintaining the  $n$ -bit security. This is because for constructions that are different from these 56 TBCs with  $2n$ -bit tweak, we can set the  $2n$ -bit tweak  $t_1 \parallel t_2$  to  $t_1 = t_2 = t$  that is equivalent to an  $n$ -bit tweak  $t$ . As shown in [WGZ<sup>+</sup>16], only these 56 constructions can achieve  $n$ -bit security with  $n$ -bit tweak. Among these 56 constructions, if there is only one position for the  $n$ -bit tweak that is either in the internal value or in the key, then there is no hope to turn them into  $2n$ -bit tweak unless we increase the block size to be  $2n$ -bit or increase the key size to be  $2n$ -bit. If there are two or more than two positions for the  $n$ -bit tweak, then we can use independent  $n$ -bit tweaks in each of two positions and thus the total tweak size becomes  $2n$ -bit. Yet, it requires new security analyses on these constructions since the previous analyses for  $n$ -bit tweak cannot directly apply to them. As illustrated in the following, they are at most birthday-bound secure.

Note that all these 32 TBCs for which the first block cipher can be pre-computed have only one position for the tweak, and 4 out of 24 TBCs for which the first block cipher cannot be pre-computed have two positions for the tweak. Hence, we consider these 4 TBCs and check whether can they maintain the  $n$ -bit security with  $2n$ -bit tweak. See Figure 1 for the illustration of these 4 TBCs.

We observe that these four TBCs have at most  $2^{n/2}$  security based on the following observation: by fixing the second tweak  $t_2$ , if  $m_i \oplus y_1^i = m_j \oplus y_1^j$ , then  $c_i \oplus c_j = y_1^i \oplus y_1^j = m_i \oplus m_j$ , which happens with query complexity  $2^{n/2}$  by changing both  $t_1$  and  $m$ . Take  $\tilde{T}1$  as an example. The adversary  $\mathcal{A}$  can mount an attack as follows. Firstly, fixing a tweak value  $t_2$ ,  $\mathcal{A}$  selects  $2^{n/2}$  distinct tweak values  $t_1^i$  and  $2^{n/2}$  distinct plaintexts  $m_i$ , and queries  $(t_1^i \parallel t_2, m_i)$  to  $\tilde{T}1(\cdot, \cdot)$  to search a match  $m_i \oplus m_j = c_i \oplus c_j$ . Let  $(t_1^i \parallel t_2, m_i, c_i)$  and  $(t_1^j \parallel t_2, m_j, c_j)$  denote the corresponding pair for this match. Secondly,  $\mathcal{A}$  selects a constant value  $\Delta \neq 0^n$ , queries  $(t_1^i \parallel t_2, m_i \oplus \Delta)$  and  $(t_1^j \parallel t_2, m_j \oplus \Delta)$  to  $\tilde{T}1(\cdot, \cdot)$ , and receives  $c'_i$  and  $c'_j$ , respectively. Finally,  $\mathcal{A}$  outputs 1 if  $c'_i \oplus c'_j = m_i \oplus m_j$ , and outputs 0 otherwise. The complexity of  $\mathcal{A}$  is  $O(2^{n/2})$ . When interacting with  $\tilde{T}1$ ,  $\mathcal{A}$  outputs 1 as long as she succeeds to find a match at the first step, which has a probability of about  $1 - (1 - 2^{-n})^{2^{n-1}} \approx 0.4$ . When interacting with a tweakable random permutation, the probability of  $\mathcal{A}$  outputting 1 is  $2^{-n}$ . Hence, the distinguishing advantage of  $\mathcal{A}$  is computed as  $0.4 - 2^{-n} \approx 0.4$ . Similar attacks hold for  $\tilde{T}2$ ,  $\tilde{T}3$  and  $\tilde{T}4$ .

### 3.2 On the Category with Two Tweak-Dependent Keys

For the second category, we perform an exhaustive search and find that all of the constructions suffer from birthday-bound attacks. The exhaustive search is based on the framework developed by Wang et al. [WGZ<sup>+</sup>16].

In [WGZ<sup>+</sup>16], the authors classified TBCs built from two block cipher calls into three types depending on the position of a plaintext: Type I, plaintext  $m$  is XORed to compute the input to the first block cipher; Type II, plaintext  $m$  is XORed to compute the input to the second block cipher; Type III, plaintext  $m$  is XORed to compute the ciphertext  $c$ . Since Type III constructions are trivially insecure, here we consider Type I and Type

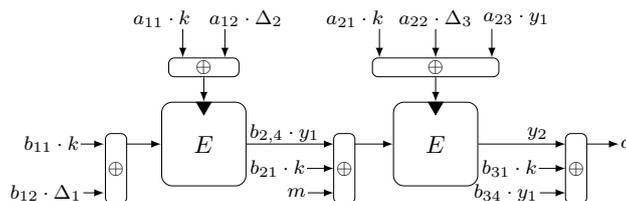


Figure 3: Type II constructions. Here  $\Delta_1, \Delta_2$  and  $\Delta_3$  represent three positions for the tweak.

II constructions that are illustrated in Figure 2 and Figure 3 respectively. As shown in [WGZ<sup>+</sup>16], XORing the tweak to the plaintext or ciphertext does not have any impact on the security. Therefore we omit these types of constructions. Intuitively, when XORing the tweak to the plaintext  $m$ , it is equivalent to querying the plaintext  $m \oplus t$  without this tweak. We refer the reader to [WGZ<sup>+</sup>16] for a more detailed discussion about these three types of constructions. In Type I and Type II constructions, coefficients  $a_{ij}$  and  $b_{ij}$  are one-bit variables being 0 or 1. Parameters  $\Delta_1, \Delta_2$  and  $\Delta_3$  are three positions for the tweak. Wang et al. [WGZ<sup>+</sup>16] consider the case when  $\Delta_1 = \Delta_2 = \Delta_3 = t$ . Here we study the possibility to use different tweaks in these positions and begin with the investigation of Type I constructions.

**TYPE I CONSTRUCTIONS.** Recall that we consider the category with two tweak-dependent keys, and hence  $a_{12} = a_{22} = 1$  in Type I constructions. Wang et al. showed birthday attacks on Type I constructions when  $(a_{12}, a_{22}) \neq (1, 1)$ ,<sup>2</sup> namely the cases of at most one tweak-dependent key. Here we show that the case  $(a_{12}, a_{22}) = (1, 1)$  can be reduced to  $(a_{12}, a_{22}) \neq (1, 1)$  when using a  $2n$ -bit tweak  $t_1 \parallel t_2$ , and thus they can achieve at most birthday-bound security. The analysis is as follows. We distinguish two cases depending on the value of  $b_{22}$ . If  $b_{22} = 0$ , then there are only two positions for the tweak such that  $\Delta_1 = t_1$  and  $\Delta_2 = t_2$  hold. We can fix either  $t_1 = 0^n$  or  $t_2 = 0^n$  which is then equivalent to  $(a_{12}, a_{22}) = (0, 1)$  or  $(a_{12}, a_{22}) = (1, 0)$ , and thus the attacks by Wang et al. [WGZ<sup>+</sup>16] apply to this case. If  $b_{22} = 1$ , then there are three positions for the tweak  $t_1 \parallel t_2$  and we have  $\binom{3}{2} = 3$  possibilities to put two  $n$ -bit tweaks  $t_1$  and  $t_2$  as follows:<sup>3</sup>

- $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_2, t_1)$ . Then we can fix  $t_2 = 0^n$  that is then equivalent to  $(a_{12}, a_{22}) = (1, 0)$ , and the attacks by Wang et al. apply.
- $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_2, t_2)$ . Then we can fix  $t_1 = 0^n$  that is equivalent to  $(a_{12}, a_{22}) = (0, 1)$ , and the attacks by Wang et al. apply.
- $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_1, t_2)$ . Then we can fix  $t_1 = 0^n$  that is equivalent to  $(a_{12}, a_{22}) = (0, 0)$ , and the attacks by Wang et al. apply.

Note that we can also use  $t_1 \oplus t_2$  in one of these three positions and bring in three more non-trivial possibilities  $(\Delta_1, \Delta_2, \Delta_3) \in \{(t_1 \oplus t_2, t_1, t_2), (t_1, t_1 \oplus t_2, t_2), (t_1, t_2, t_1 \oplus t_2)\}$ . However, these three possibilities are all subject to similar attacks as above. For example, for the case  $(\Delta_1, \Delta_2, \Delta_3) = (t_1 \oplus t_2, t_1, t_2)$ , we can set  $t_1 = t_2$  that is again equivalent to  $(a_{12}, a_{22}) = (0, 1)$  and the attacks by Wang et al. apply. Hence, Type I constructions with  $2n$ -bit tweak are secure up to birthday bound.

**TYPE II CONSTRUCTIONS.** We then consider the possibility of using a  $2n$ -bit tweak in Type II constructions. Wang et al. showed birthday attacks on the case of  $(a_{12}, a_{22}, a_{23}) =$

<sup>2</sup>They only consider attacks on  $(a_{12}, a_{22}) = (1, 0)$  and  $(0, 1)$ , but the case  $(a_{12}, a_{22}) = (0, 0)$  is implicitly covered by the attack for  $(a_{12}, a_{22}) = (1, 0)$ . Because the adversary can get to know the internal difference  $\Delta y_1 = t \oplus t'$  by one forward query  $(t, m, c)$  and one backward query  $(t', m', c)$

<sup>3</sup>Here the order of selecting  $t_1$  and  $t_2$  does not matter. For example,  $(t_2, t_1, t_1)$  is equivalent to  $(t_1, t_2, t_2)$ .

$(1, 0, 0)$ , and found 56 TBCs with  $n$ -bit tweak that can achieve  $n$ -bit security in the case of  $a_{12} = 0$ . Here we investigate the cases of using two tweak-dependent keys. We first analyze the cases when there are two out of three positions for the tweak, namely  $(b_{12}, a_{12}, a_{22}) \in \{(0, 1, 1), (1, 1, 0), (1, 0, 1)\}$ . Note that for the case when  $(b_{12}, a_{12}, a_{22}) = (1, 0, 1)$ , it uses only one tweak-dependent key and thus is at most birthday-bound secure as shown in the first category in Subsection 3.1.

For  $(b_{12}, a_{12}, a_{22}) = (0, 1, 1)$ , it holds  $\Delta_2 = t_1$  and  $\Delta_3 = t_2$  for a  $2n$ -bit tweak  $t_1 \| t_2$ . By fixing the second tweak  $t_2$ , the variability comes from  $y_1$  which will collide with birthday-bound complexity by changing  $t_1$ . Once a collision happens, the same plaintext for distinct tweaks will lead to the same ciphertext. Based on this observation, the adversary  $\mathcal{A}$  can launch an attack as follows. Firstly, fixing a tweak value  $t_2$  and a plaintext  $m$ ,  $\mathcal{A}$  selects  $2^{n/2}$  distinct tweak values  $t_1^i$  and queries  $(t_1^i \| t_2, m)$  to  $\tilde{E}_k(\cdot, \cdot)$  to search a collision among ciphertexts. Let  $t_1^i$  and  $t_1^j$  denote the corresponding tweaks for the colliding ciphertexts. Secondly,  $\mathcal{A}$  chooses another plaintext  $m'$  with  $m' \neq m$ , queries  $(t_1^i \| t_2, m')$  and  $(t_1^j \| t_2, m')$ , and receives  $c'_i$  and  $c'_j$ , respectively. Finally,  $\mathcal{A}$  outputs 1 if  $c'_i = c'_j$ , and outputs 0 otherwise. The complexity of  $\mathcal{A}$  is  $O(2^{n/2})$ . When interacting with  $\tilde{E}$ ,  $\mathcal{A}$  outputs 1 once she finds a collision at the first step which happens with probability about  $1 - (1 - 2^{-n})^{2^{n-1}} \approx 0.4$ . When interacting with a tweakable random permutation, the probability that  $\mathcal{A}$  outputs 1 is  $2^{-n}$ . Thus, the distinguishing advantage of  $\mathcal{A}$  is  $0.4 - 2^{-n} \approx 0.4$ .

For  $(b_{12}, a_{12}, a_{22}) = (1, 1, 0)$ , it holds  $\Delta_1 = t_1$  and  $\Delta_2 = t_2$ . Similarly, by fixing  $t_1$ , the variability comes from  $y_1$  which will collide with birthday-bound complexity by changing the value of  $t_2$ . The attack procedure is analogous to the above case.

Finally, we analyze the cases when there are three positions for the tweak  $t_1 \| t_2$ , namely  $(b_{12}, a_{12}, a_{22}) = (1, 1, 1)$ . Then we have  $\binom{3}{2}$  possibilities to put two  $n$ -bit tweaks  $t_1$  and  $t_2$  as follows:

- $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_1, t_2)$ . Then by fixing  $t_2$ , the variability comes from  $y_1$  which will collide with birthday-bound complexity by changing the value of  $t_1$ . The attack is similar to the above cases.
- $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_2, t_1)$ . Similarly, by fixing  $t_1$ , the variability comes from  $y_1$  and the above attack applies by changing the value of  $t_2$ .
- $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_2, t_2)$ . Then by fixing  $t_2 = 0^n$ , this case is the same as  $(b_{12}, a_{12}, a_{22}) = (1, 0, 0)$  that is covered in the case of  $(b_{12}, a_{12}) = (1, 0)$ . Wang et al. found that only 24 TBCs satisfying the condition  $(b_{12}, a_{12}) = (1, 0)$  can achieve  $n$ -bit security with  $n$ -bit tweak while the rest of them have the security up to birthday bound.<sup>4</sup> Hence, we can merely consider these 24 TBCs with constraints  $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_2, t_2)$  and  $(b_{12}, a_{12}, a_{22}) = (1, 1, 1)$ . These 24 TBCs are illustrated in Figure 6 and Figure 7 in Appendix A. As shown in following analyses, all of them are subject to birthday-bound attacks.

REMARK. Similar to Type I constructions, we can also use  $t_1 \oplus t_2$  in one of these three positions that results in three more non-trivial possibilities  $(\Delta_1, \Delta_2, \Delta_3) \in \{(t_1 \oplus t_2, t_1, t_2), (t_1, t_1 \oplus t_2, t_2), (t_1, t_2, t_1 \oplus t_2)\}$ . Yet, similar attacks as above apply to these possibilities. For instance, for the case  $(\Delta_1, \Delta_2, \Delta_3) = (t_1 \oplus t_2, t_1, t_2)$ , by fixing  $t_2 = 0^n$ , the variability comes from  $y_1$  that will collide with birthday-bound complexity by changing the value of  $t_1$ .

SECURITY OF THE REMAINING 24 TBCs. We now investigate the security of the 24 TBCs (from  $\tilde{T}5$  to  $\tilde{T}28$ ) illustrated in Figure 6 and Figure 7. For TBCs from  $\tilde{T}5$  to  $\tilde{T}8$ , the attack is similar to that of  $\tilde{T}1$  and based on the following observation: by fixing the

<sup>4</sup> With the constraint that  $(a_{22}, a_{23}) \neq (0, 0)$ . Otherwise they are at most birthday-bound secure.

second tweak  $t_2$ , if  $m_i \oplus y_1^i = m_j \oplus y_1^j$ , then  $c_i \oplus c_j = y_1^i \oplus y_1^j = m_i \oplus m_j$  that happens with query complexity  $2^{n/2}$  by changing both  $t_1$  and  $m$ .

For TBCs from  $\tilde{T}9$  to  $\tilde{T}13$ , the variability comes from the subkey that depends on  $y_1 \oplus t_2$  which will collide with birthday bound complexity by changing  $t_2$ . Once a collision happens on the subkey, the same plaintext for distinct tweaks will lead to the same ciphertext. Thus, the adversary can mount an attack as follows (we take  $\tilde{T}9$  as an example, and similar attacks apply to other TBCs). Firstly, fixing a tweak value  $t_1$  and a plaintext  $m$ ,  $\mathcal{A}$  selects  $2^{n/2}$  distinct tweak values  $t_2^i$  and queries  $(t_1 \parallel t_2^i, m)$  to  $\tilde{T}9(\cdot, \cdot)$  to search a collision among ciphertexts. Let  $t_2^i$  and  $t_2^j$  be the corresponding tweaks for the colliding ciphertexts. Secondly,  $\mathcal{A}$  chooses another plaintext  $m'$  with  $m' \neq m$ , and queries  $(t_1 \parallel t_2^i, m')$  and  $(t_1 \parallel t_2^j, m')$  to receive  $c'_i$  and  $c'_j$ , respectively. Finally,  $\mathcal{A}$  outputs 1 if  $c'_i = c'_j$ , and outputs 0 otherwise. The complexity of  $\mathcal{A}$  is  $O(2^{n/2})$  queries. The probability that  $\mathcal{A}$  outputs 1 when interacting with  $\tilde{T}9$  is about  $1 - (1 - 2^{-n})^{2^{n-1}} \approx 0.4$ , while the probability that  $\mathcal{A}$  outputs 1 when interacting with a tweakable random permutation is  $2^{-n}$ . Hence, the distinguishing advantage of  $\mathcal{A}$  is around 0.4.

For TBCs from  $\tilde{T}14$  to  $\tilde{T}23$ , either the input or the output of the second block cipher is additionally masked by  $y_1$ . Hence, we cannot simply detect the subkey collision by checking ciphertexts. However, we can check the XOR of either two plaintexts or two ciphertexts to detect the subkey collision. Take  $\tilde{T}14$  as an example. First, fixing a tweak value  $t_1$  and two ciphertexts  $c$  and  $c'$ , adversary  $\mathcal{A}$  selects  $2^{n/2}$  distinct tweak values  $t_2^i$  and queries  $(t_1 \parallel t_2^i, c)$  and  $(t_1 \parallel t_2^i, c')$  to  $\tilde{T}14^{-1}(\cdot, \cdot)$  to search a collision on  $m_i \oplus m'_i$ . Let  $t_2^i$  and  $t_2^j$  be the corresponding tweaks for this collision, namely  $y_1^i \oplus t_2^i = y_1^j \oplus t_2^j$  that implies  $m_i \oplus m'_i = m_j \oplus m'_j$ . Secondly,  $\mathcal{A}$  selects another two different ciphertexts  $\bar{c}$  and  $\bar{c}'$  with  $\bar{c}, \bar{c}' \notin \{c, c'\}$ , queries  $(t_1 \parallel t_2^i, \bar{c})$ ,  $(t_1 \parallel t_2^i, \bar{c}')$ ,  $(t_1 \parallel t_2^j, \bar{c})$  and  $(t_1 \parallel t_2^j, \bar{c}')$  to  $\tilde{T}14^{-1}(\cdot, \cdot)$ , and receives  $\bar{m}_i, \bar{m}'_i, \bar{m}_j$ , and  $\bar{m}'_j$ , respectively. Finally,  $\mathcal{A}$  outputs 1 if  $\bar{m}_i \oplus \bar{m}'_i = \bar{m}_j \oplus \bar{m}'_j$ . The complexity and advantage of  $\mathcal{A}$  are around  $O(2^{n/2})$  and 0.4, respectively.

For TBCs from  $\tilde{T}24$  to  $\tilde{T}28$ , both the input and the output of the second block cipher are masked by  $y_1$ . Hence, to find a collision on outputs of the second block cipher, it requires both subkey collision and input collision which may need  $2^n$  queries complexity at the first glance. Yet, since two tweaks  $t_1$  and  $t_2$  are arbitrarily chosen by the adversary, eventually these two collisions can be degenerated to one collision as shown below. Take  $\tilde{T}24$  as an example. For a pair of queries  $(t_1^i \parallel t_2^i, m_i, c_i)$  and  $(t_1^j \parallel t_2^j, m_j, c_j)$ , the subkey collision requires  $y_1^i \oplus t_2^i = y_1^j \oplus t_2^j$  and the input collision requires  $m_i \oplus y_1^i = m_j \oplus y_1^j$ . If  $m_i = t_2^i$  and  $m_j = t_2^j$ , then the above two equations are degenerated to one equation  $y_1^i \oplus y_1^j = t_2^i \oplus t_2^j$ . Once this collision happens, it is detectable by checking  $m_i \oplus m_j = c_i \oplus c_j = y_1^i \oplus y_1^j$ . Hence the adversary  $\mathcal{A}$  can launch an attack as follows. Firstly, fixing a tweak value  $t_1$ ,  $\mathcal{A}$  selects  $2^{n/2}$  distinct tweak values  $t_2^i$  and queries  $(t_1 \parallel t_2^i, m_i)$  where  $m_i = t_2^i$  to  $\tilde{T}24(\cdot, \cdot)$  to search a match  $m_i \oplus m_j = c_i \oplus c_j$ . Let  $(t_1 \parallel t_2^i, m_i, c_i)$  and  $(t_1 \parallel t_2^j, m_j, c_j)$  be the corresponding pairs for this match. Secondly,  $\mathcal{A}$  selects a constant value  $\Delta \neq 0^n$  and queries  $(t_1 \parallel t_2^i, m_i \oplus \Delta)$  and  $(t_1 \parallel t_2^j, m_j \oplus \Delta)$  to  $\tilde{T}24$  to receive  $c'_i$  and  $c'_j$ , respectively. Finally,  $\mathcal{A}$  outputs 1 if  $c'_i \oplus c'_j = m_i \oplus m_j$ , and outputs 0 otherwise. The complexity and advantage of  $\mathcal{A}$  are around  $O(2^{n/2})$  and 0.4 respectively.

## 4 Three Block Cipher Calls

In this section, we study how to build a  $2n$ -bit tweak TBC with  $n$ -bit security from three block cipher calls. We do not rely on the exhaustive search as adopted by Wang et al. [WGZ<sup>+</sup>16] for two block cipher calls since the number of possible TBCs from three block cipher calls grows exponentially (the total number is around  $2^{34}$ ) and it is hard to investigate them completely. Instead, we begin with some important conditions to build a

$2n$ -bit tweak TBC from three block cipher calls and propose a construction called  $\tilde{G}1$  that likely meets them. Interestingly, after an in-depth investigation, we find some weakness in this construction and consequently propose an attack with birthday-bound complexity against it. Based on  $\tilde{G}1$ , we propose another construction called  $\tilde{G}2$  that can fix this weakness and is provably  $n$ -bit secure. We believe there may be other constructions with the same  $n$ -bit security, but arguably  $\tilde{G}2$  is one of the most efficient constructions among them since (i) the first two block cipher calls can be computed in parallel; (ii) it only requires one tweak-dependent key.

#### 4.1 Some Possible Conditions for $n$ -bit Security

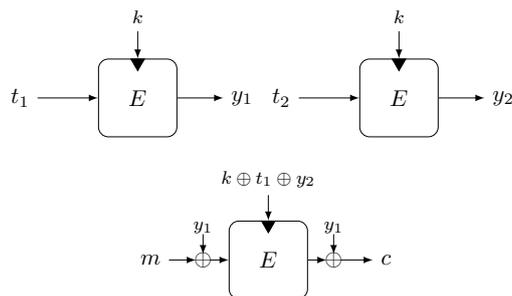
Recall that the main idea from [Men15, WGZ<sup>+</sup>16] to build an  $n$ -bit tweak TBC with  $n$ -bit security from two block cipher calls can be summarized as follows: the first block cipher call with a fixed key  $k$  is used to derive an internal value  $y_1$  that will be used as either a subkey or a mask for the second block cipher call; then the second block cipher call is invoked to encrypt the plaintext. To prove the security of these TBCs, we essentially need to consider the influence of offline computations that is captured by  $p$  (the number of ideal-cipher queries to  $E$  and  $E^{-1}$ ), and the influence of online queries that is captured by  $q$  (the number of construction queries to the targeted TBC and its backward counterpart). The reason why these TBCs can achieve  $n$ -bit security is generally based on the following observations: (i) since the first block cipher uses a fixed key, if the adversary wants to attack this part, then it requires to guess the correct key that happens with probability  $p/2^n$  by using  $p$  ideal-cipher queries; (ii) for the second block cipher, the impact from ideal-cipher queries can be bounded by  $pq/2^{2n}$  since a collision between this block cipher call and an ideal-cipher query requires both collisions on key and input that are masked by  $k$  and  $y_1$ ; (iii) for the second block cipher, the impact from construction queries is easy to argue since a collision between this block cipher call and another second block cipher call requires both collisions on key and input and there is no hope to have a collision on the subkey since changing the tweak will make the subkey different.

Although it is impossible to build a  $2n$ -bit tweak TBC with  $n$ -bit security from two block cipher calls as illustrated in Section 3, the idea is generically useful. Hence, we generalize this idea to build a  $2n$ -bit TBC with  $n$ -bit security from three block cipher calls: the first two block cipher calls are invoked to derive two internal values  $y_1$  and  $y_2$  that may be used as either a subkey or a mask for the third block cipher call; then the third block cipher call is invoked to encrypt the plaintext. Following this idea, we propose two natural constructions called  $\tilde{G}1$  and  $\tilde{G}2$  below. Note that  $\tilde{G}1$  is subject to a birthday-bound attack, but serves as a stepstone for the design of  $\tilde{G}2$ , as it is helpful to understand why  $\tilde{G}2$  can achieve  $n$ -bit security.

#### 4.2 The First Construction $\tilde{G}1$

**SCHEME DESCRIPTION.** Let  $E : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$  be a block cipher. The TBC  $\tilde{G}1 : \{0, 1\}^n \times \{0, 1\}^{2n} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$  is built from a block cipher  $E$  as follows. Two block cipher calls are first invoked in parallel to produce two masks  $y_1$  and  $y_2$  from the tweaks  $t_1, t_2$  and the master key  $k$ . By using  $y_1$  to mask both the input and the output, using  $y_2$  and  $t_1$  to provide variety in the subkey, a third block cipher call is then invoked to encrypt the message  $m$  to the ciphertext  $c$ . The construction  $\tilde{G}1$  is illustrated in Figure 4. We next discuss the security of  $\tilde{G}1$  and show why it is at most birthday-bound secure.

**DISCUSSION AND ATTACK.** At the first glance, construction  $\tilde{G}1$  satisfies the above conditions since (i) the first two block cipher calls use a fixed key  $k$  and thus it is hard for the adversary to attack this part; (ii) for the third block cipher call, the key and the input are

Figure 4: The first TBC construction  $\tilde{G}1$  based on three block cipher calls.

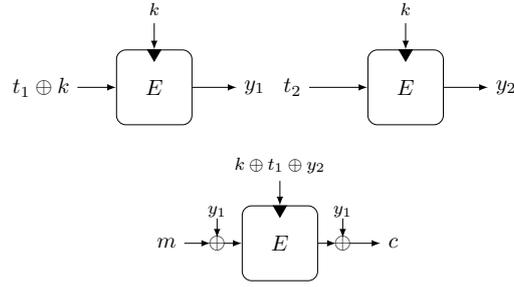
both masked by  $k$ ,  $y_1$  and  $y_2$ , and hence a collision between this block cipher call and an ideal-cipher query requires collisions on both the key and input that happens with probability around  $pq/2^{2n}$ ; (iii) for the third block cipher call, the influence from construction queries is also limited since both the key and the input are masked and changing any one of two tweaks  $t_1$  and  $t_2$  will also change the subkey  $k \oplus t_1 \oplus y_2$ . However, after a closer inspection, we find the influence of point (iii) is more than expected because the two collisions on the key and the input can be degenerated to one collision. In the following, we present an attack with birthday-bound complexity.

The attack is based on the following fact. For two construction queries  $(t_1^i \parallel t_2^i, m_i, c_i)$  and  $(t_1^j \parallel t_2^j, m_j, c_j)$ , if the key collision  $k \oplus t_1^i \oplus y_2^i = k \oplus t_1^j \oplus y_2^j$  and the input collision  $m_i \oplus y_1^i = m_j \oplus y_1^j$  happen, then the outputs satisfy  $c_i \oplus c_j = y_1^i \oplus y_1^j$  which is detectable by checking  $c_i \oplus c_j = m_i \oplus m_j$ . Since the adversary can choose both  $t_1$  and  $t_2$ , by fixing  $m_i = t_1^i$ ,  $m_j = t_1^j$ ,  $t_1^i = t_2^i$  and  $t_1^j = t_2^j$  that implies  $y_1^i = y_2^i$  and  $y_1^j = y_2^j$ , the above two collisions will be reduced to one equation  $y_1^i \oplus y_1^j = t_1^i \oplus t_1^j$  that happens with query complexity  $2^{n/2}$  by changing  $t_1$ . Hence, the adversary  $\mathcal{A}$  can mount an attack against  $\tilde{G}1$  as follows. Firstly,  $\mathcal{A}$  selects  $2^{n/2}$  distinct tweak values  $t_1^i$ , and queries  $(t_1^i \parallel t_1^i, t_1^i)$  to  $\tilde{G}1(\cdot, \cdot)$  to search a match  $c_i \oplus c_j = t_1^i \oplus t_1^j$ . Let  $(t_1^i \parallel t_1^i, t_1^i, c_i)$  and  $(t_1^j \parallel t_1^j, t_1^j, c_j)$  be the corresponding pair for this match. Secondly,  $\mathcal{A}$  selects a constant value  $\Delta \neq 0^n$ , and queries  $(t_1^i \parallel t_1^i, t_1^i \oplus \Delta)$  and  $(t_1^j \parallel t_1^j, t_1^j \oplus \Delta)$  to  $\tilde{G}1(\cdot, \cdot)$  to receive  $c'_i$  and  $c'_j$  respectively. Finally,  $\mathcal{A}$  outputs 1 if  $c'_i \oplus c'_j = t_1^i \oplus t_1^j$ , and outputs 0 otherwise. The complexity of  $\mathcal{A}$  is  $O(2^{n/2})$ . When interacting with  $\tilde{G}1$ ,  $\mathcal{A}$  outputs 1 as long as she successfully finds a match at the first step that has a probability of about  $1 - (1 - 2^{-n})^{2^{n-1}} \approx 0.4$ . When interacting with a tweakable random permutation, the probability that  $\mathcal{A}$  outputs 1 is  $2^{-n}$ . Hence, the distinguishing advantage of  $\mathcal{A}$  is around 0.4.

### 4.3 The Second Construction $\tilde{G}2$

Note that the above attack against  $\tilde{G}1$  essentially relies on the fact that if  $t_1^i = t_2^i$ , then  $y_1^i = y_2^i$ . Otherwise the above two equations  $t_1^i \oplus y_2^i = t_1^j \oplus y_2^j$  and  $m_i \oplus y_1^i = m_j \oplus y_1^j$  would not be reduced to one equation since they involve two different variables  $y_1^i$  and  $y_2^i$  in each equation. In the following, we propose another construction called  $\tilde{G}2$  that can fix this weakness and is provably  $n$ -bit secure. We begin with the description of  $\tilde{G}2$  and then present its security analysis.

**SCHEME DESCRIPTION.** Let  $E : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$  be a block cipher. The TBC  $\tilde{G}2 : \{0, 1\}^n \times \{0, 1\}^{2n} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$  with a  $2n$ -bit tweak is constructed as follows. Two block cipher calls are first invoked in parallel to produce two masks  $y_1$  and  $y_2$  from the tweaks  $t_1, t_2$  and the master key  $k$ . By using  $y_1$  to mask both the input and output, using  $y_2$  and  $t_1$  to provide variety in the subkey, a third block cipher call is then invoked

Figure 5: The second TBC construction  $\tilde{G}2$  based on three block cipher calls.

to encrypt the message  $m$  to the ciphertext  $c$ . A pictorial illustration of  $\tilde{G}2$  is given in Figure 5. Note that in  $\tilde{G}2$ , the input  $t_1$  to the first block cipher call is XORed with  $k$  and thus the adversary cannot trivially make  $y_1^i = y_2^i$  that avoids the weakness in  $\tilde{G}1$ .

**SECURITY ANALYSIS OF  $\tilde{G}2$ .** We define the queries to  $\tilde{G}2$  and  $\tilde{G}2^{-1}$  as construction queries, and the queries to  $E$  and  $E^{-1}$  as ideal-cipher queries. Assuming that  $E$  is an ideal block cipher, the following theorem shows  $\tilde{G}2$  is a strong tweakable pseudorandom permutation.

**Theorem 1.** *Let  $\mathcal{A}$  be an adversary making at most  $q$  construction queries and  $p$  ideal-cipher queries, including both forward and backward queries. Then,*

$$\text{Adv}_{\tilde{G}2}^{\text{sprp}}(\mathcal{A}) \leq \frac{p + 2q}{2^n} + \frac{4pq + 12q^2}{2^{2n}}$$

by assuming  $q \leq 2^{n-2}$ .

**DISCUSSION AND OVERVIEW OF THE PROOF.** The security bound implies that as long as the number of ideal-cipher queries does not exceed  $O(2^n)$  and the number of construction queries does not exceed  $O(2^n)$ ,  $\tilde{G}2$  behaves like a tweakable random permutation. The assumption of  $q \leq 2^{n-2}$  is used to upper bound some probabilities in the proof.

The proof is based on the following observations. Since the first two block cipher calls use a fixed key  $k$ , it is hard for the adversary to attack this part unless she guesses correctly the key that happens with probability  $p/2^n$ . For the third block cipher call, we need to consider collisions from ideal-cipher queries and construction queries. Since the key and the input of the third block cipher call are both masked either as  $k \oplus t_1 \oplus y_2$  or by  $y_1$ , a collision between this block cipher call and an ideal-cipher query requires collisions on these two positions that happen with probability around  $pq/2^{2n}$ . For a collision with other construction queries, it also requires collisions on both the key and input that can be written as two equations  $t_1^i \oplus y_2^i = t_1^j \oplus y_2^j$  and  $m_i \oplus y_1^i = m_j \oplus y_1^j$ . Here, we should distinguish two cases depending on whether  $t_1^i \oplus k = t_2^i$ . If  $t_1^i \oplus k \neq t_2^i$ , then  $y_1^i$  and  $y_2^i$  behave like two independent variables and these two equations have a rank of 2. If  $t_1^i \oplus k = t_2^i$  and  $t_1^j \oplus k = t_2^j$ , then the above two equations will be degenerated to one equation  $t_1^i \oplus t_1^j = y_1^i \oplus y_1^j$  but the collision on  $t_1^i \oplus k = t_2^i$  and  $t_1^j \oplus k = t_2^j$  already counts as at least one equation. Hence, in both two cases, the collision requires two equations that happens with probability around  $q^2/2^{2n}$ . More details can be found in the formal proof.

*Proof.* We consider a computationally unbounded and thus deterministic adversary  $\mathcal{A}$ . Without loss of generality, we assume that the adversary  $\mathcal{A}$  does not make redundant queries, i.e., neither repeating a prior query nor making forward and backward queries that will result in the same query-response tuple. We will use the H-coefficient technique

as introduced in Section 2. Here the real system corresponds to the world when  $\mathcal{A}$  has bidirectional access to the oracles  $(\tilde{G}2, E)$  where  $k \xleftarrow{\$} \{0, 1\}^n$  and  $E \xleftarrow{\$} \text{BC}(\{0, 1\}^n, \{0, 1\}^n)$ , and the ideal system corresponds to the world when  $\mathcal{A}$  has bidirectional access to the oracles  $(\tilde{\pi}, E)$  where  $\tilde{\pi} \xleftarrow{\$} \widetilde{\text{Perm}}(\{0, 1\}^{2n}, \{0, 1\}^n)$  and  $E \xleftarrow{\$} \text{BC}(\{0, 1\}^n, \{0, 1\}^n)$ .

SETUP. In the real world, after the adversary finishes querying, we disclose the master key  $k$  and masks  $y_1^i$  and  $y_2^i$  to it. In the ideal world, we instead disclose to the adversary a truly random string  $k \xleftarrow{\$} \{0, 1\}^n$  that is independent of her queries and the corresponding masks  $y_1^i$  and  $y_2^i$  by invoking  $E$ . Thus the transcript implicitly includes the ideal-cipher queries  $y_1^i \leftarrow E_k(t_1^i \oplus k)$  and  $y_2^i \leftarrow E_k(t_2^i)$ . This additional information can only help the adversary. Hence a transcript consists of the revealed key, the granted ideal-cipher queries, and the following information:

- **Construction queries.** The adversary makes at most  $q$  queries to oracle  $(O_1, O_2) \in \{(\tilde{G}2, \tilde{G}2^{-1}), (\tilde{\pi}, \tilde{\pi}^{-1})\}$ , and these are recorded by entries  $(\text{con}, t_1^1 \parallel t_2^1, m_1, c_1), \dots, (\text{con}, t_1^q \parallel t_2^q, m_q, c_q)$  where  $t_1^i \parallel t_2^i$  is the tweak,  $m_i$  the plaintext and  $c_i$  the ciphertext.
- **Ideal-cipher queries.** The adversary makes at most  $p$  queries to oracle  $(O_3, O_4) = (E, E^{-1})$ , and these are recorded by entries  $(\text{prim}, \ell_1, u_1, v_1), \dots, (\text{prim}, \ell_p, u_p, v_p)$  where  $\ell_i$  is the key,  $u_i$  the plaintext and  $v_i$  the ciphertext.

DEFINING BAD TRANSCRIPT. We say a transcript is *bad* if one of the following conditions is violated:

1. There is an entry  $(\text{prim}, \ell, u, v)$  such that  $\ell = k$ . Eliminating this case avoids potential inconsistency due to guessing correctly the master key  $k$ .
2. There are two entries  $(\text{con}, t_1 \parallel t_2, m, c)$  and  $(\text{prim}, \ell, u, v)$  such that  $m \oplus y_1 = u$  and  $k \oplus t_1 \oplus y_2 = \ell$ . This will force  $c \oplus y_1 = v$  in the real world, while there is no such constraint in the ideal world.
3. There are two entries  $(\text{con}, t_1 \parallel t_2, m, c)$  and  $(\text{prim}, \ell, u, v)$  such that  $c \oplus y_1 = v$  and  $k \oplus t_1 \oplus y_2 = \ell$ . Similarly, this will force  $m \oplus y_1 = u$  in the real world, while there is no such constraint in the ideal world.
4. There is an entry  $(\text{con}, t_1 \parallel t_2, m, c)$  such that  $k \oplus t_1 \oplus y_2 = k$ . Eliminating this case avoids potential inconsistency due to the collision between master key and subkey.
5. There are two entries  $(\text{con}, t_1^i \parallel t_2^i, m_i, c_i)$  and  $(\text{con}, t_1^j \parallel t_2^j, m_j, c_j)$  with  $i \neq j$  such that  $m_i \oplus y_1^i = m_j \oplus y_1^j$  and  $k \oplus t_1^i \oplus y_2^i = k \oplus t_1^j \oplus y_2^j$ . This will force  $c_i \oplus y_1^i = c_j \oplus y_1^j$  in the real world, while there is no such constraint in the ideal world.
6. There are two entries  $(\text{con}, t_1^i \parallel t_2^i, m_i, c_i)$  and  $(\text{con}, t_1^j \parallel t_2^j, m_j, c_j)$  with  $i \neq j$  such that  $c_i \oplus y_1^i = c_j \oplus y_1^j$  and  $k \oplus t_1^i \oplus y_2^i = k \oplus t_1^j \oplus y_2^j$ . Similarly, this will force  $m_i \oplus y_1^i = m_j \oplus y_1^j$  in the real world, while there is no such constraint in the ideal world.

If a transcript is not bad and is attainable in the ideal world, then we say it is *good*. Denote by  $X_1$  and  $X_0$  the random variables for the transcript in the real and ideal worlds respectively.

PROBABILITY OF BAD TRANSCRIPTS. We now bound the probability that  $X_0$  is bad in the ideal world. For  $1 \leq i \leq 6$ , we denote by  $\text{bad}_i$  the event that the  $i$ th condition is

violated. By the union bound,

$$\Pr[X_0 \text{ is bad}] = \Pr[\text{bad}_1 \vee \dots \vee \text{bad}_6] \leq \sum_{i=1}^6 \Pr[\text{bad}_i] .$$

We first bound the probability of event  $\text{bad}_1$ . Recall that in the ideal world,  $k$  is a uniformly random string. Hence the probability that  $\ell = k$  is  $1/2^n$ . Summing over at most  $p$  ideal-cipher queries,

$$\Pr[\text{bad}_1] \leq \frac{p}{2^n} .$$

Next, we bound the probability of event  $\text{bad}_2$ . Since  $y_1$  is distributed uniformly at random in a set of size at least  $2^n - 2q$  and  $k$  is an  $n$ -bit uniformly random string, we have

$$\Pr[\text{bad}_2] \leq \frac{pq}{2^n(2^n - 2q)} \leq \frac{2pq}{2^{2n}}$$

by assuming  $q \leq 2^{n-2}$ .

The analysis of event  $\text{bad}_3$  is similar to that of event  $\text{bad}_2$ , and we have

$$\Pr[\text{bad}_3] \leq \frac{2pq}{2^{2n}} .$$

We then analyze the probability of event  $\text{bad}_4$ . This event requires that  $t_1 \oplus y_2 = 0$ . Since  $y_2$  is selected uniformly at random from a set of size at least  $2^n - 2q$ , we have

$$\Pr[\text{bad}_4] \leq \frac{q}{2^n - 2q} \leq \frac{2q}{2^n}$$

by assuming  $q \leq 2^{n-2}$ .

We then bound the probability of event  $\text{bad}_5$ . This event can be rewritten as  $y_1^i \oplus y_1^j = m_i \oplus m_j$  and  $y_2^i \oplus y_2^j = t_1^i \oplus t_1^j$ . We consider four cases according to the value of  $t$ weak:

- **Case 1:**  $t_1^i \parallel t_2^i = t_1^j \parallel t_2^j$ , and thus  $m_i \neq m_j$ . Then the first equation cannot hold since  $m_i \oplus m_j \neq 0$ .
- **Case 2:**  $t_1^i = t_1^j$  and  $t_2^i \neq t_2^j$ . Then the second equation cannot hold since  $y_2^i \oplus y_2^j \neq 0$ .
- **Case 3:**  $t_1^i \neq t_1^j$  and  $t_2^i = t_2^j$ . Then the second equation cannot hold since  $t_1^i \oplus t_1^j \neq 0$ .
- **Case 4:**  $t_1^i \neq t_1^j$  and  $t_2^i \neq t_2^j$ . We further discuss three sub-cases:
  - **4.1:**  $t_1^i \oplus k = t_2^i$  and  $t_1^j \oplus k = t_2^j$ , which happens with probability at most  $1/2^n$  since  $k$  is a uniformly random string. Then the above two equations become  $y_1^i \oplus y_1^j = m_i \oplus m_j$  and  $y_1^i \oplus y_1^j = t_1^i \oplus t_1^j$ , which happen with probability at most  $1/(2^n - 2q)$  since both  $y_1^i$  and  $y_1^j$  are selected uniformly at random from a set of size at least  $2^n - 2q$ . By summing over at most  $\binom{q}{2}$  pairs of construction queries, the probability corresponding to this sub-case is at most

$$\frac{q(q-1)/2}{2^n(2^n - 2q)} \leq \frac{q^2}{2^{2n}}$$

by assuming  $q \leq 2^{n-2}$ .

- **4.2:**  $t_1^i \oplus k = t_2^j$  and  $t_1^j \oplus k = t_2^i$ , which happens with probability at most  $1/2^n$  since  $k$  is a uniformly random string. Similarly, the above two equations become  $y_1^i \oplus y_1^j = m_i \oplus m_j$  and  $y_1^i \oplus y_1^j = t_1^i \oplus t_1^j$  which happen with probability at most  $1/(2^n - 2q)$ . Hence the probability corresponding to this sub-case is at most  $q^2/2^{2n}$  by assuming  $q \leq 2^{n-2}$ .

- **4.3:** Neither Case 4.1 nor Case 4.2 happens. Then either  $y_2^i \notin \{y_1^i, y_1^j\}$  or  $y_2^j \notin \{y_1^i, y_1^j\}$ , and thus the above two equations will not be degenerated since each of two equations contains at least one unique variable. In the case that  $y_2^i \notin \{y_1^i, y_1^j\}$ , the probability that these two equations hold is at most  $1/(2^n - 2q)^2$  since both  $y_1^i$  and  $y_2^i$  are selected uniformly at random from a set of size at least  $2^n - 2q$ . Similar argument holds for the case  $y_2^j \notin \{y_1^i, y_1^j\}$ . By summing over at most  $\binom{q}{2}$  pairs of construction queries, we obtain a bound

$$2 \cdot \frac{q(q-1)/2}{(2^n - 2q)^2} \leq \frac{4q^2}{2^{2n}}$$

by assuming  $q \leq 2^{n-2}$ .

Summing up,

$$\Pr[\text{bad}_5] \leq \frac{6q^2}{2^{2n}}$$

by assuming  $q \leq 2^{n-2}$ .

The analysis of event  $\text{bad}_6$  is similar to that of event  $\text{bad}_5$ , and hence

$$\Pr[\text{bad}_6] \leq \frac{6q^2}{2^{2n}}.$$

Thus totally,

$$\Pr[X_0 \text{ is bad}] \leq \frac{p+2q}{2^n} + \frac{4pq+12q^2}{2^{2n}}. \quad (1)$$

**RATIO OF GOOD TRANSCRIPT.** We now analyze the ratio for any good transcript  $\tau$ . Let  $S_1(k)$  be the set of granted ideal-cipher entries  $\{(k, t_1^1 \oplus k, y_1^1), (k, t_2^1, y_2^1), \dots, (k, t_1^q \oplus k, y_1^q), (k, t_2^q, y_2^q)\}$ . For a key  $K \in \{0, 1\}^n$ , let  $S_2(K)$  be the set  $\{(\ell, u, v) \mid (\text{prim}, \ell, u, v) \in \tau \wedge \ell = K\}$ . Let  $S_3(K)$  be the set  $\{(k \oplus t_1 \oplus y_2, m \oplus y_1, c \oplus y_1) \mid (\text{con}, t_1 \parallel t_2, m, c) \in \tau \wedge k \oplus t_1 \oplus y_2 = K\}$ . For a tweak  $T \in \{0, 1\}^{2n}$ , let  $S_4(T)$  be the set  $\{(t_1 \parallel t_2, m, c) \mid (\text{con}, t_1 \parallel t_2, m, c) \in \tau \wedge t_1 \parallel t_2 = T\}$ . Let  $\alpha(K)$  be the set  $\{t_1 \parallel t_2 \mid (\text{con}, t_1 \parallel t_2, m, c) \in \tau \wedge k \oplus t_1 \oplus y_2 = K\}$ . Then for each key  $K \in \{0, 1\}^n$ , we have

$$\sum_{T \in \alpha(K)} |S_4(T)| = |S_3(K)|$$

and thus

$$\sum_{K \in \{0, 1\}^n} \sum_{T \in \alpha(K)} |S_4(T)| = \sum_{K \in \{0, 1\}^n} |S_3(K)|.$$

Since  $\tau$  is good, any two sets of  $S_1(k)$ ,  $S_2(K)$  and  $S_3(K)$  are disjoint.

Then in the ideal world, since  $\tau$  is good,

$$\begin{aligned} & \Pr[X_0 = \tau] \\ &= 2^{-n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \left( \prod_{K \in \{0, 1\}^n} \prod_{j=0}^{|S_2(K)|-1} \frac{1}{2^n - j} \right) \cdot \left( \prod_{T \in \{0, 1\}^{2n}} \prod_{s=0}^{|S_4(T)|-1} \frac{1}{2^n - s} \right) \\ &= 2^{-n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \left( \prod_{K \in \{0, 1\}^n} \prod_{j=0}^{|S_2(K)|-1} \frac{1}{2^n - j} \right) \cdot \left( \prod_{K \in \{0, 1\}^n} \prod_{T \in \alpha(K)} \prod_{s=0}^{|S_4(T)|-1} \frac{1}{2^n - s} \right) \\ &\leq 2^{-n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \left( \prod_{K \in \{0, 1\}^n} \prod_{j=0}^{|S_2(K)|-1} \frac{1}{2^n - j} \right) \cdot \left( \prod_{K \in \{0, 1\}^n} \prod_{s=0}^{|S_3(K)|-1} \frac{1}{2^n - s} \right) \\ &= 2^{-n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \left( \prod_{K \in \{0, 1\}^n} \prod_{j=0}^{|S_2(K)|-1} \frac{1}{2^n - j} \prod_{s=0}^{|S_3(K)|-1} \frac{1}{2^n - s} \right) \end{aligned}$$

On the other hand, in the real world,

$$\Pr[X_1 = \tau] = 2^{-n} \cdot \prod_{i=0}^{|S_1(k)|-1} \frac{1}{2^n - i} \cdot \prod_{K \in \{0,1\}^n} \prod_{j=0}^{|S_2(K)|+|S_3(K)|-1} \frac{1}{2^n - j}$$

Hence

$$\frac{\Pr[X_1 = \tau]}{\Pr[X_0 = \tau]} \geq 1 \quad (2)$$

since  $\prod_{j=0}^{|S_2(K)|-1} \frac{1}{2^n - j} \prod_{s=0}^{|S_3(K)|-1} \frac{1}{2^n - s} \leq \prod_{j=0}^{|S_2(K)|+|S_3(K)|-1} \frac{1}{2^n - j}$  for each  $K \in \{0,1\}^n$ .

WRAPPING UP. From Lemma 1 with  $\epsilon = 0$ , Equation 1, and Equation 2,

$$\text{Adv}_{\widetilde{G2}}^{\text{sprp}}(\mathcal{A}) \leq \frac{p + 2q}{2^n} + \frac{4pq + 12q^2}{2^{2n}}$$

as claimed.  $\square$

## 5 Conclusion

In this paper, we study the problem of building tweakable block ciphers with tweak size of more than  $n$  bits from merely an  $n$ -bit block cipher. We first show a negative result that all tweakable block ciphers with  $2n$ -bit tweaks built from two block cipher calls are subject to birthday-bound attacks. For this purpose, we perform an exhaustive search for tweakable block ciphers with  $2n$ -bit tweaks from two block cipher calls, and show all of them suffer from birthday-bound attacks. We then investigate the conditions to build a tweakable block cipher with  $n$ -bit security from a block cipher. Based on these conditions, we propose two natural constructions called  $\widetilde{G1}$  and  $\widetilde{G2}$  with  $2n$ -bit tweaks that are both built from three block cipher calls. Although the first construction  $\widetilde{G1}$  is still subject to a birthday-bound attack, it serves as a stepstone for the design of  $\widetilde{G2}$  and is helpful to understand why  $\widetilde{G2}$  can achieve  $n$ -bit security. We then provide a security proof to show that  $\widetilde{G2}$  can achieve  $n$ -bit security with  $2n$ -bit tweak. Following the works of Mennink [Men15] and Wang et al. [WGZ<sup>+</sup>16] that successfully built an  $n$ -bit secure tweakable block cipher with  $n$ -bit tweaks from two block cipher calls, our work goes one step further with respect to the tweak size and suggests that we can also build an  $n$ -bit secure tweakable block cipher with  $2n$ -bit tweaks but requires at least three block cipher calls. A tweakable block cipher with a large tweak is in general more flexible and is helpful for the design of modes of operation. An interesting future work is to consider how to build an  $n$ -bit secure tweakable block cipher with  $tn$ -bit tweaks where  $t > 2$ . We conjecture that it may require at least  $(t + 1)$  block cipher calls for this purpose.

**Acknowledgments.** François-Xavier Standaert is senior research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). This work and its presentation have been funded in parts by the ERC consolidator grant 724725 (acronym SWORD) and the ERC Advanced Grant 101096871 (acronym BRIDGE).

## References

- [Ava17] Roberto Avanzi. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.

- [BGGS20] Zhenzhen Bao, Chun Guo, Jian Guo, and Ling Song. TNT: how to tweak a block cipher. In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, pages 641–673, 2020.
- [BGPS21] Francesco Berti, Chun Guo, Thomas Peters, and François-Xavier Standaert. Efficient leakage-resilient macs without idealized assumptions. In *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part II*, pages 95–123, 2021.
- [BJK<sup>+</sup>16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 123–153, 2016.
- [CJPS22] Benoît Cogliati, Jérémy Jean, Thomas Peyrin, and Yannick Seurin. A long tweak goes a long way: High multi-user security authenticated encryption from tweakable block ciphers. *Cryptology ePrint Archive*, Paper 2022/846, 2022. <https://eprint.iacr.org/2022/846>.
- [Cro00] Paul Crowley. Mercy: A fast large block cipher for disk sector encryption. In *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, pages 49–63, 2000.
- [CS06] Debrup Chakraborty and Palash Sarkar. A general construction of tweakable block ciphers and different modes of operations. In *Information Security and Cryptology, Second SKLOIS Conference, Inscrypt 2006, Beijing, China, November 29 - December 1, 2006, Proceedings*, pages 88–102, 2006.
- [CS08] Debrup Chakraborty and Palash Sarkar. HCH: A new tweakable enciphering scheme using the hash-counter-hash approach. *IEEE Trans. Inf. Theory*, 54(4):1683–1699, 2008.
- [CS14] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 327–350, 2014.
- [Dwo10] Morris Dworkin. Recommendation for block cipher modes of operation: The xts-aes mode for confidentiality on storage devices. 2010.
- [GR17] Dahmun Goudarzi and Matthieu Rivain. How fast can higher-order masking be in software? In *EUROCRYPT (1)*, volume 10210 of *Lecture Notes in Computer Science*, pages 567–597, 2017.
- [HC23] Munawar Hasan and Donghoon Chang. Lynx: Family of lightweight authenticated encryption schemes based on tweakable blockcipher. *Cryptology ePrint Archive*, Paper 2023/241, 2023. <https://eprint.iacr.org/2023/241>.
- [HR03] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 482–499, 2003.

- [HR04] Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, pages 292–304, 2004.
- [HT16] Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 3–32, 2016.
- [IMPS17] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, pages 34–65, 2017.
- [JLM<sup>+</sup>17] Ashwin Jha, Eik List, Kazuhiko Minematsu, Sweta Mishra, and Mridul Nandi. XHX - A framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. In *Progress in Cryptology - LATIN-CRYPT 2017 - 5th International Conference on Cryptology and Information Security in Latin America, Havana, Cuba, September 20-22, 2017, Revised Selected Papers*, pages 207–227, 2017.
- [JN20] Ashwin Jha and Mridul Nandi. Tight security of cascaded LRW2. *J. Cryptol.*, 33(3):1272–1317, 2020.
- [JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, pages 274–288, 2014.
- [JNPS21] Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. The deoxys AEAD family. *J. Cryptol.*, 34(3):31, 2021.
- [KR11] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, pages 306–327, 2011.
- [LL18] ByeongHak Lee and Jooyoung Lee. Tweakable block ciphers secure beyond the birthday bound in the ideal cipher model. In *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, pages 305–335, 2018.
- [LRW02] Moses D. Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable block ciphers. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 31–46, 2002.
- [LS13] Rodolphe Lampe and Yannick Seurin. Tweakable blockciphers with asymptotically optimal security. In *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, pages 133–151, 2013.

- [LST12] Will Landecker, Thomas Shrimpton, and R. Seth Terashima. Tweakable block-ciphers with beyond birthday-bound security. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 14–30, 2012.
- [MCS22] Charles Momin, Gaëtan Cassiers, and François-Xavier Standaert. Handcrafting: Improving automated masking in hardware with manual optimizations. In *COSADE*, volume 13211 of *Lecture Notes in Computer Science*, pages 257–275. Springer, 2022.
- [Men15] Bart Mennink. Optimally secure tweakable blockciphers. In *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, pages 428–448, 2015.
- [MI15] Kazuhiko Minematsu and Tetsu Iwata. Tweak-length extension for tweakable blockciphers. In *Cryptography and Coding - 15th IMA International Conference, IMACC 2015, Oxford, UK, December 15-17, 2015. Proceedings*, pages 77–93, 2015.
- [Min09] Kazuhiko Minematsu. Beyond-birthday-bound security based on tweakable block cipher. In *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, pages 308–326, 2009.
- [MM07] Kazuhiko Minematsu and Toshiyasu Matsushima. Tweakable enciphering schemes from hash-sum-expansion. In *Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings*, pages 252–267, 2007.
- [Nai15] Yusuke Naito. Full prf-secure message authentication code based on tweakable block cipher. In *Provable Security - 9th International Conference, ProvSec 2015, Kanazawa, Japan, November 24-26, 2015, Proceedings*, pages 167–182, 2015.
- [NSS20] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Lightweight authenticated encryption mode suitable for threshold implementation. In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, pages 705–735, 2020.
- [NSS22] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Secret can be public: Low-memory AEAD mode for high-order masking. In *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, pages 315–345, 2022.
- [Pat08] Jacques Patarin. The "coefficients h" technique. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, pages 328–345, 2008.
- [Rog04] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, pages 16–31, 2004.

- [Sar09] Palash Sarkar. Efficient tweakable enciphering schemes from (block-wise) universal hash functions. *IEEE Trans. Inf. Theory*, 55(10):4749–4760, 2009.
- [SO98] Rich Schroeppel and Hilarie Orman. The hasty pudding cipher. *AES candidate submitted to NIST*, page M1, 1998.
- [SPS<sup>+</sup>22] Yaobin Shen, Thomas Peters, François-Xavier Standaert, Gaëtan Cassiers, and Corentin Verhamme. Triplex: an efficient and one-pass leakage-resistant mode of operation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(4):135–162, 2022.
- [ST13] Thomas Shrimpton and R. Seth Terashima. A modular framework for building variable-input-length tweakable ciphers. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, pages 405–423, 2013.
- [Wan] Lei Wang. private communication.
- [WFW05] Peng Wang, Dengguo Feng, and Wenling Wu. HCTR: A variable-input-length enciphering mode. In *Information Security and Cryptology, First SKLOIS Conference, CISC 2005, Beijing, China, December 15-17, 2005, Proceedings*, pages 175–188, 2005.
- [WGZ<sup>+</sup>16] Lei Wang, Jian Guo, Guoyan Zhang, Jingyuan Zhao, and Dawu Gu. How to build fully secure tweakable blockciphers from classical blockciphers. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, pages 455–483, 2016.
- [ZQG22] Zhongliang Zhang, Zhen Qin, and Chun Guo. Just tweak! asymptotically optimal security for the cascaded lrw1 tweakable blockcipher. *Designs, Codes and Cryptography*, pages 1–18, 2022.

## A 24 Tweakable Block Ciphers

We list 24 tweakable block ciphers mentioned in Subsection 3.2. As detailed in Subsection 3.2, all these TBCs admit birthday-bound attacks.

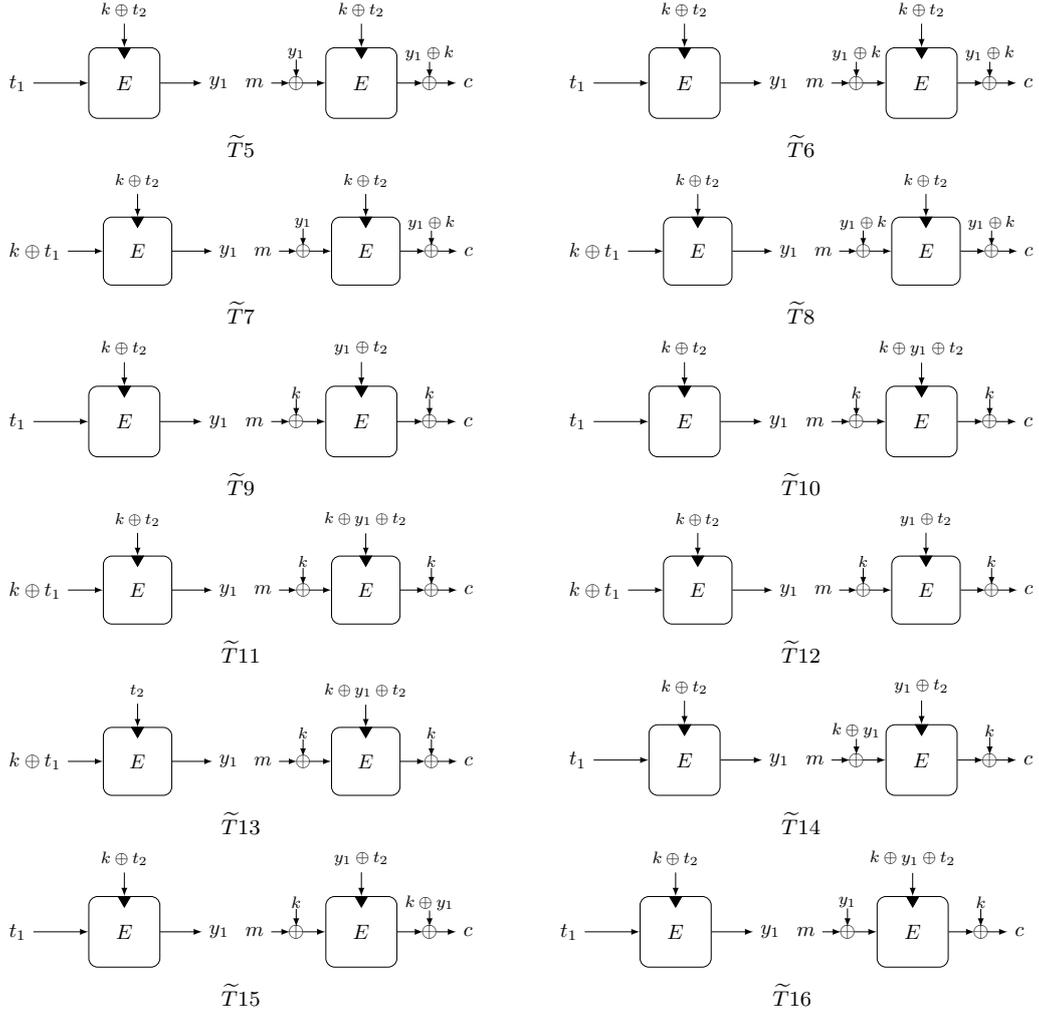


Figure 6:  $\tilde{T}5$  to  $\tilde{T}16$  of the 24 TBCs with constraints  $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_2, t_2)$  and  $(b_{12}, a_{12}, a_{22}) = (1, 1, 1)$  for  $2n$ -bit tweak  $t_1$  and  $t_2$ .

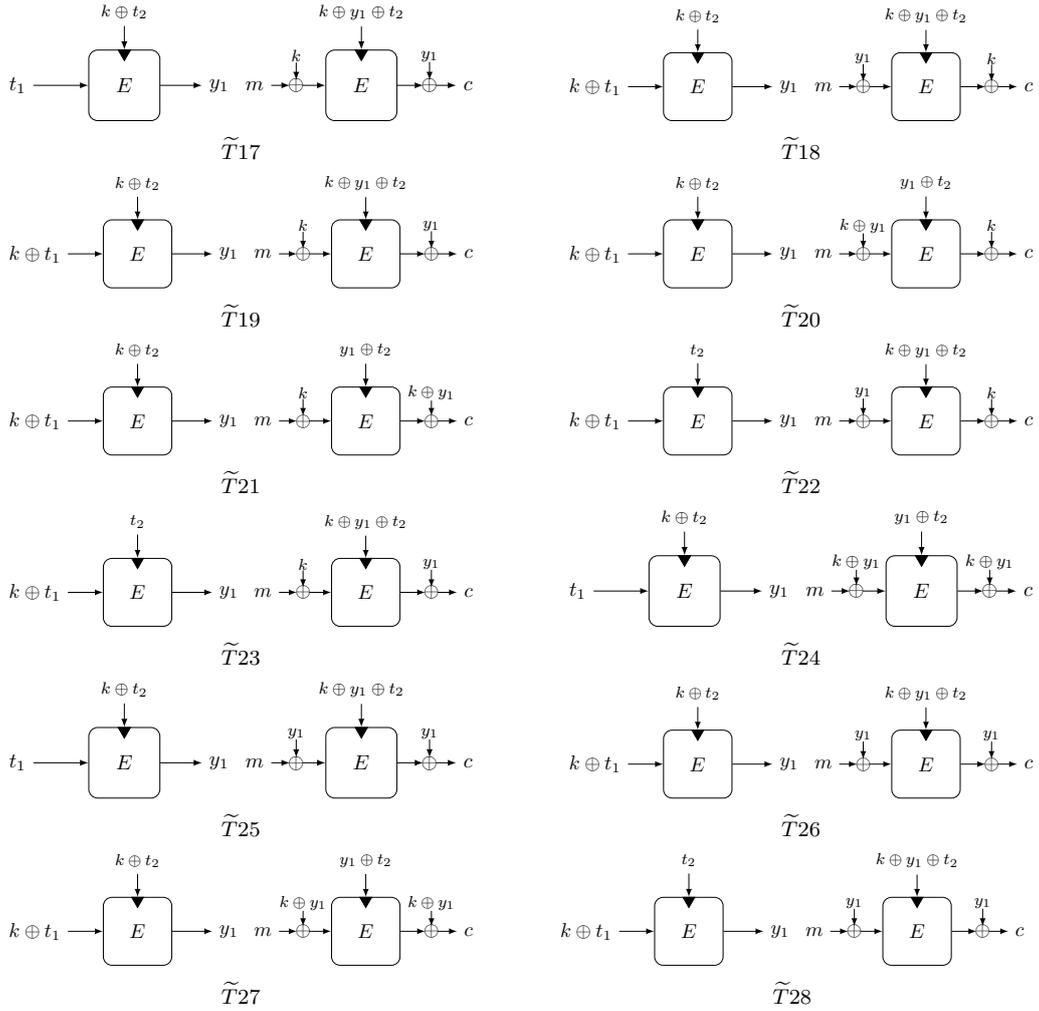


Figure 7:  $\tilde{T}17$  to  $\tilde{T}28$  of the 24 TBCs with constraints  $(\Delta_1, \Delta_2, \Delta_3) = (t_1, t_2, t_2)$  and  $(b_{12}, a_{12}, a_{22}) = (1, 1, 1)$  for  $2n$ -bit tweak  $t_1$  and  $t_2$ .