TNT: How to Tweak a Block Cipher

Zhenzhen Bao[1], Chun Guo[2,3(✉)], Jian Guo[1], and Ling Song[4,5]

[1] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore
{zzbao,guojian}@ntu.edu.sg
[2] Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University, Qingdao, China
[3] School of Cyber Science and Technology, Shandong University, Qingdao, China
chun.guo@sdu.edu.cn
[4] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[5] Jinan University, Guangzhou, China
songling.qs@gmail.com

Abstract. In this paper, we propose Tweak-aNd-Tweak (TNT for short) mode, which builds a tweakable block cipher from three independent block ciphers. TNT handles the tweak input by simply XOR-ing the unmodified tweak into the internal state of block ciphers twice. Due to its simplicity, TNT can also be viewed as a way of turning a block cipher into a tweakable block cipher by dividing the block cipher into three chunks, and adding the tweak at the two cutting points only. TNT is proven to be of beyond-birthday-bound $2^{2n/3}$ security, under the assumption that the three chunks are independent secure n-bit SPRPs. It clearly brings minimum possible overhead to both software and hardware implementations. To demonstrate this, an instantiation named TNT-AES with 6, 6, 6 rounds of AES as the underlying block ciphers is proposed. Besides the inherent proven security bound and tweak-independent rekeying feature of the TNT mode, the performance of TNT-AES is comparable with all existing TBCs designed through modular methods.

Keywords: AES · Tweakable block cipher · χ^2 method · Proof

1 Introduction

1.1 Background - The Need of BBB TBC

Together with the development of authenticated encryption (AE) in CAESAR competition [1] and the on-going lightweight cryptography competition [64], tweakable block ciphers (TBC) are playing a more and more important role. Besides the plaintext, TBCs take a *tweak* as an additional input, which can be viewed as an index to the underlying block cipher, so it becomes a family of (independent) block ciphers v.s. a single instance of block cipher. Its formalization is motivated by the needs of (more than one) independent block ciphers in

A. Canteaut and Y. Ishai (Eds.): EUROCRYPT 2020, LNCS 12106, pp. 641–673, 2020.
https://doi.org/10.1007/978-3-030-45724-2_22

some modes, e.g., OCB [67], while using multiple independent ciphers or keys could cause efficiency issues. In contrast, using a TBC that typically lends itself to very efficient (both software and hardware) implementations, a new instance of block cipher could be obtained by simply choosing a new value of the tweak.

Beyond-Birthday-Bound Security. Most of the current (tweakable) block cipher standards have a block length of 128 bits or less, providing a security level at most 64 bits when instantiated in designs offering only birthday-bound security. Such a security level has become largely inadequate [35]. Even worse, in order to save hardware implementation costs, many lightweight block cipher designs tend to have a smaller block length like 64 bits, providing a birthday security of 32 bits only. Hence, the needs of modes providing BBB security are emerging, and the same has been observed by Gueron and Lindell [35] and in this whitepaper [2].

There are two different ways to construct TBCs. Following the modular approach, they can be built from classical block ciphers via various modular constructions, and security is ensured by a reduction to that of the underlying block ciphers. Alternatively, one could appeal to (probably more efficient) dedicated algorithms, the security guarantees of which come from comprehensive cryptanalysis. Below we'll review both methods.

1.2 Modular Approach: TBCs from Block Ciphers

A classical popular approach is to construct TBCs from existing (traditional) block ciphers in a black-box fashion. Such proposals are further divided into two classes. The "old school" approach, initiated by Liskov *et al.* [54], works in the so-called standard model, models the underlying block cipher as a *pseudorandom permutation*. The "new school" approach recently popularized by Mennink [56] models the block cipher as an *ideal cipher*. The two approaches deviate not only in their security assumptions, but also in their design philosophies. Concretely, standard assumption-based constructions typically tried to avoid tweak-dependent rekeying, which were deemed as (arguably) costly. Another shortage of rekeying is the unavoidable "hybrid security loss" in their security bounds [58,69] (some withstand this loss using carefully-chosen parameters [17,61]). Such a loss doesn't appear in the ideal cipher model, and this is leveraged by many constructions for good bounds and efficiency at the same time. Indeed, ideal cipher-based TBCs have achieved $\geq n$-bit security within 1 or 2 cipher-calls [43,53,77].

In this paper we follow the standard model. In this respect, the original Liskov *et al.*'s paper [54] proposed two constructions that were subsequently named LRW1 and LRW2 by Landecker *et al.* [51]. The former is based on a block cipher E with key space $\mathcal{K}_E$ and message space $\{0,1\}^n$, and is defined as

$$\mathsf{LRW1}((K,K'),T,X) = E_{K'}\big(T \oplus E_K(X)\big). \tag{1}$$

where $(K,K') \in \mathcal{K}_E \times \mathcal{K}_E$ is the key, $T \in \mathcal{T}$ is the tweak, and $X \in \{0,1\}^n$ is the message. Unfortunately it is only CPA secure up to a tight birthday bound, *i.e.*, $2^{n/2}$ adversarial queries. Actually, achieving CCA security was an important

motivation for their second proposal LRW2, which is based on a block cipher E and message space $\{0,1\}^n$ and an almost XOR-universal (AXU) family of hash functions $\mathcal{H} = (H_K)_{K \in \mathcal{K}_H}$ from some set $\mathcal{T}$ to $\{0,1\}^n$, and defined as

$$\mathsf{LRW2}((K,K'),T,X) = H_{K'}(T) \oplus E_K(H_{K'}(T) \oplus X), \tag{2}$$

where $(K,K') \in \mathcal{K}_E \times \mathcal{K}_H$ is the key. This construction was proved CCA secure in [54] up to a tight birthday bound. To seek for beyond-birthday-bound (BBB) secure TBCs, pioneered by Landecker *et al.* [51], subsequent works studied cascade of LRW2 (with independent underlying keys): its 2-cascade was first proved secure up to about $2^{2n/3}$ queries [51] and latter improved to a tight bound of $2^{3n/4}$ queries [44,59], while its r-cascade for general r was proved secure up to roughly $2^{\frac{rn}{r+2}}$ adversarial queries.

A somewhat independent series of works considered tweakable Even-Mansour (TEM) ciphers that are built upon public random permutations [18,20,57], which could also be instantiated with fixed-key block ciphers. It is important to note their security is only provable in the ideal (permutation) model.

1.3 Development of Dedicated TBCs

The Tweakey framework was introduced in 2014 by Jean *et al.* [41], which provides a general guideline for TBC designs. The core idea is to treat the key and tweak in the same way during the primitive design process so that the cryptanalysis can be unified, and becomes simpler than before. So the word "tweakey" is invented to reflect the combined input of tweak and key. Following tweakey framework, various dedicated algorithms such as the Deoxys-BC in the Deoxys AE design [42], SKINNY [7], and Kiasu [40] have been proposed. In detail, SKINNY takes lightweightness into account, and hence makes use of lightweight linear layer—0/1 matrices—almost MDS rather than MDS, although it still follows AES-like design strategy. Up to date, Deoxys is one of the finalists of the CAESAR competition and SKINNY is one of the lightest TBCs in terms of area in the optimized hardware implementations.

When the tweak length is long, TBC-based designs [3,38] can take advantage of its efficiency to process additional input such as associated data. There is also a recent direction of designing TBCs of short tweaks to offer a small family of yet independent block ciphers [12], where tweaks are mainly used as domain separators in the design of authenticated encryption schemes.

It is well-known that, to hide the key of a block cipher, it requires several iterations of the simple round functions. Since Tweakey framework does not distinguish key and tweak, the tweak input has been iterated the same amount of rounds as well. We notice that, rather than hiding, the functionality of a tweak is no more than an index to the block cipher in most of use-cases, and are even assumed to be under attacker's full control in some cryptanalytic settings. Hence, the required level of "protection" for a tweak is essentially lower than that for the key. Inspired by this observation, a natural question to be asked is: what is the minimum number of iterations (or tweak addition) required to produce a secure TBC (especially those with BBB security), with provable security.

1.4 Our Approach (Hybrid of Two Approaches), Provable Security of TBC Modes, and Instantiation with Long-Standing Modules (Similar with AES-PRF)

We seek for an approach slotting between the above two and (hopefully) enjoying the advantages of both, *i.e.*, achieving (some level of) provable guarantees and high efficiency at the same time. Our result is a proposal of a new design of dedicated TBCs based on AES. Our approach is "prove-then-prune", *i.e.*, proving security and then instantiating with a scaled-down primitive (a reduced-round block cipher), that has been used in symmetric designs for a long time, see e.g., [60] (while the terminology was due to Hoang *et al.* [37]). Below we elaborate in detail.

TNT: A New TBC Construction with BBB Security. Our starting point is a new block cipher-based TBC construction with provable BBB security. Concretely, the idealized version of our mode is built upon three secret independent random permutations π_1, π_2, and π_3, and is defined as

$$\mathsf{TNT}^{\pi_1,\pi_2,\pi_3}(T, X) = \pi_3\big(T \oplus \pi_2\big(T \oplus \pi_1(X)\big)\big),$$

as pictured in Fig. 1. We term our mode as TNT, meaning Tweak-aNd-Tweak. It can also be viewed as a cascaded LRW1 TBC construction (if we "split" π_2 into two permutations, then the scheme turns into a cascade of two LRW1 constructions).

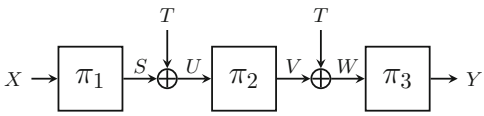

Fig. 1. The $\mathsf{TNT}^{\pi_1,\pi_2,\pi_3}$ mode with the notations (for the intermediate values) used in this paper.

While the original (two-permutation-based) LRW1 construction was proved CPA secure up to birthday $2^{n/2}$ queries and it turns out to be tight, the security of TNT (or cascaded LRW1) remains as a long-standing open problem. In this paper, using the χ^2 technique recently proposed by Dai *et al.* [24], we prove the idealized TNT construction is CCA secure up to BBB $2^{2n/3}$ queries. To our knowledge, this constitutes the first "non-trivial" application of the χ^2 technique to domain expanding constructions, and our proof thus demonstrates relevant issues and their solutions.

We refer to Table 1 for a summary of comparison to existing TBC constructions (we omit the TEM ciphers as they either appear a bit theoretical or are specific for sponges [57]). It is rather difficult to make a comparison with the ideal cipher-based designs [43,53,56,77]. In general, they achieve $\geq n$ bits security (as mentioned) at the expense of a smaller safety margin (similar concern

Table 1. Comparison with previous TBCs. The column ⊗/AXU states if the design relies on AXU hash or field multiplications ⊗. The column tdk states if the design relies on tweak-dependent rekeying. For all the ideal cipher-based designs, we assume using an ideal cipher with n-bit keys and n-bit blocks.

	#tweak	#cost	⊗/AXU?	tdk	security ($\log_2$)
LRW1	n	2 SPRPs	no	no	$n/2$ [54]
XEX	n	1 SPRP	yes	no	$n/2$ [67]
LRW2	arbitrary	1 SPRP	yes	no	$n/2$ [54]
CLRW2$_2$	arbitrary	2 SPRPs	yes	no	$3n/4$ [44,59]
CLRW2$_r$	arbitrary	r SPRPs	yes	no	$rn/(r+2)$ [50]
Min	t	2 SPRPs	no	yes	$\max\{n/2, n-t\}$ [61]
$\widetilde{F}[1]$	n	1 ideal cipher	no	yes	$2n/3$ [56]
$\widetilde{F}[2]$	n	2 ideal ciphers	no	yes	n [56]
$\widetilde{E1}, \ldots, \widetilde{E32}$	n	2 ideal ciphers	no	yes	n [77]
XHX	arbitrary	1 ideal cipher	yes	yes	n [43]
XHX2	arbitrary	2 ideal ciphers	yes	yes	$4n/3$ [53]
TNT	n	3 SPRPs	*no*	no	$2n/3$

has been raised in other settings [36]). Also, their provable bounds should be interpreted with a bit of caution [58]. In terms of efficiency, it is widely believed that tweak-dependent rekeying used in the above designs as well as [61] is a bit costly, particularly when AES-NI is available.

It appears that LRW2 and its cascades are the closest designs. In short, while LRW2 and CLRW2 accept long tweaks, their uses of AXU hash are expected to result in a lower efficiency when n-bit tweaks already suffice. The additional requirement of AXU hash usually results in lower software efficiency and/or higher gate counts as additional registers and operations are needed.

Instantiation from AES. To take the advantage of the AES-NI for better software performance, it is natural for us to instantiate TNT with AES. To further improve the software performance, we reduce the number of rounds of each of the permutations π_1, π_2, and π_3 to 6, 6, and 6 rounds respectively (rather than the full AES itself), which are named TNT-AES. Although, it is not possible to assume the round-reduced AES to be ideal any more, we show, through comprehensive cryptanalysis, the security of TNT-AES are sound. Similar design strategy was introduced by Hoang *et al.* [37] and used in the design of AES-PRF [60] by Mennink and Neves. The estimated performance shows, with help from AES-NI, TNT-AES is among the fastest TBCs in software, and in some cases it can be implemented as light as AES itself in area constrained hardware environment thanks to the simplicity of TNT, smaller than most of the existing TBCs.

Organization. The rest of the paper is organized as follows. Section 2 gives the preliminary necessary for the introduction of the new mode in Sect. 3. The security TNT is proven in Sect. 4. Section 5 proposes a concrete design following TNT based on AES, and finally Sect. 6 concludes the paper.

2 Preliminary

2.1 Notation

For a finite set $\mathcal{X}$, $X \xleftarrow{\$} \mathcal{X}$ denotes selecting an element from $\mathcal{X}$ uniformly at random and $|\mathcal{X}|$ denotes its cardinality.

2.2 TBC and Its Security

A tweakable permutation with tweak space $\mathcal{T}$ and message space $\mathcal{M}$ is a mapping $\widetilde{\Pi} : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ such that for any tweak $T \in \mathcal{T}$, $X \mapsto \widetilde{\Pi}(T, X)$ is a permutation of $\mathcal{M}$. We denote $\mathsf{TP}(\mathcal{T}, n)$ the set of all tweakable permutations with tweak space $\mathcal{T}$ and message space $\{0,1\}^n$. A tweakable block cipher with key space $\mathcal{K}$, tweak space $\mathcal{T}$, and message space $\mathcal{M}$ is a mapping $\mathsf{TBC} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ such that for any key $K \in \mathcal{K}$, $(T, X) \mapsto \mathsf{TBC}(K, T, X)$ is a tweakable permutation in $\mathsf{TP}(\mathcal{T}, n)$.

A secure TBC should be indistinguishable from a tweakable random permutation. As our mode TNT is specified in an idealized manner, our security definition is also given for such cases. For this, we denote $\mathsf{P}(n)$ the set of all n-bit permutations. By default, we always allow $\mathcal{D}$ to make forward and inverse queries to its tweakable permutation oracle (though we do not write this explicitly). With these, for the TBC construction $C^{\pi_1,\ldots,\pi_r}$ built upon r independent secret n-bit permutations, we define the advantage of any distinguisher $\mathcal{D}$ breaking its *strong tweakable pseudorandomness (STPRP)* as

$$\mathbf{Adv}_C^{\mathrm{stprp}}(\mathcal{D}) = \left| \Pr[\pi_1, \ldots, \pi_r \xleftarrow{\$} \mathsf{P}(n) : \mathcal{D}^{C^{\pi_1,\ldots,\pi_r}} = 1] - \Pr[\widetilde{\Pi} \xleftarrow{\$} \mathsf{TP}(\mathcal{T}, n) : \mathcal{D}^{\widetilde{\Pi}} = 1] \right|.$$

And for any non-negative integer q, we define the insecurity of $C^{\pi_1,\ldots,\pi_r}$ as

$$\mathbf{Adv}_C^{\mathrm{stprp}}(q) = \max_{\mathcal{D}} \mathbf{Adv}_C^{\mathrm{stprp}}(\mathcal{D}),$$

where the maximum is taken over all distinguishers $\mathcal{D}$ making exactly q queries to the oracle.

The above definition focuses on the information-theoretic setting. Later in Sect. 5 we will instantiate the multiple secret permutations $\pi_1, \ldots, \pi_r$ with multiple "independent" block ciphers $E_1, \ldots, E_r$ using the *same* secret key K (thus the key space does not increase with the number of permutations). Proving the indistinguishability of such two systems $(\pi_1, \ldots, \pi_r)$ and $((E_1)_K, \ldots, (E_r)_K)$ seems out of reach of current techniques (note that existing works typically instantiated $\pi_1, \ldots, \pi_r$ with the *same* block cipher using r *independent keys* $K_1, \ldots, K_r$, which deviates from us). As such, our mode TNT will be specified only in the idealized manner.

2.3 χ^2 Method

For the proof, we will employ the χ^2 method of Dai *et al.* [24]. We recall this technique here. Below we mainly follow Dai *et al.*'s notations (with some necessary supplementaries borrowed from Chen *et al.* [13]). Concretely, consider two stateless systems $\mathbf{S}_0$ and $\mathbf{S}_1$ (e.g., $\mathbf{S}_0$ and $\mathbf{S}_1$ may be the tweakable random permutation $\widetilde{\Pi}$ and the TNT construction $\mathsf{TNT}^{\pi_1,\pi_2,\pi_3}$ respectively) and any computationally unbounded deterministic distinguisher $\mathcal{D}$ that has query access to either of these systems. The distinguisher's goal is to distinguish the two systems. It is well-known that, the distinguishing advantage $\mathbf{Adv}_{\mathbf{S}_0,\mathbf{S}_1}(\mathcal{D})$ is bounded by the statistical distance $\|\mathsf{p}_{\mathbf{S}_0,\mathcal{D}}(\cdot) - \mathsf{p}_{\mathbf{S}_1,\mathcal{D}}(\cdot)\|$, where $\mathsf{p}_{\mathbf{S}_0,\mathcal{D}}(\cdot)$ and $\mathsf{p}_{\mathbf{S}_1,\mathcal{D}}(\cdot)$ are the respective probability distributions of the answers obtained by $\mathcal{D}$. The χ^2 method concerns with bounding $\|\mathsf{p}_{\mathbf{S}_0,\mathcal{D}}(\cdot) - \mathsf{p}_{\mathbf{S}_1,\mathcal{D}}(\cdot)\|$. To this end, if we denote the maximum amount of queries by q, we can define a transcript $\mathcal{Q} = (\tau_1, \ldots, \tau_q)$ with $\tau_i = (T_i, X_i, Y_i)$, and let $\mathcal{Q}_\ell = (\tau_1, \ldots, \tau_\ell)$ for every $\ell \leq q$. The distinguisher $\mathcal{D}$ can make its queries adaptively, but as it makes them in a deterministic manner, the ℓ-th query input is determined by the first $\ell - 1$ query-responses $\mathcal{Q}_{\ell-1}$.

For system $\mathbf{S}_b$ with $b \in \{0, 1\}$ and fixed tuple $\mathcal{Q}_{\ell-1}$, we denote by $\mathsf{p}_{\mathbf{S}_b,\mathcal{D}}(\mathcal{Q}_{\ell-1})$ the probability that $\mathcal{D}$ interacting with $\mathbf{S}_b$ yields transcript $\mathcal{Q}_{\ell-1}$ for its first $\ell-1$ queries. If $\mathsf{p}_{\mathbf{S}_b,\mathcal{D}}(\mathcal{Q}_{\ell-1}) > 0$, then we denote by $\mathsf{p}_{\mathbf{S}_b,\mathcal{D}}(R_\ell \mid \mathcal{Q}_{\ell-1})$ the conditional probability that $\mathcal{D}$ receives response R_ℓ upon its ℓ-th query, given transcript $\mathcal{Q}_{\ell-1}$ of the first $\ell - 1$ queries (that deterministically fixes the ℓ-th query). Define for any $\ell \in \{1, \ldots, q\}$ and any query-response tuple $\mathcal{Q}_{\ell-1}$:

$$\chi^2(\mathcal{Q}_{\ell-1}) = \sum_{R_\ell} \frac{\left(\mathsf{p}_{\mathbf{S}_1,\mathcal{D}}(R_\ell \mid \mathcal{Q}_{\ell-1}) - \mathsf{p}_{\mathbf{S}_0,\mathcal{D}}(R_\ell \mid \mathcal{Q}_{\ell-1})\right)^2}{\mathsf{p}_{\mathbf{S}_0,\mathcal{D}}(R_\ell \mid \mathcal{Q}_{\ell-1})}, \tag{3}$$

where the sum is taken over all R_ℓ in the support of the distribution $\mathsf{p}_{\mathbf{S}_0,\mathcal{D}}(\cdot \mid \mathcal{Q}_{\ell-1})$. The χ^2 method states the following:

Lemma 1 *(χ^2 method [24, Lemma 3]). Consider a fixed deterministic distinguisher $\mathcal{D}$ and two systems $\mathbf{S}_0, \mathbf{S}_1$. Suppose that for any $\ell \in \{1, \ldots, q\}$ and any query-response tuple $\mathcal{Q}_\ell$, $\mathsf{p}_{\mathbf{S}_0,\mathcal{D}}(\mathcal{Q}_\ell) > 0$ whenever $\mathsf{p}_{\mathbf{S}_1,\mathcal{D}}(\mathcal{Q}_\ell) > 0$. Then:*

$$\|p_{\mathbf{S}_0,\mathcal{D}}(\cdot) - p_{\mathbf{S}_1,\mathcal{D}}(\cdot)\| \leq \left(\frac{1}{2}\sum_{\ell=1}^{q} \mathbf{E}\left[\chi^2(\mathcal{Q}_{\ell-1})\right]\right)^{1/2}, \tag{4}$$

where the expectation is taken over $\mathcal{Q}_{\ell-1}$ of the $\ell - 1$ first answers sampled according to interaction with $\mathbf{S}_1$.

3 The Idealized TNT Mode

In this section, we describe our mode TNT. As discussed in Sect. 2, we only give its idealized description, which is built upon *secret random permutations* rather than efficient block ciphers.

Concretely, TNT is built upon three independent secret random permutations π_1, π_2, and π_3, and is formally defined as

$$\mathsf{TNT}^{\pi_1,\pi_2,\pi_3}(T,X) = \pi_3\big(T \oplus \pi_2\big(T \oplus \pi_1(X)\big)\big). \tag{5}$$

4 Security Proof for TNT Mode

Theorem 1. *When $q \le 2^n/2$, it holds*

$$\mathbf{Adv}^{\mathrm{stprp}}_{\mathsf{TNT}}(q) \le \frac{8q^{1.5}}{2^n}. \tag{6}$$

Proof. In our proof, $\mathbf{S}_0$ denotes the tweakable random permutation $\widetilde{\Pi}$, while $\mathbf{S}_1$ denotes the $\mathsf{TNT}^{\pi_1,\pi_2,\pi_3}$ TBC. The condition stated in Lemma 1, *i.e.*, $\forall \mathcal{Q}_\ell$, $\mathsf{p}_{\mathbf{S}_0,\mathcal{D}}(\mathcal{Q}_\ell) > 0$ whenever $\mathsf{p}_{\mathbf{S}_1,\mathcal{D}}(\mathcal{Q}_\ell) > 0$, is clearly satisfied.

Given $\mathcal{Q}_{\ell-1}$, let T_ℓ be the tweak of the ℓ-th query (note that it is determined by $\mathcal{Q}_{\ell-1}$). It is easy to see that, regardless of the direction of this query, it holds

$$\mathsf{p}_{\widetilde{\Pi},\mathcal{D}}(R_\ell \mid \mathcal{Q}_{\ell-1}) = \frac{1}{2^n - \mu_\ell},$$

where $\mu_\ell \le \ell - 1$ is the frequency of the tweak value T_ℓ in $\mathcal{Q}_{\ell-1}$, *i.e.*,

$$\mu_\ell = \Big|\big\{(X,Y) : (T_\ell, X, Y) \in \mathcal{Q}_{\ell-1}\big\}\Big|.$$

The real world probability $\mathsf{p}_{\mathsf{TNT},\mathcal{D}}(R_\ell \mid \mathcal{Q}_{\ell-1})$ however depends on the concrete state of the ℓ-th query and $\mathcal{Q}_{\ell-1}$, for which we distinguish eight cases as follows.

Case 1: the ℓ-th query is forward $\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell$, and $X_\ell, Y_\ell \in \mathcal{Q}_{\ell-1}$, *i.e.*, $\exists T', X', T^*, Y^* : (T', X', Y_\ell), (T^*, X_\ell, Y^*) \in \mathcal{Q}_{\ell-1}$. We write

$$\begin{aligned}\mathsf{p}_{\mathsf{TNT},\mathcal{D}}(Y_\ell \mid \mathcal{Q}_{\ell-1}) &= \Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] \\ &= \sum_{\mathbf{Inter}} \Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] \cdot \Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}],\end{aligned}$$

where the sum is taken over all the vectors of intermediate values

$$\mathbf{Inter} = \Big((S_1, \ldots, S_{\ell-1}), (U_1, \ldots, U_{\ell-1}), (V_1, \ldots, V_{\ell-1}), (W_1, \ldots, W_{\ell-1})\Big)$$

that are possible to appear given $\mathcal{Q}_{\ell-1}$.

Now, for a certain intermediate vector **Inter**, it can be seen that there are three possibilities, according to which we divide all intermediate vectors into three disjoint classes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$:

- Class $\mathcal{A}$: $\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] = 1$;
  - *i.e.*, the vector **Inter** specifies S_ℓ and W_ℓ as the values corresponding to X_ℓ and Y_ℓ, as well as a input-output relation on π_2 (subsequently abbreviated as π_2-relation) (U_i, V_i) such that $T_\ell \oplus S_\ell = U_i$ and $T_\ell \oplus W_\ell = V_i$.

– Class $\mathcal{B}$: $\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] = \frac{1}{N-\beta(\mathbf{Inter})}$, where $\beta(\mathbf{Inter})$ is the number of distinct U values in $(U_1, \ldots, U_{\ell-1})$;
  - *i.e.*, the two corresponding values $U_\ell = T_\ell \oplus S_\ell$ and $V_\ell = T_\ell \oplus W_\ell$ (as before) are "free", so that $\Pr[\pi_2(U_\ell) = V_\ell \mid \mathbf{Inter}] = \frac{1}{N-\beta(\mathbf{Inter})}$.
– Class $\mathcal{C}$: $\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] = 0$.
  - *i.e.*, the two corresponding values $U_\ell = T_\ell \oplus S_\ell$ and $V_\ell = T_\ell \oplus W_\ell$ (as before) are "contradictory" to **Inter:** there exists a π_2-relation (U_i, V_i) in **Inter** such that
    * $T_\ell \oplus S_\ell = U_i$ yet $T_\ell \oplus W_\ell \neq V_i$; or
    * $T_\ell \oplus S_\ell \neq U_i$ yet $T_\ell \oplus W_\ell = V_i$.

By these, we have

$$\begin{aligned}&\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}]\\ &= \sum_{\mathbf{Inter}\in\mathcal{A}} \Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}] + \sum_{\mathbf{Inter}\in\mathcal{B}} \Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}] \cdot \frac{1}{N-\beta(\mathbf{Inter})}. \qquad (7)\end{aligned}$$

With this, we derive upper and lower bounds as follows.

The Upper Bound: It's easy to see $\beta(\mathbf{Inter}) \le \ell - 1$. By this and Eq. (7), it holds

$$\begin{aligned}&\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}]\\ &\le \Pr[\mathbf{Inter} \in \mathcal{A} \mid \mathcal{Q}_{\ell-1}] + \underbrace{\Pr[\mathbf{Inter} \in \mathcal{B} \mid \mathcal{Q}_{\ell-1}]}_{\le 1} \cdot \frac{1}{2^n - \ell}. \qquad (8)\end{aligned}$$

It remains to bound $\Pr[\mathbf{Inter} \in \mathcal{A} \mid \mathcal{Q}_{\ell-1}]$. For this, note that once the values in **Inter** except for (S_ℓ, W_ℓ) have been fixed, the number of choices for (S_ℓ, W_ℓ) is at least $(2^n - \alpha(\mathcal{Q}_{\ell-1}))(2^n - \gamma(\mathcal{Q}_{\ell-1})) \ge 2^{2n}/4$, where $\alpha(\mathcal{Q}_{\ell-1}) \le q \le 2^n/2$ and $\gamma(\mathcal{Q}_{\ell-1}) \le q \le 2^n/2$ are the number of distinct values in $(S_1, \ldots, S_{\ell-1})$ and $(W_1, \ldots, W_{\ell-1})$. Out of these $\ge 2^{2n}/4$ choices, the number of choices that ensure the desired property $\mathsf{TNT}(T_\ell, X_\ell) = Y_\ell$ is at most $\ell - 1$, which results from the following selection process: we first pick a pair of input-output (U_i, V_i) with $i \le \ell - 1$, and then set $S_\ell = T_\ell \oplus U_i$ and $W_\ell = T_\ell \oplus V_i$. Therefore, $\Pr[\mathbf{Inter} \in \mathcal{A} \mid \mathcal{Q}_{\ell-1}] \le \frac{4\ell}{2^{2n}}$, and thus the upper bound in this case is

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] \le \frac{4\ell}{2^{2n}} + \frac{1}{2^n - \ell}. \qquad (9)$$

The Lower Bound: It can be seen $\beta(\mathbf{Inter}) \ge \mu_\ell$, since every previous query under the tweak T_ℓ gives rise to a unique pair (U, V) in $((U_1, V_1), \ldots, (U_{\ell-1}, V_{\ell-1}))$. Therefore, still from Eq. (7), we have

$$\begin{aligned}\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] &\ge \sum_{\mathbf{Inter}\in\mathcal{B}} \Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}] \cdot \frac{1}{2^n - \mu_\ell}\\ &= \Pr[\mathbf{Inter} \in \mathcal{B} \mid \mathcal{Q}_{\ell-1}] \cdot \frac{1}{2^n - \mu_\ell}.\end{aligned}$$

As before, out of the $(2^n - \alpha(\mathcal{Q}_{\ell-1}))(2^n - \gamma(\mathcal{Q}_{\ell-1}))$ choices of (S_ℓ, W_ℓ), the number of choices that ensure the desired property $T_\ell \oplus S_\ell \notin \{U_1, \ldots, U_{\ell-1}\}$ and $T_\ell \oplus W_\ell \notin \{V_1, \ldots, V_{\ell-1}\}$ is at least $(2^n - \ell)^2$. This means $\Pr[\mathbf{Inter} \in \mathcal{B} \mid \mathcal{Q}_{\ell-1}] \geq \frac{2^n-\ell}{2^n-\alpha(\mathcal{Q}_{\ell-1})} \cdot \frac{2^n-\ell}{2^n-\gamma(\mathcal{Q}_{\ell-1})} \geq (1 - \frac{\ell}{2^n})^2 \geq 1 - \frac{2\ell}{2^n}$, and thus

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] \geq \left(1 - \frac{2\ell}{2^n}\right) \cdot \frac{1}{2^n - \mu_\ell}. \tag{10}$$

Summary. In all, in the first case, we have

$$\begin{aligned} &\left| \Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] - \frac{1}{2^n - \mu_\ell} \right| \\ \leq &\max\left\{ \frac{4\ell}{2^{2n}} + \frac{\ell - \mu_\ell}{(2^n - \mu_\ell)(2^n - \ell)}, \frac{2\ell}{2^n} \cdot \frac{1}{2^n - \mu_\ell} \right\} \leq \frac{8\ell}{2^{2n}}. \end{aligned} \tag{11}$$

Case 2: the ℓ-th query is forward $\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell$, and $X_\ell \in \mathcal{Q}_{\ell-1}$, $Y_\ell \notin \mathcal{Q}_{\ell-1}$, *i.e.*, $\exists T', Y' : (T', X_\ell, Y') \in \mathcal{Q}_{\ell-1}$, yet $\forall T, X : (T, X, Y_\ell) \notin \mathcal{Q}_{\ell-1}$. Now, for a certain intermediate vector **Inter**, there are three possibilities, according to which we divide all intermediate vectors into three disjoint classes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$:

- Class $\mathcal{A}$: there does not exist (U_i, V_i) such that $U_i = T_\ell \oplus S_\ell$, where S_ℓ is specified by **Inter** and corresponds to X_ℓ.
- Class $\mathcal{B}$: there exists (U_i, V_i) such that $U_i = T_\ell \oplus S_\ell$, and $\Pr[\pi_3(T_\ell \oplus V_i) = Y_\ell] = \frac{1}{2^n - \gamma(\mathcal{Q}_{\ell-1})}$, where $\gamma(\mathcal{Q}_{\ell-1})$ is the number of distinct values in $(Y_1, \ldots, Y_{\ell-1})$.
- Class $\mathcal{C}$: there exists (U_i, V_i) such that $U_i = T_\ell \oplus S_\ell$, and $\Pr[\pi_3(T_\ell \oplus V_i) = Y_\ell] = 0$.

By these, we have

$$\begin{aligned} &\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] \\ = &\sum_{\mathbf{Inter} \in \mathcal{A}} \Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}] \cdot \Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] \\ &+ \sum_{\mathbf{Inter} \in \mathcal{B}} \Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}] \cdot \frac{1}{2^n - \gamma(\mathcal{Q}_{\ell-1})}. \end{aligned} \tag{12}$$

The Upper Bound: For this we need to consider $\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}]$ for any $\mathbf{Inter} \in \mathcal{A}$. Let $U_\ell = T_\ell \oplus S_\ell$. Then it can be seen

$$\begin{aligned} &\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] \\ = &\sum_{V_\ell \in \{0,1\}^n} \Pr[\pi_2(U_\ell) = V_\ell \mid \mathbf{Inter}] \cdot \Pr[\pi_3(T_\ell \oplus V_\ell) = Y_\ell \mid \mathbf{Inter}] \\ \leq &\underbrace{\sum_{V_\ell \in \{0,1\}^n} \Pr[\pi_2(U_\ell) = V_\ell \mid \mathbf{Inter}]}_{\leq 1} \cdot \frac{1}{2^n - \gamma(\mathcal{Q}_{\ell-1})}. \end{aligned} \tag{13}$$

By this, the upper bound in this case is

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] \le \sum_{\mathbf{Inter} \in \mathcal{A} \cup \mathcal{B}} \Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}] \cdot \frac{1}{2^n - \gamma(\mathcal{Q}_{\ell-1})} \le \frac{1}{2^n - \gamma(\mathcal{Q}_{\ell-1})} \le \frac{1}{2^n - \ell}.$$

The Lower Bound: Still by Eq. (13), for any $\mathbf{Inter} \in \mathcal{A}$ we have

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] \ge \sum_{W_\ell \in \mathcal{GW}} \Pr[\pi_2(U_\ell) = T_\ell \oplus W_\ell \mid \mathbf{Inter}] \cdot \Pr[\pi_3(W_\ell) = Y_\ell \mid \mathbf{Inter}],$$

where $\mathcal{GW}$ ("good W set") is the set of W_ℓ such that:

- $W_\ell \notin \{W_1, \ldots, W_{\ell-1}\}$, and
- $T_\ell \oplus W_\ell \notin \{V_1, \ldots, V_{\ell-1}\}$.

It can seen that $|\mathcal{GW}| \ge 2^n - \ell - \ell + \mu_\ell = 2^n - 2\ell + \mu_\ell$: the reason is, for any $(T_i, X_i, Y_i) \in \mathcal{Q}_{\ell-1}$ with $T_i = T_\ell$, $W_\ell \ne W_i \Leftrightarrow T_\ell \oplus W_\ell \ne V_i$. On the other hand, $\Pr[\pi_3(W_\ell) = Y_\ell \mid \mathbf{Inter}] = \frac{1}{2^n - \gamma(\mathcal{Q}_{\ell-1})} \ge \frac{1}{2^n - \mu_\ell}$, and $\Pr[\pi_2(U_\ell) = T_\ell \oplus W_\ell \mid \mathbf{Inter}] = \frac{1}{2^n - \beta(\mathbf{Inter})} \ge \frac{1}{2^n - \mu_\ell}$. Therefore, for any $\mathbf{Inter} \in \mathcal{A}$ we have

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] \ge \frac{2^n - 2\ell + \mu_\ell}{(2^n - \mu_\ell)^2}.$$

By these and Eq. (12), we have

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] \ge \sum_{\mathbf{Inter} \in \mathcal{A} \cup \mathcal{B}} \Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}] \cdot \frac{2^n - 2\ell + \mu_\ell}{(2^n - \mu_\ell)^2} = \Big(1 - \Pr[\mathbf{Inter} \in \mathcal{C} \mid \mathcal{Q}_{\ell-1}]\Big) \cdot \frac{2^n - 2\ell + \mu_\ell}{(2^n - \mu_\ell)^2}.$$

To bound $\Pr[\mathbf{Inter} \in \mathcal{C} \mid \mathcal{Q}_{\ell-1}]$, note that if $\mathbf{Inter} \in \mathcal{C}$, then there exists $Y_i \in \{Y_1, \ldots, Y_{\ell-1}\}$ such that $\Pr[\mathsf{TNT}(T_\ell, X_\ell) = Y_i \mid \mathcal{Q}_{\ell-1}] = 1$. For each such Y_i the probability is at most $\frac{4\ell}{2^{2n}}$ as analyzed in Case 1. Since there are at most $\ell - 1 \le \ell$ choices for this Y_i, we obtain

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] \ge \Big(1 - \frac{4\ell^2}{2^{2n}}\Big) \cdot \frac{2^n - 2\ell + \mu_\ell}{(2^n - \mu_\ell)^2}$$

as the lower bound. Further note that

$$\begin{aligned} &\frac{1}{2^n - \mu_\ell} - \Big(1 - \frac{4\ell^2}{2^{2n}}\Big) \cdot \frac{2^n - 2\ell + \mu_\ell}{(2^n - \mu_\ell)^2} \\ \le &\frac{1}{2^n - \mu_\ell} - \frac{2^n - 2\ell + \mu_\ell}{(2^n - \mu_\ell)^2} + \frac{4\ell^2}{2^{2n}} \cdot \frac{2^n - 2\ell + \mu_\ell}{(2^n - \mu_\ell)^2} \\ \le &\frac{2(\ell - \mu_\ell)}{(2^n - \mu_\ell)^2} + \frac{8\ell^2}{2^{3n}} \le \frac{8\ell}{2^{2n}} + \frac{8\ell}{2^{2n}} = \frac{16\ell}{2^{2n}}. \end{aligned}$$

Summary. In all, in the second case, we have

$$\begin{aligned}&\left|\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] - \frac{1}{2^n - \mu_\ell}\right| \\ \le &\max\left\{\frac{1}{2^n-\ell} - \frac{1}{2^n-\mu_\ell}, \frac{16\ell}{2^{2n}}\right\} \le \frac{16\ell}{2^{2n}}. \end{aligned} \tag{14}$$

Case 3: the ℓ-th query is forward $\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell$, and $X_\ell \notin \mathcal{Q}_{\ell-1}$, $Y_\ell \in \mathcal{Q}_{\ell-1}$. The analysis is similar to Case 2 by symmetry, resulting in the same bound

$$\left|\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] - \frac{1}{2^n - \mu_\ell}\right| \le \frac{16\ell}{2^{2n}}. \tag{15}$$

Case 4: the ℓ-th query is forward $\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell$, and $X_\ell, Y_\ell \notin \mathcal{Q}_{\ell-1}$. The analyses for this case heavily resemble Case 2. First, the same upper bound

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] \le \frac{1}{2^n - \gamma(\mathcal{Q}_{\ell-1})} \le \frac{1}{2^n - \ell}$$

can be established. Second, for any **Inter** such that $\Pr[\mathbf{Inter} \mid \mathcal{Q}_{\ell-1}] > 0$, we have

$$\begin{aligned}&\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] \\ \ge &\sum_{S_\ell \in \mathcal{GS}, W_\ell \in \mathcal{GW}} \Pr[\pi_1(X_\ell) = S_\ell \mid \mathbf{Inter}] \cdot \Pr[\pi_2(T_\ell \oplus S_\ell) = T_\ell \oplus W_\ell \mid \mathbf{Inter}] \\ &\qquad \cdot \Pr[\pi_3(W_\ell) = Y_\ell \mid \mathbf{Inter}],\end{aligned}$$

where $\mathcal{GS}$ is the set of S_ℓ such that:

- $S_\ell \notin \{S_1, \ldots, S_{\ell-1}\}$, and
- $T_\ell \oplus S_\ell \notin \{U_1, \ldots, U_{\ell-1}\}$,

and $\mathcal{GW}$ is the set of W_ℓ such that:

- $W_\ell \notin \{W_1, \ldots, W_{\ell-1}\}$, and
- $T_\ell \oplus W_\ell \notin \{V_1, \ldots, V_{\ell-1}\}$.

It is easy to see $|\mathcal{GS}|, |\mathcal{GW}| \ge 2^n - 2\ell + \mu_\ell$, $\Pr[\pi_1(X_\ell) = S_\ell \mid \mathbf{Inter}] = \frac{1}{2^n - \alpha(\mathcal{Q}_{\ell-1})} \ge \frac{1}{2^n - \mu_\ell}$, $\Pr[\pi_3(W_\ell) = Y_\ell \mid \mathbf{Inter}] = \frac{1}{2^n - \gamma(\mathcal{Q}_{\ell-1})} \ge \frac{1}{2^n - \mu_\ell}$, and $\Pr[\pi_2(U_\ell) = T_\ell \oplus W_\ell \mid \mathbf{Inter}] = \frac{1}{2^n - \beta(\mathbf{Inter})} \ge \frac{1}{2^n - \mu_\ell}$. Therefore, we have

$$\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathbf{Inter}] \ge \frac{(2^n - 2\ell + \mu_\ell)^2}{(2^n - \mu_\ell)^3},$$

for which

$$\frac{1}{2^n - \mu_\ell} - \frac{(2^n - 2\ell + \mu_\ell)^2}{(2^n - \mu_\ell)^3} \le \frac{4(\ell - \mu_\ell)(2^n - \ell)}{(2^n - \mu_\ell)^3} \le \frac{16\ell}{2^{2n}}.$$

Therefore,

$$\left|\Pr[\mathsf{TNT}(T_\ell, X_\ell) \to Y_\ell \mid \mathcal{Q}_{\ell-1}] - \frac{1}{2^n - \mu_\ell}\right| \leq \max\left\{\frac{1}{2^n - \ell} - \frac{1}{2^n - \mu_\ell}, \frac{16\ell}{2^{2n}}\right\} \leq \frac{16\ell}{2^{2n}}. \tag{16}$$

To conclude, when the ℓ-th query is forward, from Eqs. (11), (14), (15), and (16) we have

$$\left(\mathsf{p}_{\mathsf{TNT},\mathcal{D}}(Y_\ell \mid \mathcal{Q}_{\ell-1}) - \frac{1}{2^n - \mu_\ell}\right)^2 \leq \left(\frac{16\ell}{2^{2n}}\right)^2 \leq \frac{256\ell^2}{2^{4n}}.$$

The remaining Cases 5, 6, 7, and 8 concern with the case where the ℓ-th query is backward, and the analyses are similar to Cases 1, 2, 3, and 4 by symmetry, resulting in the same bound

$$\left(\mathsf{p}_{\mathsf{TNT},\mathcal{D}}(X_\ell \mid \mathcal{Q}_{\ell-1}) - \frac{1}{2^n - \mu_\ell}\right)^2 \leq \left(\frac{16\ell}{2^{2n}}\right)^2 \leq \frac{256\ell^2}{2^{4n}}.$$

Consequently,

$$\chi^2(\mathcal{Q}_{\ell-1}) \leq \sum_{R_\ell} \frac{256\ell^2/2^{4n}}{1/(2^n - \mu_\ell)} \leq 2^n \cdot 2^n \cdot \frac{256\ell^2}{2^{4n}} \leq \frac{256\ell^2}{2^{2n}},$$

and

$$\frac{1}{2}\sum_{\ell=1}^{q} \mathbf{E}\left[\chi^2(\mathcal{Q}_{\ell-1})\right] \leq \frac{1}{2}\sum_{\ell=1}^{q} \frac{256\ell^2}{2^{2n}} \leq \frac{1}{2} \cdot \frac{128q^3}{2^{2n}} = \frac{64q^3}{2^{2n}},$$

which implies Eq. (6) by Lemma 1. □

5 Concrete Proposals

In this section, we propose our instantiation of the TNT construction based on AES, which allows fast software implementations when AES-NI are available. We call the instantiation TNT-AES. To also enjoy the long-standing security of AES, we try to make minimum possible modifications over AES. Following these considerations, we only extend the number of rounds without any modification to its round function or key schedule, and pick the respective numbers of rounds for the three permutations π_1, π_2, and π_3 so that the design is secure against all relevant attacks. More explicitly, when the tweak $T = 0$, TNT-AES simply becomes AES with more rounds, which clearly leaves higher security margins over AES. Besides, we let the last round be complete instead of missing the MixColumns operation. In the remainder of the section, we give the description of TNT-AES, followed by a comprehensive cryptanalysis, and a comparison of software and hardware performances against other existing TBCs with similar security levels.

5.1 Instantiation Based on AES

The Advanced Encryption Standard (AES) [23] is an iterated block cipher with block size 128 bits and secret key sizes 128, 192, and 256 bits. The internal state of AES, as well as the round keys, can be represented as a 4×4 matrix whose elements are byte value (8 bits). The round function consists of four basic transformations in the following order (see Fig. 2):

- SubBytes (SB) is a nonlinear substitution that applies the same S-box to each byte of the internal state.
- ShiftRows (SR) is a cyclic rotation of the i-th row by i bytes to the left, for $i = 0, 1, 2, 3$.
- MixColumns (MC) is a multiplication of each column with a Maximum Distance Separable (MDS) matrix over GF(2^8).
- AddRoundKey (AK) is an exclusive-or with the round key.

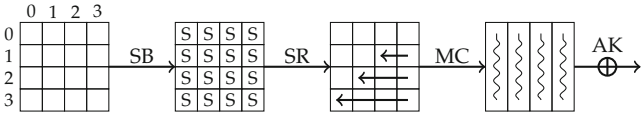

Fig. 2. AES round function

At the very beginning of the encryption, an additional pre-whitening key addition is performed, and the last round is different from the normal rounds by omitting the MixColumns operation. AES-128, AES-192, and AES-256 share the same round function with different numbers of rounds: 10, 12, and 14, respectively.

The key schedule of AES transforms the master key into subkeys that are used in each of the rounds. Here, we describe the key schedule of AES-128. The 128-bit master key is divided into four 32-bit words $(W[0], W[1], W[2], W[3])$, then $W[i]$ for $i \geqslant 4$ is computed as

$$W[i] = \begin{cases} W[i-4] \oplus \mathsf{SB}(\texttt{RotByte}(W[i-1])) \oplus Rcon[i/4] & i \equiv 0 \bmod 4, \\ W[i-4] \oplus W[i-1] & \text{otherwise.} \end{cases}$$

The i-th round key is the concatenation of 4 words $W[4i] \parallel W[4i+1] \parallel W[4i+2] \parallel W[4i+3]$. RotByte is a cyclic shift by one byte to the left, and $Rcon$ are the round constants defined as

$$Rcon[i] = \begin{cases} 1 & i = 0, \\ 2 \cdot Rcon[i-1] & \text{otherwise,} \end{cases}$$

where '$\cdot$' denotes multiplication in GF(2^8) with irreducible polynomial $x^8 + x^4 + x^3 + x + 1$.

Although AES-128 consists of 10 rounds, it can be naturally extended to more rounds, each composed of all 4 transformations (AddRoundKey ∘ MixColumns ∘ ShiftRows ∘ SubBytes), and the pre-whitening key addition to the first round is kept as it is. Then, TNT-AES[n_1, n_2, n_3] is defined to be the extension of AES to $(n_1+n_2+n_3)$ rounds, *i.e.*, π_1, π_2, π_3 are of n_1, n_2, n_3 full AES rounds respectively, and the 128-bit tweak is XOR-ed into the internal state at the output of π_1 and π_2. It is natural to set $n_1 = n_3$ due to the symmetry of the design. Concretely, we define TNT-AES[6, 6, 6], and will use TNT-AES to denote this choice for the sake of simplicity. We will justify the round numbers in the security analysis below.

5.2 Preliminary Cryptanalysis

In this subsection, we give our preliminary cryptanalysis against TNT-AES. As TNT-AES consists of 18 rounds in total, which is 8 more rounds than AES-128, we expect higher security margins of TNT-AES when the tweak is treated as a given constant. Hence, we focus on only the cases where the tweaks help the attack from cryptanalysts' point of view, *i.e.*, it is assumed the tweak is under the attacker's full control (*open* tweak), and possibly extends the existing attacks against round-reduced AES. Under such a setting, we verify the most efficient attacks in terms of number of attacked rounds, against TNT-AES and claim the absence of key-recovery attack against the full TNT-AES in the single-key setting. While we do not claim security under the related-key setting for TNT-AES due to lack of security proof for TNT in such setting, our preliminary cryptanalysis below shows that there is no key-recovery attack either.

Following the proven security bound of TNT, TNT-AES offers $2n/3$-bit security, *i.e.*, there exists no key-recovery attack, given that the data (the combination of tweak and plaintext with no restriction on individual input) and time complexities are bounded by $2^{2 \cdot 128/3} \simeq 2^{85}$. Due to the fact that there is no attack against TNT matching the $2^{2n/3}$ bound, all our security analysis against TNT-AES are following the $2^n = 2^{128}$ bound for both data and time. This allows TNT-AES offering higher security strength should a better than $2n/3$-bit bound be proven for TNT. In summary, we claim that there is no shortcut attack on TNT-AES better than the generic attacks against the corresponding TNT mode.

In what follows, explicit security margins are given under each attack method whenever possible. Before moving to the individual attack methods, an overview of the impact of the tweak to the security at model level is given as follows. As mentioned above, the security margin will be higher for TNT-AES when tweak is a given constant, and we call such a tweak *inactive*. When the tweak is *active*, it may be used to cancel differences in differential attack, or to be used as the source of input structure in integral attacks. Under the single-key setting, the activeness of the round functions will be consistent within each of the three permutations π_1, π_2, and π_3. This allows us using 0/1 to denote the activeness of the permutations with 1 for active (0 for inactive), and a simple exhaustive search shows there are activity patterns $\{(0,1,0), (0,1,1), (1,1,0), (1,0,1), (1,1,1)\}$ for differential attacks, and $\{(1,1,1), (1,1,0), (0,1,1)\}$ for integral attacks and alike.

Differential and Linear Attacks. In the single-key setting, we will employ the known results of 4-round AES to justify the security of TNT-AES. It is well-known that there are at least 25 active S-boxes in 4 rounds of AES, which makes sure that there exists no 4-round differential characteristic (resp. linear approximation) with differential probability (resp. linear correlation) higher than $2^{-6\times25}$ (resp. $2^{-3\times25}$) [22]. For the maximum expected differential and linear probability (MEDP and MELP), known results can be obtained following the work of Keliher and Sui [47], which suggests that the upper bound on the MEDP (and MELP) of 4-round AES is about $(53/2^{34})^4 \approx 2^{-110}$. For TNT-AES in the single-key setting where the difference can be injected on the plaintext or the tweak, there is at least one active permutation among π_1, π_2, π_3 since their activity patterns fall in $\{(0,1,0),(0,1,1),(1,1,0),(1,0,1),(1,1,1)\}$. As long as π_2 is active, there must be more than 25 active S-boxes. In the case of $(1,0,1)$, it happens only when the first addition of the tweak cancels out the differences introduced from plaintext through π_1, and the same difference is then re-introduced through the second addition of tweak through π_3. Due to the fact that the same tweak is added and the difference in tweak is the same as well, π_1 and π_3 can be concatenated together with respective to differences. Note that $\pi_1 + \pi_3$ is of 12 rounds in total, out of which any 4 consecutive rounds will ensure 25 active S-boxes. We also note the security analysis of TNT under such a setting is very similar to that of AES-PRF [60] except one has the control over the extra input tweak in TNT added to the unknown internal state.

In the related-key setting, we only considered differential cryptanalysis, as there is no cancellation of active S-boxes between subkeys and the state in linear approximations. In [73], it is shown that in the related-key setting, there are at least 21 active S-boxes in consecutive 6 rounds of AES-128, and the optimal 6-round differential has probability 2^{-131}. Therefore, no useful related-key differential characteristic covering more than π_2 can be found no matter whether there is a difference in the tweak or not.

Impossible-Differential Attacks. In [71], it is proven that there does not exist any truncated impossible-differential of AES which covers more than 5 rounds. Furthermore, the best impossible-differential attack, in terms of number of attacked rounds, is 7 rounds against AES-128 [55]. Following a similar discussion for differential attacks, when π_2 is active, impossible-differential attack does not apply naturally since π_2 is of 6 rounds, more than what impossible-differential distinguisher can cover. For the case of activity pattern $(1,0,1)$, there are 12 rounds in total for $\pi_1 + \pi_3$, more than the best attack against AES-128 can cover.

The Demirci-Selçuk Meet-in-the-Middle Attack. The Demirci-Selçuk meet-in-the-middle attack led to the best cryptanalytic result on 7 rounds of AES-128 in the single-key setting, where data/time/memory complexities are below 2^{100} [25]. The distinguisher covers 4 rounds, following a differential characteristic. Note, the distinguisher here tries to limit the number of possibilities

for the actual values related to the differential characteristic, and it is not clear how the addition of the tweak helps reduce that. Actually, it is not even clear the addition of round key can help reduce the counts either. Hence, round keys are treated as independent fixed constants in such attacks. Thus, we can treat the tweak in the same way. Therefore, the Demirci-Selçuk meet-in-the-middle attack would work in the same way on TNT-AES as on AES, and 7 rounds of TNT-AES can be attacked.

Yoyo Tricks. In [68], Rønjom *et al.* presented several key-independent yoyo-distinguishers on 3- to 5-round AES, which require up to $2^{25.8}$ data and $2^{24.8}$ XOR computations. A key-independent impossible-differential yoyo-distinguisher on 6-round AES requiring an amount of $2^{122.83}$ data was also proposed. Besides, a key-recovery attack on 5-round AES requiring practical complexities was devised based on the 4-round yoyo-distinguisher. In these attacks, the attacker queries pair of plaintexts to the encryption and uses swap operation on the obtained pair of ciphertexts to generate new queries to the decryption, and observes difference in the obtained pair of plaintexts, then she may continually construct new pairs of plaintexts by swapping words in the obtained pairs and iterate the same procedure enough times. It can be seen that, instead of collecting all chosen plaintexts/ciphertexts (CPs/CCs) at once, these attacks use adaptively-chosen-plaintexts/-ciphertexts (ACPs/ACCs). In TNT-AES, tweaks are always inserted as input to the encryption/decryption, and will never be output. So, for activity pattern $(0, 1, 1)$ (resp. $(1, 1, 0)$ for decryption), the attacker cannot play the yoyo game by adaptively choosing and observing the differences of tweak pairs and ciphertext (resp. plaintext) pairs. Accordingly, we claim that these yoyo-distinguishers and yoyo-distinguisher-based key-recovery attacks cannot be directly applied in their current form to TNT-AES.

Subspace Trail Attacks. Subspace trail cryptanalysis [32] can be seen as a generalization of invariant subspace cryptanalysis [52], whereas it can be launched independently on specific choices of round constants or subkeys. By analyzing subspace trails, Grassi *et al.* re-interpreted the 3-round truncated differential and integral, the 4-round impossible-differential and integral distinguishers on AES [33]. Besides, new distinguishers on round-reduced AES are found using subspace trail cryptanalysis, including the 5-round impossible-differential distinguisher [33], the 5-round multiple-of-8 distinguisher [34], the 4-round mixture-differential [31], and the 5-round (probabilistic, threshold, and impossible) mixture-differential distinguishers [30]. Exploiting the 4-round mixture-differential distinguisher, a record for key-recovery attack on 5-round AES-128 in single-key model is set [4]. In [6], Bardeh and Rønjom proposed the exchange attacks. Like in yoyo and mixture-differential attacks, exchange attacks also involve swap (exchange) operations on the pairs of chosen data. On 6-round AES, the exchange distinguishers requires $2^{88.2}$ CPs and $2^{88.2}$ encryptions. In the attacks, new plaintext pairs are obtained by exchanging certain active diagonal of other pairs that are different in diagonals, and an invariant property on

the number of active columns of the differences of ciphertext pairs under such exchange operation are considered.

Using subspace trail cryptanalysis and comparing with distinguishers on round-reduced AES, we analyze distinguishers and corresponding attacks on round-reduced TNT-AES. The activity patterns of the three permutations that we considered are $(0,1,0)$, $(1,0,1)$, $(0,1,1)$, $(1,1,0)$, and $(1,1,1)$. The activity pattern $(0,1,0)$ requires that all differences are comes from tweaks and canceled by the same tweaks through n_2 (*i.e.*, 6) AES-rounds, which has no shortcut method up to now. Considering that all subspace-trail-based distinguishers on round-reduced AES are no more than n_2 (*i.e.*, 6) AES-rounds, it seems hard to construct an exploitable subspace trail under activity patterns $(0,1,1), (1,1,0)$, $(1,1,1)$, which indicate more than a chunk of active 6-round AES. The activity pattern $(1,0,1)$ implies that the coset of subspace related to the internal states at the end of π_1 (resulted from a set of plaintexts) equals a coset of the same subspace formed by the chosen tweaks (and the differences between tweak pairs should cancel the differences caused by the plaintext pairs), and thus the coset of subspace formed by the chosen tweaks will cause the internal states at the beginning of π_3 forming a coset of the same subspace. A subspace trail on internal states can be seen as bypassing π_2 via choosing a coset of subspace of the tweak. Thus, devising an attack using a subspace trail under activity pattern $(1,0,1)$ requires that one can devise a subspace trail attack on the concatenated permutation $\pi_3 \circ \pi_1$ that is of $(n_1 + n_3)$ AES-rounds, which is unknown when $(n_1 + n_3) > 6$. In Appendix A, we discuss in detail the subspace-trail-based distinguishers and key-recovery attacks on round-reduced TNT-AES.

Cube Attack, Dynamic Cube Attack. AES is immune to cube attacks [27] or dynamic cube attacks [28] due to the high algebraic degree of the AES S-box. Specifically, the algebraic degree is 7 for one round of AES and increases to 32 ($<7^2$) and 128 ($<32 \times 7$) for two and three rounds. Therefore, AES, which has 10 rounds, is believed to be resistant to such types of attacks. So is TNT-AES since it has more rounds than AES.

Integral Attacks and Division Property. The integral attacks utilize an integral distinguisher for 3 rounds (or 4 rounds without MixColumns for the last round), with a starting point of *ALL* values for a diagonal and a *BALANCED* output, *i.e.*, the sum of each individual byte is 0. The best attack setting will be to utilize the degrees of freedom from the tweak to achieve the distinguisher starting from the input of π_2 in forward direction with activity pattern $(0,1,1)$ (or output of π_2 in backward direction with activity pattern $(1,1,0)$). The attack will start with a fixed plaintext, and take ALL values of a diagonal from tweak. Thus, the target is $\pi_2 + \pi_3$ only with a secret input to π_2. In the key-recovery phase, the attacker is able to append one round only, so this attack will work for at most $n_1 + 5$ out of $(n_1 + n_2 + n_3)$ rounds, *i.e.*, $6 + 5$ out of 18 for TNT-AES.

The division property due to Todo *et al.* [74,75] can be viewed as an extension of integral distinguisher, which has been successfully applied to many block ciphers. However, there is no reported results on AES better than integral attack so far.

Slide Attacks. The slide attack was first described by Biryukov and Wagner [10,11] in 1999 to attack round-reduced DES. The core idea is to make use of the similarity of the round functions and that of key schedule. Thus, the difference of encryption process in its original form and one (or few) rounds shifted is within control, *e.g.*, with high probability. The addition of tweak will allow canceling the difference in at most one round, while TNT-AES has 8 more rounds than AES-128. Hence, we expect higher security margin here. Furthermore, there is no reported slide attack against full AES-128 so far.

(Related-Subkey) Boomerang Attacks. Boomerang attacks [76] construct long distinguishers by connecting two short differential characteristics. Recently, a new tool named Boomerang Connectivity Table [16] was proposed to formulate the dependency that the two differential characteristics contain and offer guidance towards better boomerang distinguishers. We utilize the framework of the boomerang connectivity table when mounting boomerang attacks on TNT-AES. First, we consider the single-key setting where the difference can be introduced on the plaintext or the tweak. When the difference is introduced only on the tweak, as shown in Fig. 3 in Appendix B, high-probability boomerang distinguishers can be constructed on $n_1 + n_2 + n_3$ rounds, where n_1, n_3 can be any number and $n_2 < 6$. When $n_2 \geq 6$, such high-probability distinguishers do not exist. Note that these distinguishers with zero plaintext and ciphertext difference are not useful in key-recovery attacks. When the difference is also introduced to the plaintext or ciphertext, by making π_2 inactive through the tweak difference, the cipher can be seen as $\pi_1 \circ \pi_3$ with respective to differences and boomerang attacks of $n_2 + r$ rounds can be mounted, where r is the number of rounds that boomerang attacks of AES-128 can cover and is 5. That is, only 11 rounds can be attacked. Next, we consider the related-subkey setting where the key difference can be injected on a round key. The related-subkey setting is more powerful and usually allows longer boomerang distinguishers than the related-key setting where the difference is injected on the master key. In the related-subkey setting, there exists a 6-round boomerang distinguisher of AES-128 with probability $2^{-109.42}$ [70]. This distinguisher can be naturally extended to the 7 middle rounds of TNT-AES with the same probability under the condition that the tweak difference cancels the input difference or the output difference of the 6-round boomerang distinguisher. When we add one more round to the bottom or to the top of the 7-round distinguisher, the numbers of active S-boxes will increase at least by one, leading to a negligible probability. Therefore, there seem no boomerang distinguishers of TNT-AES in the related-subkey setting that cover more than 8 rounds.

5.3 Performance

Software Performance. We estimate the software performance of TNT-AES on the basis of the best results of AES software provided by Park *et al.* [65]. In what follows, we consider both "Plaintext" and "Tweak" as data since when used in some authenticated encryption schemes, both of them are used to process data such as associated data. Hence, the software performance is then calculated as the total number of CPU cycles divided by the total byte length of plaintext and tweak of the TBCs. To obtain a fair comparison, we estimate the same for other existing TBCs as well (omitting their additional cost for updating tweaks), using the following formula:

$$\text{original speed} \times \frac{\text{block size}}{\text{block size} + \text{tweak size}}. \tag{17}$$

For TNT-AES, the number of rounds are different from AES. To evaluate the performance, we multiply a factor to the speed of AES. Accordingly, the formula we used to calculate the software speed of TNT-AES is (where, AES means AES-128):

$$\text{speed of AES} \times \frac{\text{block size}}{\text{block size} + \text{tweak size}} \times \frac{\text{TNT-AES round number}}{\text{AES round number}}. \tag{18}$$

We note that the optimization technique proposed in [65] is for the CTR mode of AES, which extends the counter-mode caching [9,78]. It caches and reuses intermediate results up to AES round 1 (R1) or up to AES round 2 (R2). For TNT-AES, tweaks are added until round $(n_1 + 1)$. Thus, this optimization technique is applicable. Whereas, for other TBCs in which tweaks are added before the first round, this technique may not be applicable.

Table 2 presents the estimated results on software performance of TNT-AES, together with the results of other TBCs under the similar setting (considering both "Plaintext" and "Tweak" as data).

Rekeying and Retweaking. To see the scenario that profits considerably by using a tweakable block cipher processing tweak efficiently, we performed a performance comparison between retweaking in TNT-AES and rekeying in AES-128. Table 3 reports the timing results. Because in the AES-NI set, the reciprocal throughput of the `AESKEYGENASSIST` instruction that assists the key-schedule is higher than that of the instruction `AESENC` that executes one round of encryption, in Table 3, it can be seen that the process of rekeying in AES becomes slower. Whereas, the process of retweaking in TNT-AES benefits a lot from the fast AES-NI instruction for encryption.[1]

[1] Please refer to https://github.com/TNT-AES/Rekeying_vs_Retweaking for a very simple implementation of the TNT-AES encryption and this performance comparison.

Table 2. A table of comparison with other TBCs on software (all TBCs are with 128-bit block, 128-bit master key). The platform is Intel Haswell CPU i7-4770, which is the commonly used CPU in references [8,40,42,65].

Cipher	Data type	Tech.	Speed in cycles per byte, given messages in bytes										Ref.
			128	256	512	1024	2048	4096	8192	20480	40960	65536	
AES	Plaintext	Table-based				8.38		8.34		8.37	8.37		[65]
	Plaintext	Bitsliced				4.70		4.43		4.40	4.40		[65]
	Plaintext	AES-NI 1 × 1 R1				1.03		1.02		1.07	1.07		[65]
	Plaintext	AES-NI 1 × 1 R2				0.93		0.92		1.04	1.04		[65]
	Plaintext	AES-NI 1 x 4 R1				0.63		0.62		0.62	0.62		[65]
	Plaintext	AES-NI 1 x 4 R2				0.59		0.58		0.58	0.58		[65]
TNT- AES	Plaintext + Tweak	Table-based				7.54		7.51		7.53	7.53		⋆ [65]
	Plaintext + Tweak	Bitsliced				4.23		3.99		3.96	3.96		⋆ [65]
	Plaintext + Tweak	AES-NI 1 × 1 R1				0.92		0.92		0.97	0.96		⋆ [65]
	Plaintext + Tweak	AES-NI 1 × 1 R2				0.83		0.83		0.94	0.94		⋆ [65]
	Plaintext + Tweak	AES-NI 1 × 4 R1				0.57		0.56		0.56	0.56		⋆ [65]
	Plaintext + Tweak	AES-NI 1 × 4 R2				0.53		0.52		0.52	0.52		⋆ [65]
SKINNY -128-128	Plaintext	Bitsliced-64-block				† 3.78							[8]
SKINNY -128-256	Plaintext + Tweak	Bitsliced-64-block				‡ 2.27							[8]
Deoxys -BC-256	Plaintext	AES-NI	4.74	2.85	1.90	1.43	1.18	1.07	1.01			0.96	[42]
	Plaintext + Tweak	AES-NI	2.37	1.43	0.95	0.72	0.59	0.54	0.51			0.48	[42]
Kiasu≠ -BC-64	Plaintext	AES-NI	0.97	0.84	0.78	0.76	0.75	0.74					[40]
	Plaintext + Tweak	AES-NI	0.65	0.56	0.52	0.51	0.50	0.49					[40]

- The reference for TNT-AES indicated by ⋆ means that basing on the results of AES in [65] and using Eq. (18), we calculated the presented results for TNT-AES.
- The value for SKINNY-128-256 indicated by ‡ is calculated using the value for SKINNY-128-128 indicated by † basing on a formula similar to Eq. (18).

Hardware Performance. We estimate the hardware performance of TNT-AES with area minimization as optimizations target. The current record of minimized area of AES is kept by the bit-serial implementations provided by Jean *et al.* [39]. Apart from AES, Jean *et al.* also provided bit-serial implementations of another tweakable block cipher SKINNY. Using those state-of-the-art results provided

by Jean *et al.* [39], we estimate the area and latency of TNT-AES and make comparisons with other TBCs. The results are summarized in Table 5.

In the table, results for AES, SKINNY-128-256, and Deoxys-BC-256 are all from existing studies. The results for TNT-AES are calculated using the following method based on the results for AES. Let δ be the number of bits in the data path in all implementations. Let $C_{\texttt{1DFF}}$ be the cost of a 1-bit D flip-flop (D FF), let $C_{\texttt{XOR}}$ be the cost of a 2-input XOR gate, and let $C_{\texttt{MUX}}$ be the cost of a 2-to-1 Multiplexer in a library. We use Table 4 to estimate $C_{\texttt{1DFF}}$, $C_{\texttt{XOR}}$, and $C_{\texttt{MUX}}$ in various libraries.

Table 3. Software performance of AES-128 when rekeying for every block and that of TNT-AES when fixing a key but retweaking for every block, both with plaintexts as data (unlike in Table 2 where we consider both "Plaintext" and "Tweak" as data), and both with help of AES-NI (on an Intel(R) Core(TM) i7-8565U CPU 1.80 GHz, which belongs to products formerly Whiskey Lake).

$\|M\|$ (bytes)	Rekeying in AES-128 (cycles/byte)	Retweaking in TNT-AES (cycles/byte)
128	4.60	1.50
256	4.60	1.00
512	4.60	0.80
1024	4.60	0.70
2048	4.60	0.60
4096	4.60	0.60
8192	4.60	0.60

Compared with implementations of AES, the additional area cost for implementations of TNT-AES comes from the cost for storing a 128-bit tweak and the cost for implementing the XOR with tweak (we ignore the additional cost for the signals controlling the tweak/key inputs). We note that there are cases where as input, the tweak can be sent twice by the external provider. In such cases, extra storage for the tweak can be saved. We note that this is possible for a design without a "tweak-schedule". For other designs, such as that permute the bytes of the tweak, this becomes difficult as it requires this permutation to be followed by external provider if not stored locally. In TNT-AES, there is no tweak-schedule, hence no storage for tweak is required. When storage is required, the 128-bit tweak can be stored using 128 1-bit D FF. To implement the XOR with tweak, besides δ 2-input XOR gates, δ 2-to-1 multiplexers are also required for selecting the bits of tweak after the n_1-th round and the (n_1+n_2)-th round and selecting constant 0 after other rounds. The additional area cost for XOR gates and multiplexers is $\delta \times (C_{\texttt{XOR}} + C_{\texttt{MUX}})$. Thus, additional area cost is $128 \times C_{\texttt{1DFF}} + \delta \times (C_{\texttt{XOR}} + C_{\texttt{MUX}})$ when the tweak needs to be stored locally, and $\delta \times (C_{\texttt{XOR}} + C_{\texttt{MUX}})$ otherwise. To get a better view of the performances, we provide the gate sizes for both scenarios.

Table 4. The (estimated) cost (in Gate Equivalent, GE) of regular flip-flops, scan flip-flops, 2-input XOR gates, and 2-to-1 Multiplexers in different libraries.

	UMC 180	UMC 130	UMC 90	Ngate 45	IBM 130
1-bit D FF	4.67	5.00	4.25	5.67	4.25
1-bit Scan FF	6.00	6.25	5.75	7.67	5.50
1-bit XOR	2.67	2.75	2.50	2.00	2.00
2-to-1 MUX	2.33	2.25	2.25	2.33	2.25

Table 5. A table of comparison with other TBCs on hardware area (in GEs) and latency (all TBCs are with 128-bit block, 128-bit master key, and 128-bit tweak)

Cipher	data path Bits	UMC 180 GEs	UMC 130 GEs	UMC 90 GEs	Ngate 45 GEs	IBM 130 GEs	Latency Cycles	Ref.
AES	1	1727	1902	1596	1982	1560	∗ 1776/168	[39]
	2	1796	1992	1667	2054	1625	∗ 888/84	[39]
	4	1920	2168	1784	2146	1731	∗ 520/50	[39]
	8	2112	2360	1968	2337	1912	∗ 282/27	[39]
	8	2400	3574	2292	2768	2182	∗ 226/21	[63]
TNT- AES	1	† 2330/1732	† 2547/1907	† 2145/1601	† 2712/1986	† 2108/1564	3152	⋆ [39]
	2	† 2404/1806	† 2642/2002	† 2221/1677	† 2788/2063	† 2178/1634	1576	⋆ [39]
	4	† 2538/1940	† 2828/2188	† 2347/1803	† 2889/2163	† 2292/1748	932	⋆ [39]
	8	† 2750/2152	† 3040/2400	† 2550/2006	† 3097/2372	† 2490/1946	502	⋆ [39]
	8	† 3038/2440	† 4254/3614	† 2874/2330	† 3528/2803	† 2760/2216	394	⋆ [63]
SKINNY-128-256	1	2082	2278	1937	2501	1905	8448	[39]
	2	2130	2318	1988	2554	1941	4224	[39]
	4	2248	2433	2108	2694	2044	2112	[39]
	8	2456	2662	2325	2949	2223	1056	[39]
Deoxys-BC-256	8	2860					338	[42]

∗ In column 8 for AES, in the form x/y, x is the number of cycles taken by the entire encryption, y is the number of cycles taken by one full round which is used to estimate the latency of TNT-AES.
† In column 3–7 for TNT-AES, in the form x/y, x is the area when the tweak is stored locally, y is the area when the tweak is not stored locally.
⋆ The references for TNT-AES indicated by ⋆ means that basing on the results of AES in these works, we calculated the presented results for TNT-AES.

For latency, selecting and XOR-ing bits of tweak can be implemented in the same clock cycles for AddRoundKey and SubBytes, thus cost no additional cycles. The additional cycle-cost comes from the fact that TNT-AES has more rounds and the last round is complete instead of missing the MixColumns. Thus, to estimate the latency of TNT-AES, we use the clock cycles taken by one full round of AES (denoted by $Cycles_{\texttt{round}}$), times the total number of rounds $(n_1+n_2+n_3)$, plus the cycles taken by the last AddRoundKey ($128/\delta$ cycles), *i.e.*, $Cycles_{\texttt{round}} \times (n_1+n_2+n_3)+128/\delta$, where $Cycles_{\texttt{round}}$ is listed in Table 5 (column 8 for AES).

From Table 5, when the tweak has to be stored locally, the hardware performance of TNT-AES is slightly inferior to those of SKINNY-128-256 and Deoxys-BC-256, otherwise, the hardware performance of TNT-AES can be superior.

Comparison to TAES. Here, we briefly discuss the comparison between the performance of TNT-AES and that of TAES, where TAES is an AES-based TBC used to instantiate ZOCB and ZOTR that are two tweakable blockcipher modes for authenticated encryption with full absorption [3]. TAES tweaks AES-256 by simply replacing the second half part of the secrete key with 128-bit tweak and keeping all other operations and parameters unchanged. Thus, it has 14 rounds, 128-bit blocks, 128-bit keys, and 128-bit tweaks.

Because TNT-AES consists of 18 AES-rounds, *i.e.*, 4 more rounds than TAES, under the use-cases where both the key and the tweak are fixed and all sub-tweaks/sub-keys can be precomputed, TAES outperforms TNT-AES. Whereas, for other use-cases where retweaking is necessary, TNT-AES is expected to perform better. The reasons are as follows. TNT-AES has no tweak-schedule, while that for TAES is related to the key-schedule for AES-256. For software implementation using AES-NI, the instruction for one-round encryption outperforms that for the key-schedule as mentioned above. Thus, in retweaking use-cases, TNT-AES will be much faster than TAES. For hardware implementation, when the 128-bit tweak can be stored in external storage, TNT-AES does not need additional storage to process the tweak. The area requirement is hence much less than that of TAES, which requires local storage to hold and process the tweak.

6 Conclusion and Open Questions

In this paper, we proposed a new mode named TNT for constructing tweakable block ciphers with proven BBB security based on three block ciphers. To demonstrate the effectiveness of the mode, an instantiation based on AES named TNT-AES was proposed, which enjoys the long-standing security of AES, fast software performance due to AES new instructions, and hardware efficiency due to the simplicity of TNT mode. Following the prove-then-prune design strategy, we reduced the number of rounds of the three underlying AES-based block ciphers from 10 for the original AES, to 6, 6, and 6, respectively. Our comprehensive cryptanalysis shows no security issues against TNT-AES, while the reduced number of rounds allow achieving competitive software and hardware performances with existing TBCs designed through modular methods. We expect TNT to be a generic way to turn a block cipher into a tweakable block cipher securely, especially for those lightweight block ciphers with smaller block lengths.

Potential Applications. While TNT-AES only supports n-bit tweaks which seems a limitation compared to $\mathsf{CLRW2}_2$, such a parameter has already been sufficient for many important applications. For example, many TBC-based MACs, including the chaining-via-tweak mode proposed by Liskov *et al.* [54] (its security was later proved optimal by Landecker *et al.* [51]) and the AXU-hash-based MACs proposed by Cogliati *et al.* [19], are exactly built from TBCs with n-bit tweaks, and thus instantiating the TBCs with $\mathsf{CLRW2}_2$ (as done in [51]) clearly wastes power and causes unnecessary efficiency loss. Consequently, TNT-AES

would probably be a better building block. Moreover, TNT-AES could also be used to build BBB secure variable length domain extenders via the construction of Chen *et al.* [13] or double-length block cipher via the construction of Coron *et al.* [21]. As discussed in [13], such construction may further motivate highly secure format-preserving encryption schemes might be a very valuable alternative to the recently broken standards.

Besides, TNT-AES could be used to replace the TBC module in the standard OCB3 mode and the OTR mode [62] (the 2nd round candidate during CAESAR competition). Both modes are optimally secure when the underlying TBC-module is optimal [49,62] but fall down to the birthday bound due to instantiating the TBC with XEX-like constructions [67]. Therefore, once instantiating with TNT-AES, we obtain corresponding variants secure against BBB $2^{2n/3}$ queries in both cases. Consider the application to OCB3 for concreteness. The resulting AE TNT-AES-ΘCB is a ΘCB instance [49] with TNT-AES being its underlying TBC, and the security is boosted from $n/2$ bits of OCB3 to $2n/3$ bits. Perhaps surprisingly, the hardware efficiency might be improved as well: the original OCB3 mode requires to store an AXU hash key $E_K(0)$ during the lifetime of the master key K, which is avoided in TNT-AES-TAE.

We anticipate more such applications, especially when AES-based TBCs are used and constructed from other modes than TNT.

The Security Gap. Although the security of TNT is proven to be $2^{2n/3}$, there is no matching attack – note that Dinur *et al.*'s attack strategy [26] against the 3-round Even-Mansour ciphers does not help here since the permutations in TNT cannot be queried by the adversary, and Mennink's distinguisher [59] does not work directly either due to the $2^{3n/2}$ offline computational complexity besides the $2^{3n/4}$ online query complexity. Then, the same applies to the instantiation TNT-AES. It will be interesting to see the closure of this gap, by either improving the proven security bound or finding a better attack. We leave this as an open problem to the community.

Acknowledgements. We thank the anonymous reviewers for their helpful comments and thank Tetsu Iwata, Eik List and Kazuhiko Minematsu for fruitful discussions. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Strategic Capability Research Centres Funding Initiative, Nanyang Technological University under grant M4082123, Singapore's Ministry of Education under grants RG 18/19 and MOE2019-T2-1-060, and National Natural Science Foundation of China (No. 61961146004). The second author is partially supported by the Program of Qilu Young Scholars (Grant No. 61580089963177) of Shandong University. The fourth author is partially supported by the National Natural Science Foundation of China (Grants No. 61802399, 61802400, 61732021 and 61772519), the Youth Innovation Promotion Association CAS, the National Key Research and Development Project (Grant No. 2018YFA0704704) and Chinese Major Program of National Cryptography Development Foundation (Grant No. MMJJ20180102).

A Subspace Trail Cryptanalysis of TNT-AES

In this section, we discuss subspace-trail-based distinguishers and key-recovery attacks on TNT-AES under the activity pattern $(0, 1, 1)$. Attacking encryption under the pattern $(1, 1, 0)$ can be seen as attacking decryption with the pattern $(0, 1, 1)$. Thus, similar attacks under the pattern $(1, 1, 0)$ can be devised once attacks under the pattern $(0, 1, 1)$ are established. Subspace trail cryptanalysis of TNT-AES under the activity pattern $(0, 1, 1)$ can be compared with subspace trail cryptanalysis of $(n_2 + n_3)$-round AES. The difference lies in that the initial coset of concerned subspace is formed by chosen tweaks instead of by chosen plaintext and elements in the coset will be XOR-ed with the internal state (an unknown constant) c^* which can not be observed during the attack. Besides, the same chosen tweaks are XOR-ed after π_2.

As introduced in Sect. 5.2, a series of attacks on round-reduced (no more than 6 rounds) AES based on subspace trail cryptanalysis and the extended mixture-differential, exchange attacks were proposed in [4–6,31–34]. Among these r-round distinguishers, those which do not require the knowledge of part of the secret key can be directly turned into $(n_1 + r)$-round distinguishers with the same complexity on round-reduced TNT-AES. This can be done by using a unique plaintext p and a structure of tweaks to construct required cosets of concerned subspace at the beginning of π_2. Although the exact cosets are unknown, required relations among input states at the beginning of the active permutation can be constructed using chosen tweaks. For example, when turn the 4-round mixture-differential distinguisher on AES [4,30] into an $(n_1 + 4)$-round distinguisher on TNT-AES, if some chosen tweaks can form mixture quadruples, then after being XOR-ed with a common unknown internal state, the resulting states still keep the relation of being mixture quadruples. There are r-round distinguishers on round-reduced AES that require considering part of the key, which can also be turned into $(n_1 + r)$-round distinguisher on TNT-AES. Take the 5-round impossible-differential distinguishers based on the impossible subspace trail on 4-round AES [33] for example. When we turn it into $(n_1 + 5)$-round distinguisher on TNT-AES, we use a unique plaintext p and structures of chosen tweaks (chosen in the way of choosing plaintexts in the original distinguisher). Then, unlike in the original distinguisher on AES, where we guess the single-byte key difference $k_{0,0} \oplus k_{1,1}$, we guess the single-byte difference $c^*_{0,0} \oplus c^*_{1,1}$, where c^* is the unknown internal state before XOR-ing the tweak. Again, the complexities of these $(n_1 + r)$-round distinguishers on TNT-AES will be almost the same with those r-round distinguishers on AES.

As for key-recovery attacks exploiting those r-round distinguishers on round-reduced AES (*e.g.*, the 5-round key-recovery attack exploiting the 4-round mixture-differential distinguisher [4] and the 6-round key-recovery attack exploiting the 5-round probabilistic mixture-differential distinguisher [30]), they add one round in front of the distinguisher, and guess parts of the whitening key (*e.g.*, key bits in $SR^{-1}(Col(i))$, or say in diagonal space $\mathcal{D}_i$, $i \in \{0, 1, 2, 3\}$) to filter out useful plaintexts from a chosen structure or to classify chosen plaintexts into properly defined sets. Such attacks may not be directly used to construct corresponding attacks on $(n_1 + 1 + r)$-round TNT-AES by guessing part of the subkey, because the internal state is also unknown. However, by guessing the internal state before XOR-ing the tweak, we can recover this unknown state part by part (instead of recovering key bits). Using this recovered internal state, one may further analyze π_1 to recover the key. However, because the dependent unknown values are in the diagonal $SR^{-1}(Col(i))$ that depend on the full state one round before, extending such attacks to cover one more round seems to be difficult. Thus, exploiting current techniques in such attacks on r-round AES-128, an attack on TNT-AES is limited to be no more than $(n_1 + 1 + r)$ rounds. Key-recovery attacks using those $(n_1 + r)$-round $(r \geq 5)$ distinguishers to recovery the subkey in an appended (complete) round seems also very hard. That is because, the considered cosets at the end of the exploited distinguishers are commonly cosets of mixed space $\mathcal{M}_I$ $(I \subseteq \{0, 1, 2, 3\})$, which are mapped into the full state. Thus, in an $(n_1 + r + 1)$-round $(r \geq 5)$ key-recovery attack, checking the distinguishable properties one round before the last round requires guessing the entire key.

Based on these analyses and together with previous analyses of other activity patterns, we believe TNT-AES is strong enough to resist subspace trail attacks.

B Examples of the Related-Tweak Boomerang Distinguishers of TNT-AES

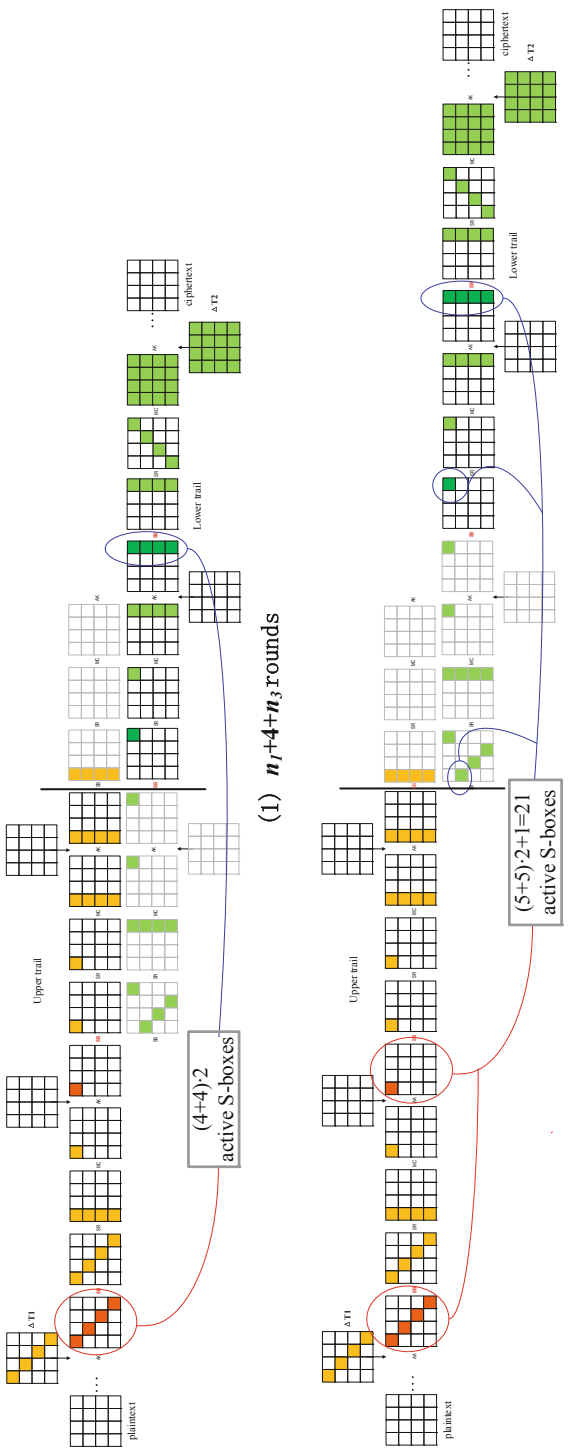

Fig. 3. Examples of the related-tweak boomerang distinguishers of TNT-AES, where the (n_1+4+n_3)-round distinguisher has probability 2^{-96} and the (n_1+5+n_3)-round distinguisher has probability slightly higher than 2^{-128}.

References

1. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (2014–2019). https://competitions.cr.yp.to/caesar.html
2. Aumasson, J.P., et al.: Challenges in authenticated encryption, March 2017. https://chae.cr.yp.to/chae-20170301.pdf
3. Bao, Z., Guo, J., Iwata, T., Minematsu, K.: ZOCB and ZOTR: tweakable blockcipher modes for authenticated encryption with full absorption. IACR Trans. Symmetr. Cryptol. **2019**(2), 1–54 (2019)
4. Bar-On, A., Dunkelman, O., Keller, N., Ronen, E., Shamir, A.: Improved key recovery attacks on reduced-round AES with practical data and memory complexities. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 185–212. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96881-0_7
5. Bardeh, N.G.: A key-independent distinguisher for 6-round AES in an adaptive setting. Cryptology ePrint Archive, Report 2019/945 (2019). https://eprint.iacr.org/2019/945
6. Bardeh, N.G., Rønjom, S.: The exchange attack: how to distinguish six rounds of AES with $2^{88.2}$ chosen plaintexts. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11923, pp. 347–370. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34618-8_12
7. Beierle, C., et al.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 123–153. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_5
8. Beierle, C., et al.: The SKINNY family of block ciphers and its low-latency variant MANTIS. Cryptology ePrint Archive, Report 2016/660 (2016). http://eprint.iacr.org/2016/660
9. Bernstein, D.J., Schwabe, P.: New AES software speed records. In: Chowdhury et al. [15], pp. 322–336 (2008)
10. Biryukov, A., Wagner, D.: Slide attacks. In: Knudsen [48], pp. 245–259 (1999)
11. Biryukov, A., Wagner, D.: Advanced slide attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 589–606. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_41
12. Chakraborti, A., Datta, N., Jha, A., Lopez, C.M., Nandi, M., Sasaki, Y.: Elastictweak: a framework for short tweak tweakable block cipher. Cryptology ePrint Archive, Report 2019/440 (2019). https://eprint.iacr.org/2019/440
13. Chen, Y.L., Mennink, B., Nandi, M.: Short variable length domain extenders with beyond birthday bound security. In: Peyrin and Galbraith [66], pp. 244–274 (2018)
14. Cheon, J.H., Takagi, T. (eds.): Advances in Cryptology - ASIACRYPT 2016, Part I, Hanoi, Vietnam, 4–8 December 2016. LNCS, vol. 10031. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6
15. Chowdhury, D.R., Rijmen, V., Das, A. (eds.): Progress in Cryptology - INDOCRYPT 2008: 9th International Conference in Cryptology in India, Kharagpur, India 14–17 December 2008. LNCS, vol. 5365. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89754-5
16. Cid, C., Huang, T., Peyrin, T., Sasaki, Y., Song, L.: Boomerang connectivity table: a new cryptanalysis tool. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 683–714. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_22

17. Cogliati, B.: Tweaking a block cipher: multi-user beyond-birthday-bound security in the standard model. Des. Codes Cryptogr. **86**(12), 2747–2763 (2018). https://doi.org/10.1007/s10623-018-0471-8
18. Cogliati, B., Lampe, R., Seurin, Y.: Tweaking even-mansour ciphers. In: Gennaro and Robshaw [29], pp. 189–208 (2015)
19. Cogliati, B., Lee, J., Seurin, Y.: New constructions of MACs from (tweakable) block ciphers. IACR Trans. Symmetr. Cryptol. **2017**(2), 27–58 (2017)
20. Cogliati, B., Seurin, Y.: Beyond-birthday-bound security for tweakable even-mansour ciphers with linear tweak and key mixing. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 134–158. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_6
21. Coron, J.-S., Dodis, Y., Mandal, A., Seurin, Y.: A domain extender for the ideal cipher. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 273–289. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_17
22. Daemen, J., Rijmen, V.: AES Proposal: Rijndael (1999)
23. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer, Heidelberg (2002). https://doi.org/10.1007/978-3-662-04722-4
24. Dai, W., Hoang, V.T., Tessaro, S.: Information-theoretic indistinguishability via the Chi-Squared method. In: Katz and Shacham [46], pp. 497–523 (2017)
25. Derbez, P., Fouque, P.-A., Jean, J.: Improved key recovery attacks on reduced-round, in the single-key setting. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 371–387. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_23
26. Dinur, I., Dunkelman, O., Keller, N., Shamir, A.: Key recovery attacks on 3-round even-mansour, 8-step LED-128, and Full AES^2. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 337–356. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_18
27. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 278–299. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_16
28. Dinur, I., Shamir, A.: Breaking grain-128 with dynamic cube attacks. In: Joux [45], pp. 167–187 (2011)
29. Gennaro, R., Robshaw, M.J.B. (eds.): Advances in Cryptology - CRYPTO 2015, Part I, Santa Barbara, CA, USA, 16–20 August 2015. LNCS, vol. 9215. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7
30. Grassi, L.: Structural truncated differential attacks on round-reduced AES. Cryptology ePrint Archive, Report 2017/832 (2017). http://eprint.iacr.org/2017/832
31. Grassi, L.: Mixture differential cryptanalysis: a new approach to distinguishers and attacks on round-reduced AES. IACR Trans. Symmetr. Cryptol. **2018**(2), 133–160 (2018)
32. Grassi, L., Rechberger, C., Rønjom, S.: Subspace trail cryptanalysis and its applications to AES. IACR Trans. Symmetr. Cryptol. **2016**(2), 192–225 (2016). http://tosc.iacr.org/index.php/ToSC/article/view/571
33. Grassi, L., Rechberger, C., Rønjom, S.: Subspace trail cryptanalysis and its applications to AES. Cryptology ePrint Archive, Report 2016/592 (2016). http://eprint.iacr.org/2016/592
34. Grassi, L., Rechberger, C., Rønjom, S.: A new structural-differential property of 5-Round AES. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 289–317. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_10

35. Gueron, S., Lindell, Y.: Better bounds for block cipher modes of operation via nonce-based key derivation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017: 24th Conference on Computer and Communications Security, 31 October–2 November 2017, pp. 1019–1036. ACM Press, Dallas (2017)
36. Gueron, S., Lindell, Y., Nof, A., Pinkas, B.: Fast garbling of circuits under standard assumptions. J. Cryptol. **31**(3), 798–844 (2018)
37. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 15–44. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_2
38. Iwata, T., Minematsu, K., Peyrin, T., Seurin, Y.: ZMAC: a fast tweakable block cipher mode for highly secure message authentication. In: Katz and Shacham [46], pp. 34–65
39. Jean, J., Moradi, A., Peyrin, T., Sasdrich, P.: Bit-sliding: a generic technique for bit-serial implementations of SPN-based primitives. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 687–707. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_33
40. Jean, J., Nikolić, I., Peyrin, T.: KIASU v1. Additional first-round candidates of CAESAR compeition (2014). https://competitions.cr.yp.to/caesar-submissions.html
41. Jean, J., Nikolić, I., Peyrin, T.: Tweaks and keys for block ciphers: the TWEAKEY framework. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 274–288. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8_15
42. Jean, J., Nikolić, I., Peyrin, T., Seurin, Y.: Deoxys-II. Finalist of CAESAR compeition (2014). https://competitions.cr.yp.to/caesar-submissions.html
43. Jha, A., List, E., Minematsu, K., Mishra, S., Nandi, M.: XHX - a framework for optimally secure tweakable block ciphers from classical block ciphers and universal hashing. Cryptology ePrint Archive, Report 2017/1075 (2017). https://eprint.iacr.org/2017/1075
44. Jha, A., Nandi, M.: Tight security of cascaded LRW2. Cryptology ePrint Archive, Report 2019/1495 (2019). https://eprint.iacr.org/2019/1495
45. Joux, A. (ed.): Fast Software Encryption - FSE 2011, Lyngby, Denmark, 13–16 February 2011. LNCS, vol. 6733. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21702-9
46. Katz, J., Shacham, H. (eds.): Advances in Cryptology - CRYPTO 2017, Part III, Santa Barbara, CA, USA, 20–24 August 2017. LNCS, vol. 10403. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-63688-7
47. Keliher, L., Sui, J.: Exact maximum expected differential and linear probability for 2-round advanced encryption standard (AES). Cryptology ePrint Archive, Report 2005/321 (2005). http://eprint.iacr.org/2005/321
48. Knudsen, L.R. (ed.): Fast Software Encryption - FSE 1999, Germany, Rome, Italy 24–26 March 1999. LNCS, vol. 1636. Springer, Heidelberg (1999)
49. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux [45], pp. 306–327 (2011)
50. Lampe, R., Seurin, Y.: Tweakable blockciphers with asymptotically optimal security. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 133–151. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43933-3_8

51. Landecker, W., Shrimpton, T., Terashima, R.S.: Tweakable blockciphers with beyond birthday-bound security. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 14–30. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_2
52. Leander, G., Abdelraheem, M.A., AlKhzaimi, H., Zenner, E.: A cryptanalysis of PRINTcipher: the invariant subspace attack. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 206–221. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_12
53. Lee, B., Lee, J.: Tweakable block ciphers secure beyond the birthday bound in the ideal cipher model. In: Peyrin and Galbraith [66], pp. 305–335
54. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 31–46. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_3
55. Lu, J., Dunkelman, O., Keller, N., Kim, J.: New impossible differential attacks on AES. In: Chowdhury et al. [15], pp. 279–293 (2011)
56. Mennink, B.: Optimally secure tweakable blockciphers. Cryptology ePrint Archive, Report 2015/363 (2015). http://eprint.iacr.org/2015/363
57. Mennink, B.: XPX: generalized tweakable Even-Mansour with improved security guarantees. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 64–94. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_3
58. Mennink, B.: Insuperability of the standard versus ideal model gap for tweakable blockcipher security. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 708–732. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_24
59. Mennink, B.: Towards tight security of cascaded LRW2. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part II. LNCS, vol. 11240, pp. 192–222. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03810-6_8
60. Mennink, B., Neves, S.: Optimal PRFs from blockcipher designs. IACR Trans. Symmetr. Cryptol. **2017**(3), 228–252 (2017)
61. Minematsu, K.: Beyond-birthday-bound security based on tweakable block cipher. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 308–326. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03317-9_19
62. Minematsu, K.: Parallelizable rate-1 authenticated encryption from pseudorandom functions. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 275–292. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_16
63. Moradi, A., Poschmann, A., Ling, S., Paar, C., Wang, H.: Pushing the limits: a very compact and a threshold implementation of AES. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 69–88. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_6
64. NIST: Lightweight Cryptography Competition (2019). https://csrc.nist.gov/projects/lightweight-cryptography
65. Park, J.H., Lee, D.H.: FACE: Fast AES CTR mode encryption techniques based on the reuse of repetitive data. IACR Trans. Cryptogr. Hardw. Embedd. Syst. **2018**(3), 469–499 (2018). https://tches.iacr.org/index.php/TCHES/article/view/7283
66. Peyrin, T., Galbraith, S. (eds.): Advances in Cryptology - ASIACRYPT 2018, Part I, Brisbane, Queensland, Australia, 2–6 December 2018. LNCS, vol. 11272. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-030-03326-2

67. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30539-2_2
68. Rønjom, S., Bardeh, N.G., Helleseth, T.: Yoyo tricks with AES. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 217–243. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_8
69. Shrimpton, T., Terashima, R.S.: Salvaging weak security bounds for blockcipher-based constructions. In: Cheon and Takagi [14], pp. 429–454 (2016)
70. Song, L., Qin, X., Hu, L.: Boomerang connectivity table revisited. IACR Trans. Symmetr. Cryptol. **2019**(1), 118–141 (2019)
71. Sun, B., Liu, M., Guo, J., Rijmen, V., Li, R.: Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 196–213. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_8
72. Sun, B., et al.: Links among impossible differential, integral and zero correlation linear cryptanalysis. In: Gennaro and Robshaw [29], pp. 95–115 (2015)
73. Sun, S., et al.: Analysis of AES, SKINNY, and others with constraint programming. IACR Trans. Symmetr. Cryptol. **2017**(1), 281–306 (2017)
74. Todo, Y.: Integral cryptanalysis on full MISTY1. In: Gennaro and Robshaw [29], pp. 413–432 (2015)
75. Todo, Y., Morii, M.: Bit-based division property and application to Simon family. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 357–377. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-52993-5_18
76. Wagner, D.: The boomerang attack. In: Knudsen [48], pp. 156–170 (1999)
77. Wang, L., Guo, J., Zhang, G., Zhao, J., Gu, D.: How to build fully secure tweakable blockciphers from classical blockciphers. In: Cheon and Takagi [14], pp. 455–483 (2016)
78. Wu, H.: Hongjun's optimized C-code for AES-128 and AES-256. eSTREAM project (2007). http://www.ecrypt.eu.org/stream/svn/viewcvs.cgi/ecrypt/trunk/benchmarks/aes-ctr/aes-128/hongjun/v1/?rev=203#dirlist