Efficient Instances of Docked Double Decker With AES, and Application to Authenticated Encryption

Christoph Dobraunig[1], Krystian Matusiewicz[2], Bart Mennink[3], and Alexander Tereschenko[2]

[1] Intel Labs, Hillsboro, USA
christoph.dobraunig@intel.com
[2] Intel Corporation, Gdańsk, Poland
krystian.matusiewicz@intel.com, aleksandr.v.tereschenko@intel.com
[3] Radboud University, Nijmegen, The Netherlands
b.mennink@cs.ru.nl

Abstract. A tweakable wide blockcipher is a construction which behaves in the same way as a tweakable blockcipher, with the difference that the actual block size is flexible. Due to this feature, a tweakable wide blockcipher can be directly used as a strong encryption scheme that provides full diffusion when encrypting plaintexts to ciphertexts and vice versa. Furthermore, it can be the basis of authenticated encryption schemes fulfilling the strongest security notions. In this paper, we present three instantiations of the docked double decker tweakable wide blockcipher: *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES*. These instances exclusively use similar building blocks as AES-GCM (AES and finite field multiplication), are designed for maximal parallelism, and hence, can make efficient use of existing hardware accelerators. *ddd-AES* is a birthday bound secure scheme, and *ddd-AES*$^+$ is an immediate generalization to allow for variable length tweaks. *bbb-ddd-AES* achieves security beyond the birthday bound provided that the same tweak is not used too often. Moreover, *bbb-ddd-AES* builds upon a novel conditionally beyond birthday bound secure pseudorandom function, a tweakable variant of the XOR of permutations, facilitating in the need to include a tweak in the AES evaluations without sacrificing flexibility in docked double decker. We furthermore introduce an authenticated encryption mode *aaa* specifically tailored to be instantiated with *ddd-AES* and *bbb-ddd-AES*, where special attention is given to how the nonce and associated data can be processed. We prove that this mode is secure in the nonce-respecting setting, in the nonce-misuse setting, as well as in the setting where random nonces are used. We finally present a comparison with other tweakable wide blockciphers, give a high-level idea of the efficiency potential of our schemes, and provide benchmarks that confirm this idea.

Keywords: symmetric cryptography, tweakable wide blockcipher, accordion cipher mode, docked double decker, tweakable XOR of permutations, authenticated encryption.

1 Introduction

1.1 Motivation

The US NIST (National Institute of Standards and Technology) has standardized a number of pure confidentiality modes of operation, like ECB, CBC, CFB, OFB, and CTR in NIST SP 800-38A [20], and XTS-AES in NIST SP 800-38E [21]. Although these modes see a wide-spread use in applications, they come with their own limitations. Most notably, none of the above-mentioned encryption methods provides full diffusion for encryption as well as decryption if the data to be encrypted exceeds a few blocks. This stands in sharp contrast with the fact that for modes that just provide confidentiality, full diffusion behavior is often a practical security benefit, since it limits the ability of an attacker to target specific fractions of the encrypted data [5].

In addition, with modern Internet- and cloud-scale data creation and processing volumes being routinely measured in exabytes and approaching zettabytes, many existing ciphers become a bottleneck and sometimes even a security risk, because they were not designed to be used at such scale. As indicated in some cloud service provider (CSP) comments [32, 35, 47], the limitations of block size and corresponding birthday bounds lead to many standardized mainstream ciphers and modes becoming too brittle when used for such large data sets. To protect that data while complying with cipher key/nonce pair uniqueness requirements and data processing volume limitations, CSPs are forced to either employ inefficient techniques like frequent rekeying (every week or two, down to potentially mere seconds), or use tricks like having a static nonce and rekeying for every message.

1.2 Tweakable Wide Blockciphers (Accordion Cipher Modes)

A very suitable solution, or building block for a solution to the aforementioned problems, are tweakable wide blockciphers (also referred to as accordion cipher mode [12]) with beyond birthday security (e.g. more than 2^{64} blocks in case AES is used as a building block). Not surprisingly, in the recent third NIST workshop on blockcipher modes of operation in 2023, the organizers stated that "*NIST is particularly interested in discussing the possibility of standardizing a tweakable wide block encryption technique that could support a large range of input lengths.*" [41].

Indeed, a tweakable wide blockcipher extends the definition of a tweakable blockcipher [34] to arbitrarily large input and output size, this way allowing for flexibility in the block size. Note that such a tweakable wide blockcipher also, unlike existing modes such as ECB, CBC, CFB, OFB, CTR, and XTS-AES, allows for full diffusion. This way, it serves as viable drop-in replacement of these modes in many applications.

Furthermore, it can serve as the basis of an authenticated encryption scheme, or directly as authenticated encryption scheme, by either appending the nonce to the plaintext or putting the nonce in the tweak and appending zeros to the plaintext to strengthen authenticity [28]. The resulting construction essentially allows for flexibly sized tags and nonces, and has the potential to be misuse resistant and context committing.

Other desirable properties are performance and beyond birthday bound security, and this brings us to our contribution.

1.3 Our Contributions

We present three different tweakable wide blockciphers: *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES*. These are all based on the same components as used in many NIST standardized schemes. Notably, they are based on the AES blockcipher [16, 17], as well as on operations in binary extension fields as used by GHash in AES-GCM [37, 52].

Our schemes are based on the docked double decker mode of Gunsing et al. [23] (see Fig. 1). Docked double decker operates on top of a universal hash function H and a pseudorandom function F, and has the feature that it allows to provide beyond birthday bound security assuming it is not used with the same tweak too often. All our instances *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES* take *Polyval* [22] as universal hash function. The choice of pseudorandom function is different for the constructions:

- In *ddd-AES*, the pseudorandom function is based on an *XE*-style [50] tweakable blockcipher using a fixed length tweak, itself built on top of AES, evaluated in counter mode (see Sec. 4.2). The resulting construction achieves birthday bound security;
- *ddd-AES*$^+$ is very similar to *ddd-AES* but is designed to accommodate variable-length tweaks.[4] Its pseudorandom function is again based on an *XE*-style tweakable blockcipher, but where the mask is replaced by a sum of blockcipher calls on individual tweak blocks. We prove that the resulting construction achieves birthday bound security;
- In *bbb-ddd-AES*, to accommodate the fixed length tweak, we wished to instantiate the pseudorandom function with a slightly compressing construction on top of AES that achieves beyond birthday bound security. To this end, we took the *XORP* construction as used in CENC [30], and extended it to include a tweak. In detail, this construction $\widetilde{XORP}$ extends *XORP* by including the tweak in an *XE*-style [50] manner (Sec. 4.4). Although $\widetilde{XORP}$ also achieves birthday bound security in the general case, it achieves beyond birthday bound $2n/3$-bit security assuming it is, just like *ddd*, not used with the same tweak too often. We remark that this result — the introduction and security analysis of $\widetilde{XORP}$ as a "tweakable PRF" — is of independent interest.

On top of this, we also introduce an authenticated encryption mode *aaa*. *aaa* is specifically designed to work well when using tweakable wide blockciphers

[4] We remark that, at the NIST workshop on the requirements for an accordion mode in 2024, there was significant support for variable-length tweaks [55].

with a fixed tweak length like $\mathit{ddd}\text{-}\mathit{AES}$ and $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$ (see Sec. 8). The design is inspired by the idea [28] to concatenate τ zeros to the plaintext before encrypting, but we significantly extended this idea to (i) capture associated data and (ii) to accommodate for nonces that could be larger than the limited tweak size of $\mathit{ddd}\text{-}\mathit{AES}$ and $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$. We prove that the aaa mode is secure in the nonce-respecting setting, in the nonce-misuse setting, as well as in the setting where random nonces are used. We remark that $\mathit{ddd}\text{-}\mathit{AES}^+$ already natively allows for larger tweaks, and the construction of Hoang et al. [28] (which we also revisit in Sec. 8) already does the job.

We furthermore demonstrate that $\mathit{ddd}\text{-}\mathit{AES}$ and $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$ compare favorably with existing designs [9,10,15,25–27,36,39,53]. In fact, there is no other blockcipher based scheme to date achieving beyond birthday bound security in a reasonably efficient way (there does exist a number of tweakable blockcipher based schemes [8, 54]). We also provide a high-level intuition why the $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$ design can be implemented efficiently, and we back this up with implementation figures.

1.4 Outline

We first discuss some preliminaries in Sec. 2. The docked double decker construction of Gunsing et al. [23] is recalled in Sec. 3. We specify $\mathit{ddd}\text{-}\mathit{AES}$, $\mathit{ddd}\text{-}\mathit{AES}^+$, and $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$ in Sec. 4, with the description of *Polyval* (as used in each of $\mathit{ddd}\text{-}\mathit{AES}$, $\mathit{ddd}\text{-}\mathit{AES}^+$, and $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$) in Sec. 4.1, the description of the pseudorandom function used in $\mathit{ddd}\text{-}\mathit{AES}$ in Sec. 4.2, the description of the pseudorandom function used in $\mathit{ddd}\text{-}\mathit{AES}^+$ in Sec. 4.3, and the description of the pseudorandom function used in $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$ in Sec. 4.4. The security of $\mathit{ddd}\text{-}\mathit{AES}$, $\mathit{ddd}\text{-}\mathit{AES}^+$, and $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$ is analyzed in Sec. 5, with the security proof of the pseudorandom function of $\mathit{ddd}\text{-}\mathit{AES}^+$ in Sec. 6 and the security proof of $\widetilde{\mathit{XORP}}$ (which is of independent interest) in Sec. 7. We present the application of our scheme to authenticated encryption in Sec. 8. We give interpretations of the bound of $\mathit{bbb}\text{-}\mathit{ddd}\text{-}\mathit{AES}$ in Sec. 9. In Sec. 10, we provide a high-level comparison of our schemes with state-of-the-art blockcipher based constructions. We discuss the efficiency of our schemes in Sec. 11. We conclude in Sec. 12.

2 Preliminaries

For $n \in \mathbb{N}$, $\{0,1\}^n$ denotes the set of bit strings of length n, and $\{0,1\}^* = \cup_{n=0}^{\infty}\{0,1\}^n$ denotes the set of bit strings of arbitrary length. For a bit string $X \in \{0,1\}^n$ and for $m \in \mathbb{N}$ such that $m \leq n$, we denote by $\mathrm{left}_m(X)$ the leftmost m bits of X and by $\mathrm{right}_m(X)$ the rightmost m bits of X. For a finite set $\mathcal{S}$, we denote by $s \xleftarrow{\$} \mathcal{S}$ the uniform random selection of s from $\mathcal{S}$. For $n, p \in \mathbb{N}$, we denote by $(n)_p = n(n-1)\cdots(n-p+1)$ the falling factorial.

2.1 Tweakable Wide Blockciphers

Our tweakable wide blockciphers will be parameterized by a value $n \in \mathbb{N}$. This will also be called the **block size**. They will require plaintexts of size at least $2n$ bits. Our tweakable wide blockciphers will also be parameterized by a key size $\kappa \in \mathbb{N}$ and a tweak size $w \in \mathbb{N}$. Finally, to formally argue security, we also limit the maximum size of an input plaintext or output ciphertext to some value $\ell_{\max} \in \mathbb{N}$ such that $\ell_{\max} \geq 2n$. We define the plaintext and ciphertext space to

$$\mathcal{S} := \bigcup_{i=2n}^{\ell_{\max}} \{0,1\}^i \,. \tag{1}$$

A tweakable wide blockcipher $TWBC : \{0,1\}^\kappa \times \{0,1\}^w \times \mathcal{S} \to \mathcal{S}$ is a family of permutations on $\mathcal{S}$ indexed by key $K \in \{0,1\}^\kappa$ and tweak $W \in \{0,1\}^w$. In other words, $TWBC$ satisfies the property that for fixed $K \in \{0,1\}^\kappa$ and $W \in \{0,1\}^w$,

$$TWBC_{K,W}(\cdot) := TWBC(K, W, \cdot)$$

is a length-preserving bijection. Its inverse for fixed K and W is denoted by $TWBC^{-1}_{K,W}$. We slightly abuse notation and denote by $TWBC_K$ and $TWBC^{-1}_K$ the interfaces where the adversary still has the freedom to choose the tweak.

Define by $\mathrm{perm}(w, 2n : \ell_{\max})$ the family of all length-preserving bijections on $\mathcal{S}$ of (1). The security of a tweakable wide blockcipher $TWBC$ is defined by how hard it is for an adversary $\mathcal{A}$ to distinguish $TWBC_K$ for a random and secret key $K \xleftarrow{\$} \{0,1\}^\kappa$ from a tweakable wide random permutation $TWRP \xleftarrow{\$} \mathrm{perm}(w, 2n : \ell_{\max})$:

$$\mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}) = \left| \mathbf{Pr}\left(\mathcal{A}^{TWBC_K, TWBC^{-1}_K} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{TWRP, TWRP^{-1}} = 1\right) \right| \,, \tag{2}$$

where the probabilities are taken over $K \xleftarrow{\$} \{0,1\}^\kappa$, $TWRP \xleftarrow{\$} \mathrm{perm}(w, 2n : \ell_{\max})$, and the random coins of $\mathcal{A}$. The adversary is typically bounded by a certain number of queries q, and a total data complexity σ that counts the total amount of output data bits. Here, we remark that the amount of input data bits equals the amount of output data bits plus the tweak, the latter of which is of fixed size for each of the q queries. The adversary is also bounded by a certain amount of time in which it can make offline evaluations, but this time is not explicitly included.

2.2 Pseudorandom Permutations

A blockcipher $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ is a family of permutations on $\{0,1\}^n$ indexed by key $K \in \{0,1\}^\kappa$. We denote $E_K(\cdot) = E(K, \cdot)$, and its inverse for fixed K is denoted by E^{-1}_K.

Define by $\mathrm{perm}(n)$ the family of all bijections on $\{0,1\}^n$. The security of a blockcipher E is defined by how hard it is for an adversary $\mathcal{A}$ to distinguish

E_K for a random and secret key $K \xleftarrow{\$} \{0,1\}^\kappa$ from a random permutation $RP \xleftarrow{\$} \text{perm}(n)$:

$$\mathbf{Adv}_E^{\text{prp}}(\mathcal{A}) = \left|\mathbf{Pr}\left(\mathcal{A}^{E_K} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{RP} = 1\right)\right| , \tag{3}$$

where the probabilities are taken over $K \xleftarrow{\$} \{0,1\}^\kappa$, $RP \xleftarrow{\$} \text{perm}(n)$, and the random coins of $\mathcal{A}$. The adversary is typically bounded by a certain number of queries q. Note that each query is of fixed size n bits.

2.3 Pseudorandom Functions

Let $a, b \in \mathbb{N} \cup \{*\}$. A pseudorandom function $F : \{0,1\}^\kappa \times \{0,1\}^a \to \{0,1\}^b$ is a family of functions from $\{0,1\}^a$ to $\{0,1\}^b$ indexed by key $K \in \{0,1\}^\kappa$. We denote $F_K(\cdot) = F(K, \cdot)$.

Define by $\text{func}(a, b)$ the family of all functions from $\{0,1\}^a$ to $\{0,1\}^b$. The security of a pseudorandom function F is defined by how hard it is for an adversary $\mathcal{A}$ to distinguish F_K for a random and secret key $K \xleftarrow{\$} \{0,1\}^\kappa$ from a random function $RF \xleftarrow{\$} \text{func}(a, b)$:

$$\mathbf{Adv}_F^{\text{prf}}(\mathcal{A}) = \left|\mathbf{Pr}\left(\mathcal{A}^{F_K} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{RF} = 1\right)\right| , \tag{4}$$

where the probabilities are taken over $K \xleftarrow{\$} \{0,1\}^\kappa$, $RF \xleftarrow{\$} \text{func}(a, b)$ (lazily-sampled), and the random coins of $\mathcal{A}$. The adversary is typically bounded by a certain number of queries q, and a total output data complexity σ that counts the total amount of output data bits. Here, we remark that we will always use F on fixed input size and on varying output size.

In our case, the input to the function F may consist of a comma-separated list of multiple inputs. To be precise, we will use a function F that operates on a κ-bit key K, an n-bit input I, a domain separator nibble B, and a w-bit tweak W, that produces a variable length output O:

$$F(K, I, B, W) = O\,.$$

The function F internally concatenates I, B, and W.

2.4 Universal Hash Functions

Let $a, b \in \mathbb{N} \cup \{*\}$. A family of hash functions $H : \{0,1\}^\kappa \times \{0,1\}^a \to \{0,1\}^b$ is called ϵ-XOR-universal if for any two distinct $X, X' \in \{0,1\}^a$ and any $Y \in \{0,1\}^b$,

$$\mathbf{Pr}\left(H(K, X) \oplus H(K, X') = Y\right) \leq \epsilon\,,$$

where the probability is taken over $K \xleftarrow{\$} \{0,1\}^\kappa$. It is called ϵ-universal if this condition holds for $Y = 0^b$.

2.5 Patarin's H-Coefficient Technique

Consider any two oracles $\mathcal{O}$ and $\mathcal{P}$, and a deterministic adversary $\mathcal{A}$ that has query access to either of these oracles, and write

$$\mathbf{Adv}(\mathcal{A}) = \left|\mathbf{Pr}\left(\mathcal{A}^{\mathcal{O}} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\mathcal{P}} = 1\right)\right| . \tag{5}$$

The adversary can make q queries, and its communication with its oracle is recorded in a transcript τ. Denote by $X_{\mathcal{O}}$ the probability distribution of transcripts in interaction with $\mathcal{O}$, and similarly $X_{\mathcal{P}}$ the probability distribution of transcripts in interaction with $\mathcal{P}$. A transcript τ is called attainable if $\mathbf{Pr}\left(X_{\mathcal{P}} = \tau\right) > 0$, and we denote by $\mathcal{T}$ the set of all attainable transcripts.

Patarin's H-coefficient technique [11, 42, 44] states the following:

Theorem 1 (H-coefficient technique). *Let $\delta, \varepsilon \in [0,1]$. Consider a partition $\mathcal{T} = \mathcal{T}_{\mathrm{bad}} \cup \mathcal{T}_{\mathrm{good}}$ of the set of attainable transcripts such that*

- $\mathbf{Pr}\left(X_{\mathcal{P}} \in \mathcal{T}_{\mathrm{bad}}\right) \leq \delta$,
- *for all* $\tau \in \mathcal{T}_{\mathrm{good}}$, $\dfrac{\mathbf{Pr}\left(X_{\mathcal{O}} = \tau\right)}{\mathbf{Pr}\left(X_{\mathcal{P}} = \tau\right)} \geq 1 - \varepsilon$.

Then, the distinguishing advantage of (5) satisfies $\mathbf{Adv}(\mathcal{A}) \leq \delta + \varepsilon$.

3 Docked Double Decker

Let $\kappa, w, n, \ell_{\max}, m_{\max} \in \mathbb{N}$ such that $2n \leq \ell_{\max}$ and $m_{\max} = \lceil \ell_{\max}/n \rceil$. In this paper, we propose instantiations of the docked double decker (*ddd*) of Gunsing et al. [23]. The scheme is depicted in Fig. 1. It gets as input two keys $K \in \{0,1\}^{\kappa}$, and $L \in \{0,1\}^{n}$, a tweak $W \in \{0,1\}^{w}$, and a plaintext $P \in \mathcal{S}$ of size at least $2n$ bits and at most $\ell_{\max}$ bits (see (1)). The plaintext P is parsed as $P = T \| U \| V$, where T and V are both n-bit long. Then, a four-round structure based on two independent instances of a pseudorandom function $F_K : \{0,1\}^{n+4+w} \to \{0,1\}^{*}$ and two instances of a universal hash function $H_L : \{0,1\}^{*} \to \{0,1\}^{n}$ is evaluated to obtain the ciphertext $C = X \| Y \| Z$, where X and Z are n-bit long and Y matches the size of U. We denote this as

$$ddd_{K,L}^{F,H}(W, T \| U \| V) = X \| Y \| Z \,. \tag{6}$$

We remark that we have slightly deviated from the specification of Gunsing et al. [23] in the sense that we do not use two different keys for F but rather use domain separation. However, their analysis directly carries over. In detail, Gunsing et al. [23] proved security under the assumption that the function F is a pseudorandom function (PRF) and H a blinded keyed hash function. An XOR-universal hash function is a specific type of blinded keyed hash function, and we will adopt a simplification of their result to XOR-universal hash functions.

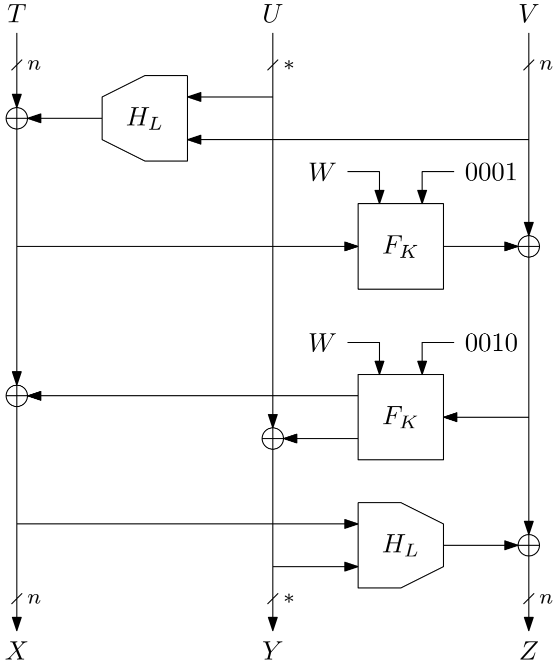


Fig. 1: The docked double decker construction.

Theorem 2 (Gunsing et al. [23, Theorem 1]). *Consider the docked double decker construction ddd on top of a pseudorandom function $F : \{0,1\}^\kappa \times \{0,1\}^{n+4+w} \to \{0,1\}^*$ and an ϵ-XOR-universal hash function family $H : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$. For any adversary $\mathcal{A}$ making at most q queries, each of size at least $2n$ and at most $\ell_{\max}$ bits, and in total of size at most σ bits, and where q_W is the number of queries made for tweak $W \in \{0,1\}^w$, we have*

$$\mathbf{Adv}_{ddd}^{\mathrm{stwprp}}(\mathcal{A}) \leq \mathbf{Adv}_F^{\mathrm{prf}}(\mathcal{A}') + \sum_{W \in \{0,1\}^w} \binom{q_W}{2} \cdot \left(2\epsilon + \frac{1}{2^{2n}}\right) ,$$

for some adversary $\mathcal{A}'$ with a total query complexity $q' = 2q$ and a total data complexity $\sigma' = \sigma$ bits.

We remark that $\mathcal{A}'$ in fact makes q queries whose output is of size n bits, and q queries whose output is of arbitrary size but that add up to $\sigma - qn$ bits. We furthermore highlight the fact that the result of Gunsing et al. guarantees beyond birthday bound security in case (i) the number of reuses per tweak is limited, and (ii) F achieves beyond birthday bound security at least as long as the number of reuses per tweak is limited.

4 Specification of *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES*

We will describe how we suggest to instantiate *ddd* using AES to obtain a birthday bound secure *ddd-AES* and *ddd-AES*$^+$, and a beyond birthday bound secure *bbb-ddd-AES* (if the number of tweak reuses is limited). For both of them, we suggest the same instantiation of H, as described in Sec. 4.1. The main bottleneck, however, will be the design of F, which gets an input of size $n+4+w$ bits and should operate on top of AES with a block size of $n = 128$ bits. We will assume that $4+w \leq n$. The instantiation of F for *ddd-AES*, including rationale, is given in Sec. 4.2. The extension of this construction to *ddd-AES*$^+$ is given and explained in Sec. 4.3. The instantiation of F for *bbb-ddd-AES*, again including rationale, is given in Sec. 4.4.

4.1 Instantiation of *H*

Due to the addition of carry-less multiplication instructions on modern CPUs, instances for H_L based on polynomial evaluation are a viable option. Hence, we decided to instantiate H_L using *Polyval* [22]. On input of a key L and a list of s field elements I_i, all elements of $GF(2^{128})[x]/(x^{128} + x^{127} + x^{126} + x^{121} + 1)$, it is defined as

$$Polyval_L(I_1, I_2, \ldots, I_s) = \sum_{i=1}^{s} \left(L^{s-i+1} \cdot I_i \cdot x^{-128\cdot(s-i+1)} \right) , \tag{7}$$

We will use it for arbitrary-length bit strings, always of length at most $\ell_{\max} - n$ bits. To process such string using $Polyval_L$, it is first 0-padded to the nearest multiple of n bits. Then, an n-bit string encoding the bit length of I is appended. The resulting bit string then represents $I_1 \| I_2 \| \cdots \| I_s$, noting that we can uniquely map elements from this field to bit strings in $\{0,1\}^{128}$. Particularly, in our case, $s \leq m_{\max}$, and for this case, *Polyval* is an ϵ-XOR-universal hash function with $\epsilon = m_{\max}/2^n$ [22, Lemma 3].

4.2 Instantiation of *F* for *ddd-AES*

We realize F by turning the AES-128 blockcipher E_K into an *XE*-style [50] tweakable blockcipher, where B and W function as tweak, and plugging this tweakable blockcipher into counter mode to obtain a keystream of arbitrary length. Note that the *XE*-style is sufficient as opposed to the *XEX*-style, as the primitive is never evaluated in inverse direction.

In detail, we define $F_K : \{0,1\}^{n+4+w} \to \{0,1\}^*$ as

$$F_K(I, B, W) = \left\lfloor E_K(I \oplus 2^0 S) \| E_K(I \oplus 2^1 S) \| \ldots \| E_K(I \oplus 2^{m_{\max}-1} S) \right\rfloor_{\ell_{\max}} , \tag{8}$$

where $S = E_K(B \| W)$ serves as tweak-dependent subkey. In this case, we can support a tweak with a length of $w = 124$ bits. The keylength κ depends on the actual instance chosen for AES [16, 17].

4.3 Instantiation of F for *ddd-AES*$^+$

To realize a function F that can achieve a similar level of security as that of Sec. 4.2 but that can instead accommodate arbitrary-length tweaks, we again turn the AES-128 blockcipher E_K into an *XE*-style [50] tweakable blockcipher, where B and W function as tweak. The difference is only in the subkey S: instead of $S = E_K(B\|W)$ as in Sec. 4.2, we concatenate $W\|B'$, with $B' = B \oplus 1000$, pad it with 0s into w-bit blocks $W_0, W_1, \dots, W_{l-1}\|B'\|0^*$, and define

$$S = E_K(W_0\|0) \oplus E_K(W_1\|1) \oplus \dots \oplus E_K(W_{l-1}\|B'\|0^*\|(l-1))\,, \tag{9}$$

where $0, \dots, l-1$ function as an $(n-w)$-bit counter. Here, it is important to note that this padding is injective, as in *ddd* we have $B \in \{0001, 0010\}$. In case of AES, where $n = 128$, we suggest to use 32 bits for the counter. We can then support a tweak with a length of at most $2^{32} \cdot 96 - 4$ bits, which we assume to be at least $\ell_{\max}$. The keylength κ depends on the actual instance chosen for AES [16,17].

4.4 Instantiation of F for *bbb-ddd-AES*

To realize a function F that achieves beyond birthday bound security (in case of limited tweak reuse, cf., comment below Theorem 2), we extend the $XORP[v]$ [31] that underlies CENC [30] to include a tweak.

Our tweak inclusion will be similar to the *XE*-style approach, albeit with counter included in the subkey. In detail, we assume F to have two keys instead of one, $K = K_1\|K_2 \in \{0,1\}^{2\kappa}$, and we consider the following approach for the subkey computation:

$$S_j = E_{K_2}(B\|W\|c\|j)\,, \tag{10}$$

where j will function as "inner counter" in the evaluation of F and c as "outer counter" for the mode employing F.

We subsequently define $\widetilde{XORP}[v]$ for $v \in \mathbb{N}$ on top of a blockcipher $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ as

$$\begin{aligned}\widetilde{XORP}[v]^E_K(I,B,W,c) = &\big(E_{K_1}(I \oplus S_0) \oplus E_{K_1}(I \oplus S_1)\big)\| \dots \\ &\dots \|\big(E_{K_1}(I \oplus S_0) \oplus E_{K_1}(I \oplus S_v)\big)\,.\end{aligned} \tag{11}$$

This construction is depicted in Fig. 2. This approach leaves us with $n-4$ bits that can be distributed between the outer counter c, the inner counter j, and the tweak W. In case of AES, where $n = 128$, we suggest to use 28 bits split between the counters c and j, where j occupies $\lceil \log_2(v+1)\rceil \le 28$ bits and c gets $28 - \lceil \log_2(v+1)\rceil$ bits of space. This leaves room for a $(w = 96)$-bit tweak.

We finally define $F_K : \{0,1\}^{n+4+w} \to \{0,1\}^*$ as counter mode on top of $\widetilde{XORP}[v]$ truncated to the required length:

$$F_K(I,B,W) = \Big\lfloor \widetilde{XORP}[v]^E_K(I,B,W,0)\| \dots \|\widetilde{XORP}[v]^E_K(I,B,W,\lceil m_{\max}/v\rceil)\Big\rfloor_{\ell_{\max}}\,. \tag{12}$$

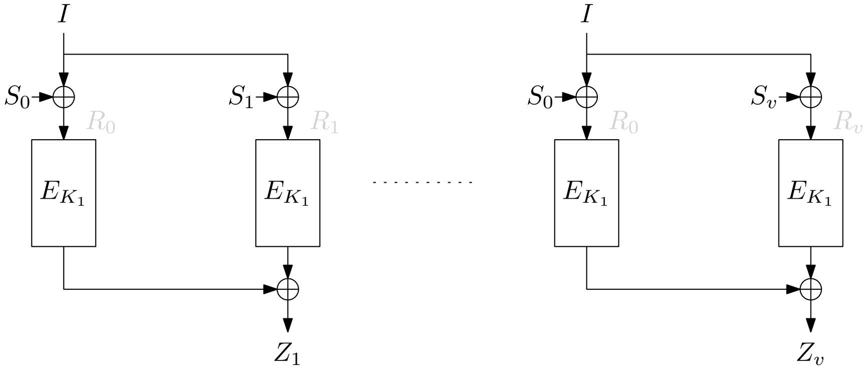


Fig. 2: The $\widetilde{XORP}[v]$ construction. Here, $S_j = E_{K_2}(B\|W\|c\|j)$ of (10). The parameters R_j will be used of the proof of Theorem 3 in Sec. 7.

5 Security of *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES*

We will discuss the security of *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES* in the security model of Sec. 2.1. The security analyses of all three functions have in common that they rely on the XOR-universality of H, which is already briefly stated in Sec. 4.1, but which we formally repeat here for convenience.

Lemma 1 (Gueron et al. [22, Lemma 3]). *The universal hash function Polyval of (7) is ϵ-XOR-universal with $\epsilon = m_{\max}/2^n$.*

Security of *ddd-AES* is now treated in Sec. 5.1, security of *ddd-AES*$^+$ in Sec. 5.2, and security of *bbb-ddd-AES* in Sec. 5.3.

5.1 Security of *ddd-AES*

The *ddd-AES* scheme is based on the *XE* construction that operates on a blockcipher $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$:

$$XE_K^E(I, B, W, j) = E_K(I \oplus 2^j E_K(B\|W)) \,. \tag{13}$$

Rogaway [50] proved that this *XE* construction[5] behaves like a random tweakable permutation as long as the total number of evaluations q satisfies $4.5q^2/2^n$ and as long as E is PRP-secure after at most q queries. However, we will rather use the *XE* construction as a PRF, and looking at the proof of [50, Theorem 1], which can be found in the full version [51, Appendix B], it *first* proves *XE* to be PRF-secure and then as last step makes an RF-to-(T)RP switch at the cost of $0.5q^2/2^n$. We will require PRF-security of the *XE* construction, thus allowing us to use a slightly tighter bound.

[5] A small change is in the split of the nonce into B and W, and in the fact that the subkey $E_K(B\|W)$ is multiplied only by 2^j.

Lemma 2 (Rogaway [50, Theorem 1]). *Consider the construction XE of (13) on top of a pseudorandom permutation* $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$. *For any adversary* $\mathcal{A}$ *making at most* q *queries, each of output size* n *bits, we have*

$$\mathbf{Adv}^{\mathrm{prf}}_{XE}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') + \frac{4q^2}{2^n} \,,$$

for some adversary $\mathcal{A}'$ *with a total query complexity* $q' = 2q$.

Looking ahead, *XE* is a simplification of XE^+ of Section 5.2 and the proof of Lemma 3 carries over, although it is slightly worse due to generality.

The security of *ddd-AES* is now a direct corollary of Theorem 2, Lemma 1, and Lemma 2, the only work actually being the data complexity translation from bits queried in *ddd-AES* to actual evaluations of the underlying *AES*. To be precise, in *ddd* the underlying F is evaluated $2q$ times with a total output data complexity of σ bits. These amount to at most $\lceil \sigma/n \rceil$ evaluations of *XE* of (13).

Corollary 1. *Consider ddd-AES, the docked double decker construction ddd on top of Polyval* $: \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^n$ *and AES* $: \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ *through XE of (13). For any adversary* $\mathcal{A}$ *making at most* q *queries, each of size at least* $2n$ *and at most* $\ell_{\max}$ *bits, and in total of size at most* σ *bits, we have*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{stwprp}}_{ddd\text{-}AES}(\mathcal{A}) \leq{}& \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') + \frac{4(\lceil \sigma/n \rceil)^2}{2^n} \\ &+ \sum_{W \in \{0,1\}^w} \binom{q_W}{2} \cdot \left(\frac{2m_{\max}}{2^n} + \frac{1}{2^{2n}} \right) ,\end{aligned}$$

for some adversary $\mathcal{A}'$ *with a total query complexity* $q' = 2\lceil \sigma/n \rceil$, *and where* q_W *is the number of queries made for tweak* $W \in \{0,1\}^w$.

5.2 Security of *ddd-AES*$^+$

The *ddd-AES*$^+$ scheme is based on the *XE* construction that operates on a blockcipher $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, but using a different and more involved mask. Let us call this construction XE^+:

$$XE^{+E}_{K}(I, B, W, j) = E_K(I \oplus 2^j S) \,, \tag{14}$$

with S of (9). As this subkey is more involved, we cannot directly rely on the result of Rogaway (Lemma 2). Nevertheless, we can derive a comparable bound:

Lemma 3. *Consider the construction* XE^+ *of (14) on top of a pseudorandom permutation* $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$. *For any adversary* $\mathcal{A}$ *making at most* q *queries, each of output size* n *bits, in total of tweak input size* σ *bits, and where the total number of padded tweak blocks is at most* ρ, *we have*

$$\mathbf{Adv}^{\mathrm{prf}}_{XE^+}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') + \frac{3\rho q + 3\binom{q}{2}}{2^n} \,,$$

for some adversary $\mathcal{A}'$ *with a total query complexity* $q' = \rho + q$.

The proof of Lemma 3 is a bit technical, and is given in Sec. 6. Note that $\rho \geq q$. In case we would restrict to a masking consisting of one blockcipher call, we would have $\rho = q$, and in this case, Lemma 3 is only marginally worse than Lemma 2 (due to generality of the proof).

The security of *ddd-AES*$^+$ is now a direct corollary of Theorem 2, Lemma 1, and Lemma 3, identical to the reduction in Sec. 5.1.

Corollary 2. *Consider ddd-AES$^+$, the docked double decker construction ddd on top of Polyval : $\{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^n$ and AES : $\{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ through XE$^+$ of (14). For any adversary $\mathcal{A}$ making at most q queries, each of size at least $2n$ and at most $\ell_{\max}$ bits, in total of size at most σ bits, and where the total number of padded tweak blocks is at most ρ, we have*

$$\mathbf{Adv}^{\text{stwprp}}_{ddd\text{-}AES^+}(\mathcal{A}) \leq \mathbf{Adv}^{\text{prp}}_{E}(\mathcal{A}') + \frac{3\rho(\lceil\sigma/n\rceil) + 3\binom{\lceil\sigma/n\rceil}{2}}{2^n} + \sum_{W\in\{0,1\}^w} \binom{q_W}{2} \cdot \left(\frac{2m_{\max}}{2^n} + \frac{1}{2^{2n}}\right),$$

for some adversary $\mathcal{A}'$ with a total query complexity $q' = \rho + \lceil\sigma/n\rceil$, and where q_W is the number of queries made for tweak $W \in \{0,1\}^w$.

Note that here, $\rho \geq \lceil\sigma/n\rceil$ by definition.

5.3 Security of *bbb-ddd-AES*

We will consider the security of the *bbb-ddd-AES* scheme. However, this analysis is not as simple as that of *ddd-AES* of Sec. 5.1. The reason for this is that *bbb-ddd-AES* is based on a new pseudorandom function design, namely $\widetilde{XORP}[v]$ of (11). Thus, we first have to analyze the PRF-security of $\widetilde{XORP}[v]$.

Theorem 3. *Let $v \in \mathbb{N}$. Consider the construction $\widetilde{XORP}[v]$ of (11) on top of a pseudorandom permutation $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$. For any adversary $\mathcal{A}$ making at most q queries, each of output size vn bits, and where q_{BWc} is the number of queries made for tweak $B\|W\|c \in \{0,1\}^{n-\lceil\log_2(v+1)\rceil}$, we have*

$$\mathbf{Adv}^{\text{prf}}_{\widetilde{XORP}[v]}(\mathcal{A}) \leq 2\mathbf{Adv}^{\text{prp}}_{E}(\mathcal{A}') + \sum_{BWc} \binom{q_{BWc}}{2} \frac{2(v+1)^2}{2^n} + \frac{(v+1)^4 q^3}{5 \cdot 2^{2n}} + \frac{\binom{v+1}{2} q}{2^n} + \frac{2\binom{v+1}{2}^2 q^2}{2^{2n}},$$

for some adversary $\mathcal{A}'$ with a total query complexity $q' = (v+1)q$, where we assume that $n(2v+1)^2 + (2v+1) \leq 2^{n/2}$ and $(2v+1)^2(v+1)q \leq 2^n/12$.

The proof of Theorem 3 is technically involved, and is given in Sec. 7.

The security of *bbb-ddd-AES* is now a direct corollary of Theorem 2, Lemma 1, and Theorem 3, the only work actually being the data complexity translation from bits queried in *bbb-ddd-AES* to actual evaluations of the underlying *AES*.

Consider a single evaluation of *bbb-ddd-AES* on input of ℓ_i bits and $m_i = \lceil \ell_i/n \rceil$ blocks. One evaluation of F is for 1 n-bit output block: it makes 1 evaluation of $\widetilde{XORP}[v]$ that costs 2 calls to each blockcipher. One evaluation of $\widetilde{F}$ is for $m_i - 1$ n-bit output blocks: it makes $\lceil (m_i-1)/v \rceil$ evaluations of $\widetilde{XORP}[v]$ that cost at most $(v+1)\lceil (m_i-1)/v \rceil$ calls to each blockcipher. Summing over all q queries, *bbb-ddd-AES* incurs

$$\sum_{i=1}^{q} \left(\left\lceil \frac{m_i - 1}{v} \right\rceil + 1 \right) \leq \frac{1}{v} \lceil \sigma/n \rceil + \frac{v+1}{v} q =: q_x \tag{15}$$

evaluations of $\widetilde{XORP}[v]$ with a total amount of at most

$$\sum_{i=1}^{q} \left((v+1) \left\lceil \frac{m_i - 1}{v} \right\rceil + 2 \right) \leq \frac{v+1}{v} \lceil \sigma/n \rceil + \frac{3v+1}{v} q =: q_e \tag{16}$$

calls to each blockcipher, where we used that $\sum_{i=1}^{q} \ell_i \leq \sigma$ and thus $\sum_{i=1}^{q} m_i \leq \lceil \sigma/n \rceil + q$.

Finally, we remark that

$$\sum_{BWc} \binom{q_{BWc}}{2} \leq 2 \sum_{W \in \{0,1\}^w} \binom{q_W}{2} ,$$

and we will use this observation to slightly simplify the bound further.

Corollary 3. *Consider bbb-ddd-AES, the docked double decker construction ddd on top of Polyval* $: \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^n$ *and AES* $: \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ *through* $\widetilde{XORP}$ *of (11). Let* $v \in \mathbb{N}$ *and let* q_x *and* q_e *be as in (15) and (16). For any adversary* $\mathcal{A}$ *making at most* q *queries, each of size at least* $2n$ *and at most* $\ell_{\max}$ *bits (equivalent to* $m_{\max}$ n*-bit blocks), and in total of size at most* σ *bits, and where* q_W *is the number of queries made for tweak* $W \in \{0,1\}^w$*, we have*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{stwprp}}_{bbb\text{-}ddd\text{-}AES}(\mathcal{A}) \leq 2\mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') + \frac{(v+1)^4 q_x^3}{5 \cdot 2^{2n}} + \frac{\binom{v+1}{2} q_x}{2^n} + \frac{2\binom{v+1}{2}^2 q_x^2}{2^{2n}} \\ + \sum_{W \in \{0,1\}^w} \binom{q_W}{2} \cdot \left(\frac{4(v+1)^2}{2^n} + \frac{2m_{\max}}{2^n} + \frac{1}{2^{2n}} \right) ,\end{aligned}$$

for some adversary $\mathcal{A}'$ *with a total query complexity* $q' = q_e$*, where we assume that* $n(2v+1)^2 + (2v+1) \leq 2^{n/2}$ *and* $(2v+1)^2(v+1)q_x \leq 2^n/12$.

We refer to Sec. 9 for an interpretation of the bound.

6 Proof of Lemma 3

A proof overview is given in Sec. 6.1, with the definition of bad transcripts in Sec. 6.2, and probability analyses in Sec. 6.3 and Sec. 6.4. The proof is concluded in Sec. 6.5.

6.1 Proof Overview

Let $K \xleftarrow{\$} \{0,1\}^\kappa$. We consider an adversary $\mathcal{A}$ that makes q queries to either ${XE^+}_K^E$ of (14) on top of a pseudorandom permutation $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, or to a random function RF with the same domain and range of ${XE^+}_K^E$, and it aims to distinguish them:

$$\mathbf{Adv}^{\mathrm{prf}}_{XE^+}(\mathcal{A}) = \left| \mathbf{Pr}\left(\mathcal{A}^{{XE^+}_K^E} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{RF} = 1\right) \right| . \tag{17}$$

Each query induces at most l_i (padded) tweak blocks, and we have $\sum_{i=1}^{q} l_i \leq \rho$.

As a first step, we replace the blockcipher evaluation E_K by a random permutation $\pi \xleftarrow{\$} \mathrm{perm}(n)$. This serves as key in the construction, and we abuse notation and denote it by XE^+_π. We have

$$\mathbf{Adv}^{\mathrm{prf}}_{XE^+}(\mathcal{A}) \leq \left| \mathbf{Pr}\left(\mathcal{A}^{XE^+_\pi} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{RF} = 1\right) \right| + \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') , \tag{18}$$

for some adversary $\mathcal{A}'$ with a total query complexity $q' = \rho + q$.

Transcripts. The adversary $\mathcal{A}$ makes q queries to its construction (either XE^+_π or RF) and these are summarized in a transcript

$$\tau = \{(I_i, B_i, W_i, j_i, Z_i)\}_{i=1}^{q} .$$

Without loss of generality, we assume that $(I_i, B_i, W_i, j_i) \neq (I_{i'}, B_{i'}, W_{i'}, j_{i'})$ whenever $i \neq i'$. For each query, denote the padded blocks by $W_{i,k}$ for $k = 0, \ldots, l_i - 1$. (We recall that the padding is injective, cf., Sec. 4.3.)

Note that in the real world, there are additional values related to the evaluation of XE^+_π, namely

$$S_{i,k} = \pi(W_{i,k} \| k) \tag{19}$$

for $i \in \{1, \ldots, q\}$ and $k \in \{0, \ldots, l_i - 1\}$. We extend the transcript by adding those values:

$$\tau_{\mathrm{ext}} = \{(I_i, B_i, W_i, j_i, S_{i,0} \| \cdots \| S_{i,l_i-1}, Z_i)\}_{i=1}^{q} . \tag{20}$$

In the ideal world, the values $S_{i,k}$ will be *placeholder* values sampled uniformly without replacement whenever the value $W_{i,k} \| k$ is different (simply said, in the ideal world we will use a random permutation $\pi' \xleftarrow{\$} \mathrm{perm}(n)$ to draw those values $S_{i,k}$, independently of the Z_i's).

Finally, we write for $i \in \{1, \ldots, q\}$:

$$S_i = S_{i,0} \oplus \cdots \oplus S_{i,l_i-1} . \tag{21}$$

These values are implicit in the extended transcript τ_{ext}.

Meaning of Transcripts. In the real world, each transcript tuple $(I_i, B_i, W_i, j_i, S_{i,0}\| \cdots \| S_{i,l_i-1}, Z_i) \in \tau_{\text{ext}}$ basically consists of two portions.

Firstly, there are the l_i distinct evaluations of π of the form (19):

$$\begin{cases} \pi(W_{i,0}\|0) = S_{i,0}\,, \\ \vdots \\ \pi(W_{i,l_i-1}\|(l_i-1)) = S_{i,l_i-1}\,. \end{cases} \tag{22}$$

If two queries $i, i' \in \{1, \ldots, q\}$ have the same k^{th} (padded) tweak block $W_{i,k} = W_{i',k}$, their permutation evaluations coincide; otherwise they are all distinct.

Secondly, there is a single evaluation of π of the form:

$$\pi(I_i \oplus 2^{j_i} S_i) = Z_i\,. \tag{23}$$

6.2 Bad Transcripts

We will define bad events that would make the H-coefficient technique inapplicable. Intuitively, we have to assure that (i) within isolated queries, the evaluations of the form (22) and (23) do not have two evaluations with the same input and different output, or vice versa, and (ii) the same holds between any two evaluations of different queries. However, what simplifies in our case is that if two different queries have the same k^{th} (padded) tweak block, their evaluations coincide by definition, and if they are different, their outputs will also differ. In other words, the q sets of evaluations of the form (22) never conflict. We only have to deal with cross-collisions: a masking block generation within (22) that coincides with a final transformation (23). Furthermore, we have to deal with collisions among the q final transformations (23).

In detail, for the case of problems within queries, case (i) of above paragraph, the H-coefficient technique is inapplicable if a transcript in τ_{ext} satisfies the following event:

$\mathsf{BAD}^{\star}_{\mathsf{cross}}$ There exist $i \in \{1, \ldots, q\}$ and $k \in \{0, \ldots, l_i - 1\}$, such that $W_{i,k}\|k = I_i \oplus 2^{j_i} S_i$ or $S_{i,k} = Z_i$.

For the case of problems among queries, case (ii) of above paragraph, the event generalizes as follows:

$\mathsf{BAD}^{\star\star}_{\mathsf{cross}}$ There exist distinct $i, i' \in \{1, \ldots, q\}$ and $k \in \{0, \ldots, l_i - 1\}$, such that $W_{i,k}\|k = I_{i'} \oplus 2^{j_{i'}} S_{i'}$ or $S_{i,k} = Z_{i'}$.

Finally, we have the following for collisions among the evaluations of (23).

$\mathsf{BAD}^{\star\star}_{\mathsf{final}}$ There exist distinct $i, i' \in \{1, \ldots, q\}$ such that $I_i \oplus 2^{j_i} S_i = I_{i'} \oplus 2^{j_{i'}} S_{i'}$ or $Z_i = Z_{i'}$.

We write

$$\mathsf{BAD} = \mathsf{BAD}^{\star}_{\mathsf{cross}} \vee \mathsf{BAD}^{\star\star}_{\mathsf{cross}} \vee \mathsf{BAD}^{\star\star}_{\mathsf{final}}\,. \tag{24}$$

6.3 Probability of Bad Transcripts

Following Theorem 1, we have to upper bound the probability that a bad transcript occurs in the ideal world, i.e., for RF. By basic probability theory,

$$\mathbf{Pr}\left(X_{RF} \in \mathcal{T}_{\mathrm{bad}}\right) \leq \mathbf{Pr}\left(\mathsf{BAD}^{\star}_{\mathsf{cross}}\right) + \mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{cross}}\right) + \mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{final}}\right) . \tag{25}$$

We investigate the probabilities separately.

$\mathbf{Pr}\left(\mathsf{BAD}^{\star}_{\mathsf{cross}}\right)$. Consider any of the q choices for i and any of the l_i choices for k (at most ρ choices in total). From (21), we can conclude that the bad event is set if

$$S_{i,0} = 2^{-j_i}\big(W_{i,k} \| k \oplus I_i \oplus 2^{j_i} \bigoplus_{k' \neq 0} S_{i,k'}\big) \text{ or } S_{i,k} = Z_i \,.$$

The value $S_{i,0}$ is drawn uniformly randomly from a set of size at least $2^n - \rho$ elements, and Z_i from a set of size 2^n elements, and thus, above equation is satisfied with probability at most $\frac{1}{2^n-\rho} + \frac{1}{2^n}$.

In conclusion, the bad event is set with probability at most

$$\frac{\rho}{2^n - \rho} + \frac{\rho}{2^n} \leq \frac{3\rho}{2^n} \,,$$

using that $\rho \leq 2^{n-1}$ for the inequality.

$\mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{cross}}\right)$. Consider any of the $\binom{q}{2}$ choices for i, i' and any of the l_i choices for k (at most $\rho(q-1)$ choices in total). From (21), we can conclude that the bad event is set if

$$S_{i',0} = 2^{-j_{i'}}\big(W_{i,k} \| k \oplus I_{i'} \oplus 2^{j_{i'}} \bigoplus_{k' \neq 0} S_{i',k'}\big) \text{ or } S_{i,k} = Z_{i'} \,.$$

The value $S_{i',0}$ is drawn uniformly randomly from a set of size at least $2^n - \rho$ elements, and $Z_{i'}$ from a set of size 2^n elements, and thus, above equation is satisfied with probability at most $\frac{1}{2^n-\rho} + \frac{1}{2^n}$.

In conclusion, the bad event is set with probability at most

$$\frac{\rho(q-1)}{2^n - \rho} + \frac{\rho(q-1)}{2^n} \leq \frac{3\rho(q-1)}{2^n} \,,$$

using that $\rho \leq 2^{n-1}$ for the inequality.

$\mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{final}}\right)$. Consider any of the $\binom{q}{2}$ choices for i, i'. From (21), we can conclude that the bad event is set if

$$S_{i,0} = 2^{-j_i}\big(I_i \oplus I_{i'} \oplus 2^{j_{i'}} S_{i'} \oplus 2^{j_i} \bigoplus_{k' \neq 0} S_{i,k'}\big) \text{ or } Z_i = Z_{i'} \,.$$

The value $S_{i,0}$ is drawn uniformly randomly from a set of size at least $2^n - \rho$ elements, and Z_i from a set of size 2^n elements, and thus, above equation is satisfied with probability at most $\frac{1}{2^n-\rho} + \frac{1}{2^n}$.

In conclusion, the bad event is set with probability at most

$$\frac{\binom{q}{2}}{2^n - \rho} + \frac{\binom{q}{2}}{2^n} \leq \frac{3\binom{q}{2}}{2^n} \,,$$

using that $\rho \leq 2^{n-1}$ for the inequality.

Conclusion. We obtain from (25) and the individual bounds that

$$\mathbf{Pr}\left(X_{RF} \in \mathcal{T}_{\text{bad}}\right) \leq \frac{3\rho}{2^n} + \frac{3\rho(q-1)}{2^n} + \frac{3\binom{q}{2}}{2^n} = \frac{3\rho q + 3\binom{q}{2}}{2^n} \,, \tag{26}$$

provided $\rho \leq 2^{n-1}$. We set δ equal to this value.

6.4 Probability Ratio for Good Transcripts

Consider any good transcript τ_{ext}. Following Theorem 1, we have to compute a lower bound on the fraction $\frac{\mathbf{Pr}\left(X_{XE_\pi^+} = \tau_{\text{ext}}\right)}{\mathbf{Pr}(X_{RF} = \tau_{\text{ext}})}$. We will first compute the actual probabilities in the numerator and the denominator, and then combine and bound them.

For the derivation of each of the two probabilities, below, consider any good transcript $\tau_{\text{ext}} = \{(I_i, B_i, W_i, j_i, S_{i,0}\| \cdots \|S_{i,l_i-1}, Z_i)\}_{i=1}^q$. For any $W\|k \in \{0,1\}^n$, let q_{Wk} denote the number of queries where the k^{th} padded tweak block satisfies $W_{i,k} = W$. Let q' denote the number of strings $W\|k$ for which $q_{Wk} > 0$ (i.e., q' denotes the number of different padded tweak block and counter combinations).

$\mathbf{Pr}\left(X_{XE_\pi^+} = \tau_{\text{ext}}\right)$. For the computation of this probability, we have to compute the probability that $\pi \xleftarrow{\$} \text{perm}(n)$ could have resulted in the transcript. The transcript defines exactly q' input-output tuples for π through (22) and exactly q through (23), and as the transcript is good, these $q' + q$ tuples are all distinct. Thus, there are exactly $(2^n - q' - q)!$ permutations π that could have yielded this transcript.

We obtain that

$$\mathbf{Pr}\left(X_{XE_\pi^+} = \tau_{\text{ext}}\right) = \frac{(2^n - q' - q)!}{2^n!} = \frac{1}{(2^n)_{q'+q}} \,.$$

$\mathbf{Pr}\left(X_{RF} = \tau_{\text{ext}}\right)$. For the computation of this probability, we can split the transcript into the two portions, either that corresponding to (22) and that corresponding to (23). For the former, by definition, the ideal world generates values $S_{i,k}$ without replacement, and the probability that the random world yields these values is exactly $(2^n)_{q'}$. For the latter, the values Z_i are randomly generated and the probability that the random world yields these values is exactly $(2^n)^q$.

We obtain that

$$\mathbf{Pr}\left(X_{RF} = \tau_{\text{ext}}\right) = \frac{1}{(2^n)_{q'}(2^n)^q}\,.$$

Conclusion. We obtain from the individual bounds that

$$\frac{\mathbf{Pr}\left(X_{XE^+_\pi} = \tau_{\text{ext}}\right)}{\mathbf{Pr}\left(X_{RF} = \tau_{\text{ext}}\right)} = \frac{(2^n)_{q'}(2^n)^q}{(2^n)_{q'+q}} \geq 1\,. \tag{27}$$

We set $\varepsilon = 0$.

6.5 Conclusion

From the H-coefficient technique of Theorem 1, the initial steps (17) and (18) of the proof, and from the values δ obtained in (26) and ε obtained in (27), we obtain

$$\mathbf{Adv}^{\text{prf}}_{XE^+}(\mathcal{A}) \leq \frac{3\rho q + 3\binom{q}{2}}{2^n} + \mathbf{Adv}^{\text{prp}}_{E}(\mathcal{A}')\,,$$

assuming that $\rho \leq 2^{n-1}$.

7 Proof of Theorem 3

The $XORP[v]$ construction was introduced by Iwata [30] and proven to achieve $2n/3$-bit security.[6] Later, Iwata et al. [31] demonstrated that $n - \log_2(w)$ security was achieved using the mirror theory [38, 40, 43, 45, 46], and Bhattacharya and Nandi [7] proved a similar bound using the χ^2 technique [18]. Very recently, a concise proof of the mirror theory (for a very large limit on the maximum component size) was delivered [13] and the authors also applied it to $XORP[v]$. In fact, this mirror theory result considers sums of permutations, where each sum can be defined as an edge in a graph between two vertices, and where it is required that there is no circle in the graph and no too large tree. For $XORP[v]$ this is the case: each evaluation of $XORP[v]$ defines v edges over $v+1$ vertices that form a tree, basically even a star, and different evaluations of $XORP[v]$ are disconnected. Thus, $XORP[v]$ is a fairly simple application of this main mirror theory result.

It turns out that the exact same mirror theory result can *also* be used to argue security of $\widetilde{XORP}[v]$, but the application is a bit more subtle. The reason is that, in our case, again any evaluation of $\widetilde{XORP}[v]$ defines a star on v edges

[6] A variant of $XORP[v]$ based on a public permutation was introduced and analyzed by Bhattacharjee et al. [6].

over $v+1$ vertices (basically as the masking values S_j of (1) are different for $j=1,\ldots,v$) but any two different stars may collide and they may collide in $(v+1)^2$ ways. Excluding any such collision would force us into birthday bound security, but there is no need to exclude such collisions as any such collision merely implies a maximum tree size up to $2v+1$ elements. In general, as long as there is no too large tree of stars, the maximum component is still "small enough" for the mirror theory result of [13] to apply.

This will also be the main proof strategy: in a nutshell, we will demonstrate that (i) there is no too large tree of stars except with a small probability, (ii) there is no cycle of stars except with a small probability, and (iii) the mirror theory of [13] can be applied akin to the example of [13, Section 4.2], with the maximum component size roughly v times the largest tree of stars.

To do this rigorously, we first describe the mirror theory in Sec. 7.1. A proof overview is given in Sec. 7.2, with the definition of bad transcripts in Sec. 7.3, and probability analyses in Sec. 7.4 and Sec. 7.5. The proof is concluded in Sec. 7.6.

7.1 Mirror Theory

Patarin's mirror theory [38, 40, 43, 45, 46] can be used to prove close to optimal security of constructions that can be described as the sum of permutations, or bijections. We adopt the notation and result of Cogliati et al. [13], albeit in their graph representation rather than in their matrix representation.

Let $m,p\in\mathbb{N}$. Consider p *distinct* n-bit unknowns $\{X_1,\ldots,X_p\}$. A system of m difference equations over these unknowns is defined as

$$\begin{cases} X_{a_1}\oplus X_{b_1}=\lambda_1\,,\\ \vdots\\ X_{a_m}\oplus X_{b_m}=\lambda_m\,,\end{cases}\tag{28}$$

where $a_i,b_i\in\{1,\ldots,p\}$ ($a_i\neq b_i$ for all i) and $\lambda_i\in\{0,1\}^n$ for $i=1,\ldots,m$. We associate a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ to this system of equations, where the unknowns are represented by vertices $\mathcal{V}=\{X_1,\ldots,X_p\}$ and equations by edges $\mathcal{E}$, where $X_a\xleftrightarrow{\lambda}X_b$ if $(a,b,\lambda)=(a_i,b_i,\lambda_i)$ for some $i\in\{1,\ldots,m\}$.

The graph is called p.d.-consistent (pairwise distinct consistent) if there is no path whose labels λ_i sum to 0. In addition, the graph is called acyclic if it is cycle-free. Finally, for a graph $\mathcal{G}$, we define the maximum component size, i.e., the size of the largest component, by $\xi_{\max}$ vertices. The mirror theory result of Cogliati et al. [13] states the following:

Theorem 4 (Mirror theory). *Consider a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ that is p.d.-consistent, acyclic, and whose largest component is at most of size $\xi_{\max}$. As long as $n\xi_{\max}^2+\xi_{\max}\leq 2^{n/2}$ and $p\leq 2^n/(12\xi_{\max}^2)$, the number of solutions for $\mathcal{V}$ such that the equations of $\mathcal{E}$ are satisfied is at least*

$$\frac{(2^n)_p}{2^{nm}}\,.$$

7.2 Proof Overview

Let $K = K_1 \| K_2 \xleftarrow{\$} \{0,1\}^{2\kappa}$. We consider an adversary $\mathcal{A}$ that makes q queries to either $\widetilde{XORP}[v]_K^E$ of (11) on top of a pseudorandom permutation $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, or to a random function RF with the same domain and range of $\widetilde{XORP}[v]_K^E$, and it aims to distinguish them:

$$\mathbf{Adv}^{\mathrm{prf}}_{\widetilde{XORP}[v]}(\mathcal{A}) = \left| \mathbf{Pr}\left(\mathcal{A}^{\widetilde{XORP}[v]_K^E} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{RF} = 1\right) \right| . \tag{29}$$

It makes at most q_{BWc} queries per tweak $B\|W\|c \in \{0,1\}^{n-\lceil \log_2(v+1) \rceil}$.

As a first step, we replace the blockcipher evaluations E_{K_1}, E_{K_2} by random permutations $\pi_1, \pi_2 \xleftarrow{\$} \mathrm{perm}(n)$, respectively. These serve as key in the construction, and we abuse notation and denote it by $\widetilde{XORP}[v]_{\boldsymbol{\pi}}$ for $\boldsymbol{\pi} = (\pi_1, \pi_2)$. We have

$$\mathbf{Adv}^{\mathrm{prf}}_{\widetilde{XORP}[v]}(\mathcal{A}) \leq \left| \mathbf{Pr}\left(\mathcal{A}^{\widetilde{XORP}[v]_{\boldsymbol{\pi}}} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{RF} = 1\right) \right| + 2\mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') , \tag{30}$$

for some adversary $\mathcal{A}'$ with a total query complexity $q' = (v+1)q$.

Transcripts. The adversary $\mathcal{A}$ makes q queries to its construction (either $\widetilde{XORP}[v]_{\boldsymbol{\pi}}$ or RF) and these are summarized in a transcript

$$\tau = \{(I_i, B_i, W_i, c_i, Z_{i,1} \| \cdots \| Z_{i,v})\}_{i=1}^q .$$

Without loss of generality, we assume that $(I_i, B_i, W_i, c_i) \neq (I_{i'}, B_{i'}, W_{i'}, c_{i'})$ whenever $i \neq i'$.

Note that in the real world, there are additional values related to the evaluation of $\widetilde{XORP}[v]_{\boldsymbol{\pi}}$, namely

$$S_{i,j} = \pi_2(B_i \| W_i \| c_i \| j) \tag{31}$$

for $i \in \{1, \ldots, q\}$ and $j \in \{0, 1, \ldots, v\}$. We extend the transcript by adding those values:

$$\tau_{\mathrm{ext}} = \{(I_i, B_i, W_i, c_i, S_{i,0} \| \cdots \| S_{i,v}, Z_{i,1} \| \cdots \| Z_{i,v})\}_{i=1}^q . \tag{32}$$

In the ideal world, the values $S_{i,j}$ will be placeholder values sampled uniformly without replacement whenever the value $B_i \| W_i \| c_i \| j$ is different (simply said, in the ideal world we will also use π_2 to draw those values $S_{i,j}$).

Finally, we write for $i \in \{1, \ldots, q\}$ and $j \in \{0, 1, \ldots, w\}$:

$$R_{i,j} = I_i \oplus S_{i,j} . \tag{33}$$

These values are implicit in the extended transcript τ_{ext}.

Meaning of Transcripts. In the real world, each transcript tuple $(I_i, B_i, W_i, c_i, S_{i,0}\|\cdots\|S_{i,v}, Z_{i,1}\|\cdots\|Z_{i,v}) \in \tau_{\text{ext}}$ basically consists of two portions.

Firstly, there are the $v+1$ distinct evaluations of π_2 of the form (31):

$$\begin{cases} \pi_2(B_i\|W_i\|c_i\|0) = S_{i,0} \,, \\ \vdots \\ \pi_2(B_i\|W_i\|c_i\|v) = S_{i,v} \,. \end{cases} \tag{34}$$

If two queries $i, i' \in \{1, \ldots, q\}$ are made for the same tweak $B_i\|W_i\|c_i = B_{i'}\|W_{i'}\|c_{i'}$, these $v+1$ evaluations coincide; otherwise they are all distinct.

Secondly, there is the relation between the values $R_{i,j}$ (implicitly defined by the transcript as (33)) and the values $Z_{i,j}$, which corresponds to v equations over $v+1$ unknowns (note, here, the outputs of the function π_1 are regarded as unknowns):

$$\begin{cases} \pi_1(R_{i,0}) \oplus \pi_1(R_{i,1}) = Z_{i,1} \,, \\ \vdots \\ \pi_1(R_{i,0}) \oplus \pi_1(R_{i,v}) = Z_{i,v} \,. \end{cases} \tag{35}$$

In graph-speak, these form a star with v edges, as $R_{i,j} \neq R_{i,j'}$ whenever $j \neq j'$. As a matter of fact, if we were not considering $\widetilde{XORP}[v]$ but rather $XORP[v]$, the q tuples in τ_{ext} together form a forest of q stars with v edges. In the case of $\widetilde{XORP}[v]$, however, cross-star collisions may occur, turning two or more stars into a tree or even a cycle. See also the explanation in Fig. 3. We will thus define a neat ensemble of bad events to avoid cycles and too large trees.

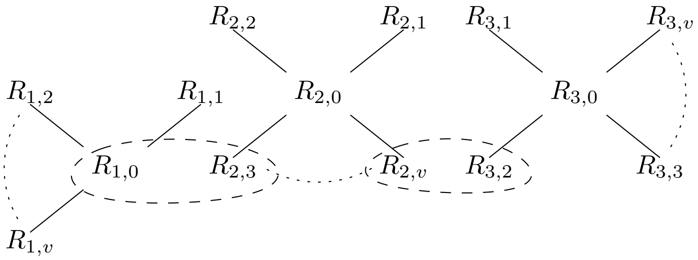


Fig. 3: Graph structure representing the evaluations of $\widetilde{XORP}[v]$. Here, each evaluation of $\widetilde{XORP}[v]$ defines a star (solid edges), but these stars may be connected to each other in case, e.g., $R_{1,0} = R_{2,3}$ and $R_{2,v} = R_{3,2}$ (dashed circle around them, meaning that they represent a single vertex).

7.3 Bad Transcripts

We will define bad events that would make the mirror theory inapplicable. Intuitively, we have to assure that (i) within stars, the system of difference equations is p.d.-consistent and acyclic, and (ii) among stars, the system of difference equations is p.d.-consistent and acyclic too. In addition, (iii) we require that there is no too large tree of stars, the reason being that any tree of μ stars basically results in a component in the graph of size exactly $\mu(v+1)-\mu+1=\mu v+1$ vertices (assuming no cycles, of course).

In detail, for the case of problems within isolated stars, case (i) of above paragraph, the mirror theory is inapplicable if a transcript in τ_{ext} satisfies one of the following events:

$\mathsf{BAD}^{\star}_{\mathsf{pdinc}}$ There exist $i \in \{1,\ldots,q\}$ and $j \in \{1,\ldots,v\}$, such that $Z_{i,j}=0^n$, or $i \in \{1,\ldots,q\}$ and distinct $j,j' \in \{1,\ldots,v\}$, such that $Z_{i,j}=Z_{i,j'}$;

$\mathsf{BAD}^{\star}_{\mathsf{cycle}}$ There exist $i \in \{1,\ldots,q\}$ and distinct $j,j' \in \{0,\ldots,v\}$, such that $R_{i,j}=R_{i,j'}$.

We note that the index sets for j,j' are *not* a typo: for $Z_{i,j}$, j,j' run from 1 to v, whereas for $R_{i,j}$, j,j' run from 0 to v. Note that any star contains paths of length 1 and length 2 only, and $\mathsf{BAD}^{\star}_{\mathsf{pdinc}}$ covers p.d.-inconsistencies over any of those paths. Event $\mathsf{BAD}^{\star}_{\mathsf{cycle}}$ will be used to excludes cycles, both of length 1 (if j or j' equals 0) and of length 2 (if both j and j' are unequal to 0).

For the case of problems among stars, case (ii) of above paragraph, these two events generalize as follows:

$\mathsf{BAD}^{\star\star}_{\mathsf{pdinc}}$ There exist $\ell \geq 2$, distinct $i_1,\ldots,i_\ell \in \{1,\ldots,q\}$, and distinct $j_\alpha,k_\alpha \in \{0,\ldots,v\}$ for each $\alpha \in \{1,\ldots,\ell\}$, such that

$$\forall_{\alpha=1}^{\ell-1} \,:\, R_{i_\alpha,j_\alpha}=R_{i_{\alpha+1},k_{\alpha+1}}\,,$$

and

$$\sum_{\alpha=1}^{\ell}\Big(Z_{i_\alpha,j_\alpha}\oplus Z_{i_\alpha,k_\alpha}\Big)=0\,,$$

where $Z_{i,0}=0^n$ for all i by definition;

$\mathsf{BAD}^{\star\star}_{\mathsf{cycle}}$ There exist $\ell \geq 2$, distinct $i_1,\ldots,i_\ell \in \{1,\ldots,q\}$, and distinct $j_\alpha,k_\alpha \in \{0,\ldots,v\}$ for each $\alpha \in \{1,\ldots,\ell\}$, such that

$$\forall_{\alpha=1}^{\ell} \,:\, R_{i_\alpha,j_\alpha}=R_{i_{\alpha+1},k_{\alpha+1}}\,,$$

where $(i_{\ell+1},k_{\ell+1})=(i_1,k_1)$ by definition.

Event $\mathsf{BAD}^{\star\star}_{\mathsf{pdinc}}$ considers the case that there is a path of ℓ distinct stars and considers all vertex paths that are included within this path of stars. Note that for any such path, for any individual inner star (so $\alpha=2,\ldots,\ell-1$) the vertex path cannot traverse freely but has to traverse from R_{i_α,j_α} to R_{i_α,k_α}, adding exactly $Z_{i_\alpha,j_\alpha}\oplus Z_{i_\alpha,k_\alpha}$ to the checksum, noting that $Z_{i,0}=0^n$ by definition. For the outer stars, so $\alpha = 1, \ell$, it may or may not traverse further to any R_{i_1,k_1} or R_{i_α,j_α} respectively, again adding exactly $Z_{i_\alpha,j_\alpha} \oplus Z_{i_\alpha,k_\alpha}$ to the checksum. Likewise, event $\mathsf{BAD}^{\star\star}_{\mathsf{cycle}}$ considers the case that there is a cycle over ℓ distinct stars. Note that for both events, the condition that $j_\alpha \neq k_\alpha$ is reasonable to make: in case of equality, there would have been a shorter path or cycle without equality at the α^{th} indices; in case of equality for all indices, both bad events would become meaningless.

Finally, there is the case of a too large tree of stars, case (iii) of above paragraph. Basically, we have to define any threshold $\mu \in \mathbb{N}$ and state the event that there is a tree that connects $\mu + 1$ stars. This is quite cumbersome to define. On the other hand, looking ahead, we will only be able to bound the probability of this event to occur for $\mu = 2$. In this case, the event is more straightforward to define (as a tree of 3 stars is necessarily a path of 3 stars):

$\mathsf{BAD}^{\star\star}_{\mathsf{tree}}$ There exist distinct $i_1, i_2, i_3 \in \{1, \ldots, q\}$ and $j_1, j_2, k_2, k_3 \in \{0, \ldots, v\}$ (with no further distinctness condition), such that

$$\begin{aligned} R_{i_1,j_1} &= R_{i_2,k_2} \,, \\ R_{i_2,j_2} &= R_{i_3,k_3} \,. \end{aligned}$$

Bad event $\mathsf{BAD}^{\star\star}_{\mathsf{tree}}$ differs from $\mathsf{BAD}^{\star\star}_{\mathsf{pdinc}}$ and $\mathsf{BAD}^{\star\star}_{\mathsf{cycle}}$ in that there is no distinctness condition on the values j_α, k_α. After all, $\mathsf{BAD}^{\star\star}_{\mathsf{tree}}$ is meant to capture, basically to upper bound, the size of the largest component in the graph. To derive this bound, all that matters is to figure out the maximum number of stars that are connected, and it is irrelevant *how* they are connected.

We write

$$\mathsf{BAD} = \mathsf{BAD}^{\star}_{\mathsf{pdinc}} \vee \mathsf{BAD}^{\star}_{\mathsf{cycle}} \vee \mathsf{BAD}^{\star\star}_{\mathsf{pdinc}} \vee \mathsf{BAD}^{\star\star}_{\mathsf{cycle}} \vee \mathsf{BAD}^{\star\star}_{\mathsf{tree}} \,. \tag{36}$$

7.4 Probability of Bad Transcripts

Following Theorem 1, we have to upper bound the probability that a bad transcript occurs in the ideal world, i.e., for RF. By basic probability theory,

$$\begin{aligned} \mathbf{Pr}\left(X_{RF} \in \mathcal{T}_{\text{bad}}\right) \leq\, &\mathbf{Pr}\left(\mathsf{BAD}^{\star}_{\mathsf{pdinc}}\right) + \mathbf{Pr}\left(\mathsf{BAD}^{\star}_{\mathsf{cycle}}\right) + \mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{tree}}\right) \\ &+ \mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{pdinc}} \mid \neg\mathsf{BAD}^{\star\star}_{\mathsf{tree}}\right) + \mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{cycle}} \mid \neg\mathsf{BAD}^{\star\star}_{\mathsf{tree}}\right) \,. \end{aligned} \tag{37}$$

We investigate the probabilities separately.

$\mathbf{Pr}\left(\mathsf{BAD}^{\star}_{\mathsf{pdinc}}\right)$. The event is set whenever $Z_{i,j} = 0$ for some i, j (vq choices) or whenever $Z_{i,j} = Z_{i,j'}$ for some i, j, j' with $j \neq j'$ ($\binom{v}{2}q$ choices). As the values $Z_{i,j}$ are uniformly randomly generated, this bad event happens with probability at most

$$\frac{(v + \binom{v}{2})q}{2^n} = \frac{\binom{v+1}{2}q}{2^n} \,.$$

(As a matter of fact, the derivation and bound are identical to that of [13, Section 4.2] with the difference that they bound $\binom{v}{2}$ to $v^2/2$.)

Pr $\left(\mathsf{BAD}^{\star}_{\mathsf{cycle}}\right)$. The event is set whenever $R_{i,j} = R_{i,j'}$ for some i, j, j' with $j \neq j'$. However, from (33), we see that this happens whenever

$$I_i \oplus S_{i,j} = I_i \oplus S_{i,j'} \,,$$

i.e., whenever $S_{i,j} = S_{i,j'}$. As in the ideal world, the placeholder values $S_{i,j}$ and $S_{i,j'}$ are drawn randomly without replacement, the event happens with probability 0.

Pr $(\mathsf{BAD}^{\star\star}_{\mathsf{tree}})$. Recall that we will perform the analysis for $\mu = 2$. Consider any of the $\binom{q}{3}$ choices for i_1, i_2, i_3 and any of the $(v+1)^4$ choices for j_1, j_2, k_2, k_3. The event is set if

$$\begin{aligned} S_{i_1,j_1} \oplus S_{i_2,k_2} &= I_{i_1} \oplus I_{i_2} \,, \\ S_{i_2,j_2} \oplus S_{i_3,k_3} &= I_{i_2} \oplus I_{i_3} \,. \end{aligned}$$

As the three queries are distinct, and the adversary never repeats queries, we necessarily have $(I_{i_1}, B_{i_1}, W_{i_1}, c_{i_1}) \neq (I_{i_2}, B_{i_2}, W_{i_2}, c_{i_2})$, which implies that the first equation can only be satisfied if $B_{i_1} \| W_{i_1} \| c_{i_1} \| j_1 \neq B_{i_2} \| W_{i_2} \| c_{i_2} \| k_2$. This means that, necessarily, $S_{i_1,j_1} \neq S_{i_2,k_2}$ and the two sources of randomness in the first equation do not cancel each other out. Likewise, the second equation can only be satisfied if $B_{i_2} \| W_{i_2} \| c_{i_2} \| j_2 \neq B_{i_3} \| W_{i_3} \| c_{i_3} \| k_3$, and the two sources of randomness S_{i_2,j_2} and S_{i_3,k_3} do not cancel each other out.

Finally, we have to argue that both equations are sufficiently independent, i.e., that neither

- $S_{i_1,j_1} = S_{i_2,j_2}$ and $S_{i_2,k_2} = S_{i_3,k_3}$, nor
- $S_{i_1,j_1} = S_{i_3,k_3}$ and $S_{i_2,k_2} = S_{i_2,j_2}$.

Suppose, to the contrary, that one of these two conditions holds. The condition particularly implies that $(B_{i_1}, W_{i_1}, c_{i_1}) = (B_{i_3}, W_{i_3}, c_{i_3})$. The condition also implies, by addition of the two equations of the event, that $I_{i_1} = I_{i_3}$. This contradicts with the condition that the queries are distinct.

Thus, there are at least three sources of randomness in the two equations (note that, if $j_2 = k_2$, $S_{i_2,k_2} = S_{i_2,j_2}$). The values $S_{i,j}$ are drawn uniformly randomly from a set of size at least $2^n - (v+1)q$ elements, and thus, the two equations are satisfied with probability at most $\left(\frac{1}{2^n-(v+1)q}\right)^2$.

In conclusion, the bad event is set with probability at most

$$\binom{q}{3}(v+1)^4 \left(\frac{1}{2^n - (v+1)q}\right)^2 \leq \frac{(v+1)^4 q^3}{5 \cdot 2^{2n}} \,,$$

using that $(v+1)q \leq 2^{n-6}$ for the inequality. (We remark that this condition is more stringent than the "usual" $\leq 2^{n-1}$, but this more stringent condition is in fact implied by a condition that we will need for the application of the mirror theory anyway.)

$\mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{pdinc}} \mid \neg\mathsf{BAD}^{\star\star}_{\mathsf{tree}}\right)$. We have to consider any $\ell \geq 2$, but w.l.o.g., $\ell \leq \mu = 2$ by negation of $\mathsf{BAD}^{\star\star}_{\mathsf{tree}}$. Consider any of the $\binom{q}{2}$ choices for i_1, i_2 and any of the $\binom{v+1}{2}^2$ choices for j_α, k_α for $\alpha = 1, 2$. The event is set if

$$\begin{aligned} S_{i_1,j_1} \oplus S_{i_2,k_2} &= I_{i_1} \oplus I_{i_2} \,, \\ Z_{i_1,j_1} \oplus Z_{i_1,k_1} \oplus Z_{i_2,j_2} \oplus Z_{i_2,k_2} &= 0 \,, \end{aligned}$$

where we recall that $Z_{i,0} = 0^n$ for all i by definition.

As in the case of $\mathsf{BAD}^{\star\star}_{\mathsf{tree}}$ above, the two sources of randomness S_{i_1,j_1} and S_{i_2,k_2} in the first equation do not cancel each other out, as the adversary never repeats queries. Thus, this equation is satisfied with probability at most $\frac{1}{2^n-(v+1)q}$. For the second equation, which is independent of the first one, note that at least one of the values j_1, k_1, j_2, k_2 is non-zero, meaning that for this index, we can rely on the random drawing of the Z-value. The equation is set with probability at most $1/2^n$.

In conclusion, the bad event is set with probability at most

$$\binom{q}{2}\binom{v+1}{2}^2 \frac{1}{2^n-(v+1)q}\frac{1}{2^n} \leq \frac{\binom{v+1}{2}^2 q^2}{2^{2n}} \,,$$

using that $(v+1)q \leq 2^{n-1}$ for the inequality.

$\mathbf{Pr}\left(\mathsf{BAD}^{\star\star}_{\mathsf{cycle}} \mid \neg\mathsf{BAD}^{\star\star}_{\mathsf{tree}}\right)$. We have to consider any $\ell \geq 2$, but w.l.o.g., $\ell \leq \mu = 2$ by negation of $\mathsf{BAD}^{\star\star}_{\mathsf{tree}}$. Consider any of the $\binom{q}{2}$ choices for i_1, i_2 and any of the $\binom{v+1}{2}^2$ choices for j_α, k_α for $\alpha = 1, 2$. The event is set if

$$\begin{aligned} S_{i_1,j_1} \oplus S_{i_2,k_2} &= I_{i_1} \oplus I_{i_2} \,, \\ S_{i_2,j_2} \oplus S_{i_1,k_1} &= I_{i_1} \oplus I_{i_2} \,. \end{aligned}$$

As in the case of $\mathsf{BAD}^{\star\star}_{\mathsf{tree}}$ above, the two sources of randomness S_{i_1,j_1} and S_{i_2,k_2} in the first equation do not cancel each other out, as the adversary never repeats queries. The same holds for S_{i_2,j_2} and S_{i_1,k_1} in the second equation.

For the rest, we make a distinction between whether the two queries i_1, i_2 are selected to have the same tweaks $(B_{i_1}, W_{i_1}, c_{i_1}) = (B_{i_2}, W_{i_2}, c_{i_2})$ or not.

- Clearly, if they have the same tweaks, then it is plausible that $S_{i_1,j_1} = S_{i_2,j_2}$ and $S_{i_2,k_2} = S_{i_1,k_1}$, which means that the two equations of the event are identical. That equation, w.l.o.g., the first one, still has two sources of randomness, which are the values $S_{i,j}$ that are drawn uniformly randomly from a set of size at least $2^n - (v+1)q$ elements. The two equations are then satisfied with probability at most $\frac{1}{2^n-(v+1)q}$. Note that in this case the choice of j_2 and k_1 is redundant, we just consider any of the $(v+1)^2$ choices for j_1, k_2;
- On the other hand, assume that the two queries i_1, i_2 have *distinct tweaks* $(B_{i_1}, W_{i_1}, c_{i_1}) \neq (B_{i_2}, W_{i_2}, c_{i_2})$. We have to argue that both equations are sufficiently independent, i.e., that neither

- $S_{i_1,j_1} = S_{i_2,j_2}$ and $S_{i_2,k_2} = S_{i_1,k_1}$, nor
- $S_{i_1,j_1} = S_{i_1,k_1}$ and $S_{i_2,k_2} = S_{i_2,j_2}$.

The first condition cannot hold by the condition that the tweaks are distinct. The second condition cannot hold in the first place as $j_1 \neq k_1$ and $j_2 \neq k_2$. Thus, there are four sources of randomness in the two equations. The values $S_{i,j}$ are drawn uniformly randomly from a set of size at least $2^n - (v+1)q$ elements, and thus, the two equations are satisfied with probability at most $\left(\frac{1}{2^n-(v+1)q}\right)^2$.

In conclusion, the bad event is set with probability at most

$$\sum_{BWc} \binom{q_{BWc}}{2} \left(\frac{(v+1)^2}{2^n - (v+1)q}\right) + \binom{q}{2}\binom{v+1}{2}^2 \left(\frac{1}{2^n - (v+1)q}\right)^2$$

$$\leq \sum_{BWc} \binom{q_{BWc}}{2} \frac{2(v+1)^2}{2^n} + \frac{\binom{v+1}{2}^2 q^2}{2^{2n}} \,,$$

using that $(v+1)q \leq 2^{n-2}$ for the inequality.

Conclusion. We obtain from (37) and the individual bounds that

$$\mathbf{Pr}\left(X_{RF} \in \mathcal{T}_{\text{bad}}\right) \leq \sum_{BWc} \binom{q_{BWc}}{2} \frac{2(v+1)^2}{2^n} + \frac{(v+1)^4 q^3}{5 \cdot 2^{2n}} + \frac{\binom{v+1}{2} q}{2^n} + \frac{2\binom{v+1}{2}^2 q^2}{2^{2n}} \,, \tag{38}$$

provided $(v+1)q \leq 2^{n-6}$. We set δ equal to this value.

7.5 Probability Ratio for Good Transcripts

Consider any good transcript τ_{ext}. Following Theorem 1, we have to compute a lower bound on the fraction $\frac{\mathbf{Pr}\left(X_{\widetilde{XORP}[v]_{\boldsymbol{\pi}}} = \tau_{\text{ext}}\right)}{\mathbf{Pr}\left(X_{RF} = \tau_{\text{ext}}\right)}$. We will actually derive a lower bound on the probability in the numerator and the actual value for the probability in the denominator, and then combine them.

For the derivation of each of the two probabilities, below, consider any good transcript $\tau_{\text{ext}} = \{(I_i, B_i, W_i, c_i, S_{i,0}\| \cdots \|S_{i,v}, Z_{i,1}\| \cdots \|Z_{i,v})\}_{i=1}^q$. For any $B\|W\|c \in \{0,1\}^{n-\lceil \log_2(v+1) \rceil}$, let q_{BWc} denote the number of tuples in τ_{ext} such that $B_i\|W_i\|c_i = B\|W\|c$. Let q' denote the number of strings $B\|W\|c$ for which $q_{BWc} > 0$ (i.e., q' denotes the number of different domain separator and tweak combinations).

$\mathbf{Pr}\left(X_{\widetilde{XORP}[v]_{\boldsymbol{\pi}}} = \tau_{\text{ext}}\right)$. For the computation of this probability, we have to compute the probability that $\boldsymbol{\pi} = (\pi_1, \pi_2) \xleftarrow{\$} \text{perm}(n)^2$ could have resulted in the transcript. The transcript consists of two portions, namely

$$\tau_{\text{ext}}^2 = \{(B_i, W_i, c_i, S_{i,0}\| \cdots \|S_{i,v})\}_{i=1}^q$$

corresponding to the evaluation of π_2, and

$$\tau_{\text{ext}}^1 = \{(R_{i,0}\| \cdots \|R_{i,v}, Z_{i,1}\| \cdots \|Z_{i,v})\}_{i=1}^q$$

corresponding to the evaluation of π_1, where we recall that $R_{i,j}$ of (33) is implicit in the transcript. As for τ_{ext}^2, this sub-transcript defines exactly vq' input-output tuples for π_2, namely (34) for all q' different domain separator and tweak combinations that occur in the transcript. There are exactly $(2^n - vq')!$ permutations π_2 that could have yielded this sub-transcript. As for τ_{ext}^1, as the transcript is good, this sub-transcript defines a graph on $m := vq$ equations and $p \leq (v+1)q$ unknowns (we do not need an exact value of p) that is p.d.-consistent, acyclic, and whose largest component is of size $\xi_{\max} := \mu v + 1 = 2v + 1$. We can thus apply Theorem 4 and obtain that, provided $n\xi_{\max}^2 + \xi_{\max} \leq 2^{n/2}$ and $p \leq 2^n/(12\xi_{\max}^2)$, there are at least

$$\frac{(2^n)_p}{2^{nvq}}$$

solutions to the p unknowns. For any of these solutions, we have exactly $(2^n - p)!$ permutations π_1 that could have yielded any of these solutions.

We obtain that

$$\mathbf{Pr}\left(X_{\widetilde{XORP[v]}_{\boldsymbol{\pi}}} = \tau_{\text{ext}}\right) \geq \frac{\frac{(2^n)_p}{2^{nvq}}(2^n - vq')!(2^n - p)!}{2^n!2^n!} = \frac{1}{(2^n)_{vq'}(2^n)^{vq}}\,.$$

$\mathbf{Pr}\left(X_{RF} = \tau_{\text{ext}}\right)$. For the computation of this probability, we can likewise split the transcript into the two portions τ_{ext}^2 and τ_{ext}^1, with the difference that, now, τ_{ext}^2 is generated by randomly selecting placeholder variables $S_{i,j}$ and τ_{ext}^1 is generated through RF. The probability that the random world yields τ_{ext}^2 equals $(2^n)_{vq'}$ by definition of how the placeholder values $S_{i,j}$ are generated, and the probability that RF yields τ_{ext}^1 equals $1/(2^n)^{vq}$.

We obtain that

$$\mathbf{Pr}\left(X_{RF} = \tau_{\text{ext}}\right) = \frac{1}{(2^n)_{vq'}(2^n)^{vq}}\,.$$

Conclusion. We obtain from the individual bounds that

$$\frac{\mathbf{Pr}\left(X_{\widetilde{XORP[v]}_{\boldsymbol{\pi}}} = \tau_{\text{ext}}\right)}{\mathbf{Pr}\left(X_{RF} = \tau_{\text{ext}}\right)} \geq \frac{\frac{1}{(2^n)_{vq'}(2^n)^{vq}}}{\frac{1}{(2^n)_{vq'}(2^n)^{vq}}} = 1\,. \tag{39}$$

We set $\varepsilon = 0$.

7.6 Conclusion

From the H-coefficient technique of Theorem 1, the initial steps (29) and (30) of the proof, and from the values δ obtained in (38) and ε obtained in (39), we obtain

$$\mathbf{Adv}^{\mathrm{prf}}_{\widetilde{XORP}[v]}(\mathcal{A}) \leq$$

$$\sum_{BWc} \binom{q_{BWc}}{2} \frac{2(v+1)^2}{2^n} + \frac{(v+1)^4 q^3}{5 \cdot 2^{2n}} + \frac{\binom{v+1}{2} q}{2^n} + \frac{2\binom{v+1}{2}^2 q^2}{2^{2n}} + 2\mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}'),$$

assuming that $(v+1)q \leq 2^{n-6}$, and $n\xi_{\max}^2 + \xi_{\max} \leq 2^{n/2}$ and $(v+1)q \leq 2^n/(12\xi_{\max}^2)$ for $\xi_{\max} := 2v+1$. These three conditions simplify to $n(2v+1)^2 + (2v+1) \leq 2^{n/2}$ and $(2v+1)^2(v+1)q \leq 2^n/12$.

8 Application to Authenticated Encryption

The *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES* tweakable wide blockciphers are in essence "just" blockciphers. Given that they are wide blockciphers, one can use them to achieve confidentiality of data. For authenticity, however, some more work needs to be done. At first sight, the most logical solution would be the approach described by Hoang et al. [28], where one appends τ zeros to the plaintext P, encrypts the entire message using *ddd-AES*, *ddd-AES*$^+$, or *bbb-ddd-AES*, and outputs the resulting $(|P| + \tau)$-bit result as ciphertext-tag combination. Upon decryption, it is first checked whether the plaintext contains τ trailing zeros before the plaintext is released. This approach works well with *ddd-AES*$^+$, as this design allows for arbitrary length tweaks that could be used to compress the nonce and associated data. On the other hand, *ddd-AES* and *bbb-ddd-AES* only allow for tweaks that are too small to accommodate associated data in addition to a nonce: up to 124 bits in the case of *ddd-AES* (see Sec. 4.2) and up to 96 bits in the case of *bbb-ddd-AES* (see Sec. 4.4). Thus, we will introduce an alternative mode of use for authenticated encryption with associated data that is specifically tailored to work well on *ddd-AES* and *bbb-ddd-AES*. We dub this mode *aaa* (advanced authenticated encryption with associated data).

We first recall the security model for authenticated encryption in Sec. 8.1. The *aaa* mode is specified in Sec. 8.2. We state security of the *aaa* mode and discuss instantiation with *ddd-AES* or *bbb-ddd-AES* in Sec. 8.3. The security proof is given in Sec. 8.4. We finally present an interpretation of our bounds in Sec. 9.

8.1 Security Model

An authenticated encryption scheme AE consists of a pair of functions (enc, dec): the encryption function enc gets as input a key $K \in \{0,1\}^\kappa$, a nonce $N \in \{0,1\}^\nu$, associated data $A \in \{0,1\}^*$, and plaintext $P \in \{0,1\}^*$, and it outputs a ciphertext $C \in \{0,1\}^*$ of the same size as P and a tag $T \in \{0,1\}^\tau$. We write $enc_K(\cdot,\cdot,\cdot) = enc(K,\cdot,\cdot,\cdot)$. The decryption function dec gets as input a key $K \in \{0,1\}^\kappa$, a nonce $N \in \{0,1\}^\nu$, associated data $A \in \{0,1\}^*$, ciphertext $C \in \{0,1\}^*$, and a tag $T \in \{0,1\}^\tau$, and it outputs either a plaintext $P \in \{0,1\}^*$

of the same size as C or a dedicated $\perp$ symbol if verification fails. We write $dec_K(\cdot,\cdot,\cdot,\cdot) = dec(K,\cdot,\cdot,\cdot,\cdot)$. We require completeness, in the sense that

$$dec_K(N, A, enc_K(N, A, P)) = P\,.$$

Slightly extending earlier definition, define by $\mathrm{func}'(\nu + * + *, * + \tau)$ the family of all functions from $\{0,1\}^\nu \times \{0,1\}^* \times \{0,1\}^*$ to $\{0,1\}^* \times \{0,1\}^\tau$ that are restricted to output values whose length equals the size of their third input plus τ. The security of an authenticated encryption scheme $AE = (enc, dec)$ is defined by how hard it is for an adversary $\mathcal{A}$ to distinguish (enc_K, dec_K) for a random and secret key $K \xleftarrow{\$} \{0,1\}^\kappa$ from $(\$, \perp)$, where $\$ \xleftarrow{\$} \mathrm{func}'(\nu+*+*, *+\tau)$ and where $\perp$ is a dedicated function that always returns the $\perp$ symbol:

$$\mathbf{Adv}^{\mathrm{ae}}_{AE}(\mathcal{A}) = \left|\mathbf{Pr}\left(\mathcal{A}^{enc_K, dec_K} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\$,\perp} = 1\right)\right|\,, \qquad (40)$$

where the probabilities are taken over $K \xleftarrow{\$} \{0,1\}^\kappa$, $\$ \xleftarrow{\$} \mathrm{func}'(\nu + * + *, * + \tau)$ (lazily-sampled), and the random coins of $\mathcal{A}$.

The adversary is not allowed to make a decryption query using the result of an earlier encryption query. In addition, we call $\mathcal{A}$ *nonce-respecting* if every encryption query is made for a nonce N that is different from all nonces used in earlier encryption queries under the same key. We call $\mathcal{A}$ *nonce-misusing* if it may reuse nonces for encryption queries. We call $\mathcal{A}$ *nonce-randomizing* if every encryption query is made for a random nonce N. Note that $\mathcal{A}$ may always freely choose the nonce in decryption queries. The adversary is typically bounded by a certain number of encryption queries q_e and decryption queries q_d, and a total data complexity σ that counts the total amount of associated data bits plus plaintext/ciphertext bits.

8.2 Specification of *aaa*

Let $\kappa, \kappa', w, n, \ell_{\max}, \nu, \tau \in \mathbb{N}$ such that $2n \leq \ell_{\max}$ and $\nu \geq w$. Let $TWBC : \{0,1\}^\kappa \times \{0,1\}^w \times \mathcal{S} \to \mathcal{S}$ be a tweakable wide blockcipher operating on $\mathcal{S}$ of (1). Let $J : \{0,1\}^{\kappa'} \times \{0,1\}^* \to \{0,1\}^\tau$ be a universal hash function family. The *aaa* authenticated encryption mode is defined by the following functions (enc, dec). Encryption *enc* operates as follows:

$$enc^{TWBC,J}_{K,L}(N, A, P) = TWBC_K\Big(\mathrm{left}_w(N), J_L\big(\mathrm{right}_{\nu-w}(N)\|A\big)\|P\Big)\,, \qquad (41)$$

parsed into $C\|T$ where $|C| = |P|$ and $|T| = \tau$. The scheme is depicted in Fig. 4. Decryption *dec* first computes $S\|P = TWBC^{-1}_K(\mathrm{left}_w(N), C\|T)$, where $|S| = \tau$ and $|P| = |C|$, and is then defined as

$$dec^{TWBC,J}_{K,L}(N, A, C, T) = \begin{cases} P\,, & \text{if } S = J_L\big(\mathrm{right}_{\nu-w}(N)\|A\big)\,, \\ \perp\,, & \text{otherwise}\,. \end{cases} \qquad (42)$$

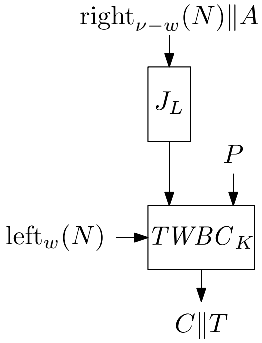


Fig. 4: The advanced authenticated encryption with associated data construction.

8.3 Security of *aaa-ddd-AES* and *aaa-bbb-ddd-AES*

We will prove security of *aaa* as an authenticated encryption mode, against nonce-respecting, nonce-misusing, and nonce-randomizing adversaries.

Theorem 5. *Consider the advanced authenticated encryption with associated data construction aaa on top of a tweakable wide blockcipher* $TWBC : \{0,1\}^\kappa \times \{0,1\}^w \times \mathcal{S} \to \mathcal{S}$ *and an* ϵ*-universal hash function family* $J : \{0,1\}^{\kappa'} \times \{0,1\}^* \to \{0,1\}^\tau$*. For any adversary* $\mathcal{A}$ *making at most* q_e *encryption queries and at most* q_d *decryption queries, with* $q = q_d + q_e$*, where each query has a nonce and associated data of size at most* $\ell_{\max}$ *bits, and plaintext of size at least* $2n-\tau$ *and at most* $\ell_{\max} - \tau$ *bits, and in total of size at most* σ *bits, we have the following result:*

- *If* $\mathcal{A}$ *is nonce-respecting on the first* w *bits of the* ν*-bit nonce, then*
$$\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') + q_d \max\left\{\epsilon, \frac{1}{2^\tau - 1}\right\} ;$$
- *If* $\mathcal{A}$ *is nonce-misusing, then*
$$\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') + \binom{q_e}{2}\left(\epsilon + \frac{1}{2^{2n}}\right) + q_d \max\left\{\epsilon, \frac{1}{2^\tau - q_e}\right\} ;$$
- *If* $\mathcal{A}$ *is nonce-randomizing on the entire* ν*-bit nonce, then*
$$\begin{aligned}\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \le{}& \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') \\ &+ 3q_e\left(\epsilon + \frac{1}{2^{2n}}\right) + q_d \max\left\{\epsilon, \frac{1}{2^\tau - 7}\right\} + \binom{q_e}{8}\frac{1}{2^{7w}} ,\end{aligned}$$

for some adversary $\mathcal{A}'$ *with a total query complexity* $q' = q_e + q_d$*, where each query is of size at least* $2n$ *and at most* $\ell_{\max}$ *bits, and total data complexity* $\sigma' = \sigma + q\tau$*.*

The proof of Theorem 5 is given in Sec. 8.4 and an interpretation of the bound can be found in Sec. 9.

To get a security bound for aaa-ddd-AES and aaa-bbb-ddd-AES, we simply have to plug the bounds of Corollary 1 or Corollary 3 respectively into the $\mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}')$ term in Theorem 5.

Remark 1. Please note that all needed different keys could be derived from a single key by using the underlying blockcipher in, e.g., the sum of permutations [2,3,18] or the summation truncation hybrid [24].

8.4 Proof of Theorem 5

Let $K \xleftarrow{\$} \{0,1\}^{\kappa}$ and $L \xleftarrow{\$} \{0,1\}^{\kappa'}$. Consider a tweakable wide blockcipher $TWBC : \{0,1\}^{\kappa} \times \{0,1\}^{w} \times \mathcal{S} \to \mathcal{S}$ and a universal hash function family $J : \{0,1\}^{\kappa'} \times \{0,1\}^{*} \to \{0,1\}^{\tau}$. We consider an adversary $\mathcal{A}$ that makes q_e queries to either $enc^{TWBC,J}_{K,L}$ of (41) or to a random function \$ that outputs a random string of appropriate length for each query, and that makes q_d queries to either $dec^{TWBC,J}_{K,L}$ of (42) or $\perp$, and it aims to distinguish them:

$$\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) = \left|\mathbf{Pr}\left(\mathcal{A}^{enc^{TWBC,J}_{K,L},dec^{TWBC,J}_{K,L}} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\$,\perp} = 1\right)\right| . \tag{43}$$

As a first step, we replace the tweakable wide blockcipher evaluations $TWBC_K$ by tweakable wide random permutation $TWRP \xleftarrow{\$} \mathrm{perm}(w, 2n : \ell_{\max})$. This serves as key in the construction, and we abuse notation and denote it by $enc^{J}_{TWRP,L}$ and $dec^{J}_{TWRP,L}$. We have

$$\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \leq \left|\mathbf{Pr}\left(\mathcal{A}^{enc^{J}_{TWRP,L},dec^{J}_{TWRP,L}} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\$,\perp} = 1\right)\right| + \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') , \tag{44}$$

for some adversary $\mathcal{A}'$ with a total query complexity $q' = q_e + q_d$, where each query is of size at least $2n$ and at most $\ell_{\max}$ bits, and total data complexity $\sigma' = \sigma + q\tau$.

We will apply the triangle inequality on the remaining difference with intermediate world "$enc^{J}_{TWRP,L}, \perp$", which gives

$$\begin{aligned}\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \leq & \left|\mathbf{Pr}\left(\mathcal{A}^{enc^{J}_{TWRP,L},dec^{J}_{TWRP,L}} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{enc^{J}_{TWRP,L},\perp} = 1\right)\right| \\ & + \left|\mathbf{Pr}\left(\mathcal{A}^{enc^{J}_{TWRP,L},\perp} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\$,\perp} = 1\right)\right| + \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') .\end{aligned} \tag{45}$$

The first difference in (45) in fact corresponds to the authenticity of aaa (noting that $\mathcal{A}$ can only distinguish by forging $dec^{J}_{TWRP,L}$), whereas the second difference in (45) corresponds to the confidentiality of aaa (noting that the oracle $\perp$ is pointless in both worlds). We now make a distinction between nonce-respecting and nonce-randomizing adversaries. We note that the nonce-misuse setting is implicit in the analysis of nonce-randomizing adversaries.

Nonce-Respecting Setting. In this case, the adversary $\mathcal{A}$ takes a unique nonce $\text{left}_w(N)$ for each encryption query (it has free choice of the rest of N, which thus basically serves as additional associated data).

For the confidentiality of *aaa*, this means that each evaluation of $TWRP$ within $enc^J_{TWRP,L}$ is made for a different tweak, results in a uniform random reply, and thus

$$\left|\mathbf{Pr}\left(\mathcal{A}^{enc^J_{TWRP,L},\perp}=1\right)-\mathbf{Pr}\left(\mathcal{A}^{\$,\perp}=1\right)\right|=0\,.$$

For the authenticity of *aaa*, consider any forgery attempt (N, A, C, T). The forgery attempt allows $\mathcal{A}$ to distinguish both worlds if

$$J_L(\text{right}_{\nu-w}(N)\|A)=\text{left}_\tau(TWRP^{-1}(\text{left}_w(N),C\|T))\,. \tag{46}$$

Denote all earlier encryption queries (w.l.o.g., q_e of them) by $\{(N_i, A_i, P_i, C_i, T_i)\}_{i=1}^{q_e}$. We distinguish among the following cases:

- For all $i \in \{1, \ldots, q_e\}$, $\text{left}_w(N) \neq \text{left}_w(N_i)$. In this case, the evaluation of $TWRP^{-1}$ within $dec^J_{TWRP,L}$ is done for a new tweak, and thus it generates a uniform random string. The condition (46) is set with probability $1/2^\tau$;
- There exists $i \in \{1, \ldots, q_e\}$, such that $\text{left}_w(N) = \text{left}_w(N_i)$. As the adversary is nonce-respecting, this value i is unique. We make a further distinction:
  - $|C\|T| \neq |C_i\|T_i|$. In this case, the evaluation of $TWRP^{-1}$ within $dec^J_{TWRP,L}$ is done for a repeated tweak but on different input size, and thus it generates a uniform random string. The condition (46) is set with probability $1/2^\tau$;
  - $|C\|T| = |C_i\|T_i|$ but $C\|T \neq C_i\|T_i$. In this case, the evaluation of $TWRP^{-1}$ within $dec^J_{TWRP,L}$ is done for a repeated tweak and on identical input size, and thus it generates a uniform random string without replacement. The condition (46) is set with probability at most $2^{|C|}/(2^{|C|+\tau}-1) \leq 1/(2^\tau-1)$;
  - $C\|T = C_i\|T_i$. In this case, the evaluation of $TWRP^{-1}$ within $dec^J_{TWRP,L}$ is identical to that of the i^{th} encryption query. This case also implies that, necessarily, $\text{right}_{\nu-w}(N)\|A \neq \text{right}_{\nu-w}(N_i)\|A_i$, as otherwise the forgery would be trivial. The condition (46) is set only if

    $$J_L(\text{right}_{\nu-w}(N)\|A)=J_L(\text{right}_{\nu-w}(N_i)\|A_i)\,,$$

    which happens with probability at most ϵ as J is ϵ-universal.

Thus, summing over all q_d forgery attempts,

$$\left|\mathbf{Pr}\left(\mathcal{A}^{enc^J_{TWRP,L},dec^J_{TWRP,L}}=1\right)-\mathbf{Pr}\left(\mathcal{A}^{enc^J_{TWRP,L},\perp}=1\right)\right|\leq q_d\max\left\{\epsilon,\frac{1}{2^\tau-1}\right\}\,.$$

Together, we obtain from (45) that in the case of a nonce-respecting adversary,

$$\mathbf{Adv}^{\text{ae}}_{aaa}(\mathcal{A})\leq\mathbf{Adv}^{\text{stwprp}}_{TWBC}(\mathcal{A}')+q_d\max\left\{\epsilon,\frac{1}{2^\tau-1}\right\}\,.$$

Nonce-Randomizing Setting. In this case, the adversary $\mathcal{A}$ takes a random ν-bit nonce N for each encryption query.

For the confidentiality of *aaa*, we can note that different evaluations of $enc^J_{TWRP,L}$ behave independently for different values $\text{left}_w(N)$ and different lengths of $|P|$. Denote the q_e encryption queries by $\{(N_i, A_i, P_i, C_i, T_i)\}_{i=1}^{q_e}$. For any $W \in \{0,1\}^w$ and $\ell \in \mathbb{N}$ such that $\ell \geq 2n - \tau$, let $q_{W\ell}$ denote the number of encryption queries such that $\text{left}_w(N_i) = W$ and $|P| = \ell$. Let q' denote the number of strings W and lengths ℓ for which $q_{W\ell} > 0$ (i.e., q' denotes the number of different left parts of the nonces and length, and thus different independent instances of $TWRP$ that are invoked). Note that $\sum_{W,\ell} q_{W\ell} = q_e$.

Clearly, the evaluations of $TWRP$ within $enc^J_{TWRP,L}$ are independent if the tweak or the input length differs. We can thus focus on a fixed choice of $W \in \{0,1\}^w$ and $\ell \in \mathbb{N}$, consider the case of the adversary making $q_{W\ell}$ queries with $\text{left}_w(N) = W$ and $|P| = \ell$, and finally sum over all choices of W and ℓ. The adversary can distinguish $enc^J_{TWRP,L}$ from \$ in two different ways:

- There exist distinct $i_1, i_2 \in \{1, \ldots, q_e\}$, such that

  $$J_L\big(\text{right}_{\nu-w}(N_{i_1}) \| A_{i_1}\big) \| P_{i_1} = J_L\big(\text{right}_{\nu-w}(N_{i_2}) \| A_{i_2}\big) \| P_{i_2} \,.$$

  This necessarily means that $\text{right}_{\nu-w}(N_{i_1}) \| A_{i_1} \neq \text{right}_{\nu-w}(N_{i_2}) \| A_{i_2}$ and

  $$J_L(\text{right}_{\nu-w}(N_{i_1}) \| A_{i_1}) = J_L(\text{right}_{\nu-w}(N_{i_2}) \| A_{i_2}) \,,$$

  which happens with probability at most ϵ as J is ϵ-universal. Summing over all $\binom{q_{W\ell}}{2}$ choices, this happens with probability at most $\binom{q_{W\ell}}{2}\epsilon$;
- For all distinct $i_1, i_2 \in \{1, \ldots, q_e\}$,

  $$J_L\big(\text{right}_{\nu-w}(N_{i_1}) \| A_{i_1}\big) \| P_{i_1} \neq J_L\big(\text{right}_{\nu-w}(N_{i_2}) \| A_{i_2}\big) \| P_{i_2} \,.$$

  In this case, the $q_{W\ell}$ evaluations of $TWRP$ within $enc^J_{TWRP,L}$ result in a different $(\ell+\tau)$-bit string without repetition, and the adversary can distinguish from random with probability at most $\binom{q_{W\ell}}{2}/2^{\ell+\tau} \leq \binom{q_{W\ell}}{2}/2^{2n}$.

Together, we obtain that

$$\left|\mathbf{Pr}\left(\mathcal{A}^{enc^J_{TWRP,L},\perp} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\$,\perp} = 1\right)\right| \leq \sum_{W \in \{0,1\}^w} \sum_{\substack{\ell \in \mathbb{N} \\ \ell \geq 2n-\tau}} \binom{q_{W\ell}}{2} \left(\epsilon + \frac{1}{2^{2n}}\right) .$$

We bound $q_{W\ell}$ later on.

For the authenticity of *aaa*, as the adversary can choose nonces in decryption queries, the analysis is identical to that in the nonce-respecting setting, with the exception that in case "$|C\|T| = |C_i\|T_i|$ but $C\|T \neq C_i\|T_i$" there may be up to $q_{W\ell}$ earlier queries for the same left w bits of the nonce and the same ciphertext length, leading to the fact that this case sets (46) with probability at most $2^{|C|}/(2^{|C|+\tau} - q_{W\ell}) \leq 1/(2^\tau - q_{W\ell})$. Thus, summing over all q_d forgery attempts,

$$\left|\mathbf{Pr}\left(\mathcal{A}^{enc^J_{TWRP,L},dec^J_{TWRP,L}} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{enc^J_{TWRP,L},\perp} = 1\right)\right| \leq q_d \max\left\{\epsilon, \frac{1}{2^\tau - q_{W\ell}}\right\} .$$

Together, we obtain from (45) that in the case of a nonce-randomizing adversary,

$$\begin{aligned}\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') \\ + \sum_{W \in \{0,1\}^w} \sum_{\substack{\ell \in \mathbb{N} \\ \ell \geq 2n-\tau}} \binom{q_{W\ell}}{2} \left(\epsilon + \frac{1}{2^{2n}}\right) + q_d \max\left\{\epsilon, \frac{1}{2^\tau - q_{W\ell}}\right\} .\end{aligned} \tag{47}$$

We can use $\sum_{W,\ell} q_{W\ell} = q_e$ to obtain a naive bounding

$$\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') + \binom{q_e}{2} \left(\epsilon + \frac{1}{2^{2n}}\right) + q_d \max\left\{\epsilon, \frac{1}{2^\tau - q_e}\right\} .$$

This naive bounding is, in fact, matching the idea of an adversary freely choosing nonces. In other words, this bound applies to the case of nonce-misusing adversaries.

However, it is very unlikely that all q_e queries are for the same left w bits of the nonce. In particular, we can observe that for any $W \in \{0,1\}^w$ and $\ell \in \mathbb{N}$,

$$\mathbf{Pr}\left(q_{W\ell} > \gamma\right) \leq \binom{q_e}{\gamma + 1} \frac{1}{2^{\gamma w}}$$

(here, the length ℓ is not used in the probability computation as it can be freely chosen by the adversary), and we can assume that $q_{W\ell} \leq \gamma$ except for this loss. Concluding, we obtain from (47) that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') \\ + \frac{q_e}{\gamma}\binom{\gamma}{2} \left(\epsilon + \frac{1}{2^{2n}}\right) + q_d \max\left\{\epsilon, \frac{1}{2^\tau - \gamma}\right\} + \binom{q_e}{\gamma + 1} \frac{1}{2^{\gamma w}} .\end{aligned}$$

In our case, we always assume that the tag size τ and the tweak size w are large enough for $\gamma = 7$ to be sufficient (i.e., no 8-fold collision), and we finally obtain

$$\mathbf{Adv}^{\mathrm{ae}}_{aaa}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{stwprp}}_{TWBC}(\mathcal{A}') + 3q_e \left(\epsilon + \frac{1}{2^{2n}}\right) + q_d \max\left\{\epsilon, \frac{1}{2^\tau - 7}\right\} + \binom{q_e}{8} \frac{1}{2^{7w}} .$$

9 Interpretation of the Bounds

We will give an interpretation of our bounds of Corollary 3 and Theorem 5. In both comparisons, we assume that AES is secure, and we ignore the $\mathbf{Adv}^{\mathrm{prp}}_{E}$ term.

Memory Encryption. First, we consider memory encryption. Here, we assume that we have memory organized in 512-bit lines that are accessed at once. For this example, we want to explore how much data can be processed if we compare AES-XTS [21] with *bbb-ddd-AES*, while limiting the advantage to $\leq 2^{-32}$.

The advantage of AES-XTS is bounded by the birthday bound $\frac{\rho^2}{2^n}$, where ρ denotes the number of 128-bit blocks, and this means that we can encrypt $\approx 2^{48}$ blocks. In other words, if we have 4 Terabyte of RAM (2^{40} lines) encrypted with one key, we can write each line 2^6 times.

For comparison, let us have a look at *bbb-ddd-AES*. We can operate it for $v = 3$ and $m_{\max} = 4$ to match the 512-bit lines. In this way, we can write, e.g., 2^{60} lines (4 Exabyte) of RAM 2^{11} times, because (from Corollary 3)

$$\begin{aligned}\frac{4^4 2^{71\cdot 3}}{5\cdot 2^{256}} + \frac{\binom{4}{2}2^{71}}{2^{128}} + \frac{2\binom{4}{2}^2 2^{71\cdot 2}}{2^{256}} + 2^{60}\binom{2^{11}}{2}\left(\frac{4\cdot 4^2}{2^{128}} + \frac{2\cdot 4}{2^{128}} + \frac{1}{2^{256}}\right)& \\ \le \frac{1}{2^{37}} + \frac{1}{2^{54}} + \frac{1}{2^{107}} + \frac{1}{2^{41}} + \frac{1}{2^{44}} + \frac{1}{2^{175}} \le 2^{-36}\,.&\end{aligned}$$

More interestingly, *bbb-ddd-AES* allows for a much heavier write load to a part of the lines of the RAM. Let us assume that a part of all the 2^{60} lines, namely 2^{20} lines, are not written 2^{11} times but 2^{30} times each. This does not have a significant impact on the total data complexity. Hence, we can continue from the bound above and just re-evaluate the additional complexity coming from the enhanced tweak reuse. This results in an additional term of at most

$$2^{20}\binom{2^{30}}{2}\left(\frac{4\cdot 4^2}{2^{128}} + \frac{2\cdot 4}{2^{128}} + \frac{1}{2^{256}}\right)\frac{1}{2^{43}} + \frac{1}{2^{46}} + \frac{1}{2^{177}} \le 2^{-42}\,,$$

and we still have an advantage smaller than 2^{-32}. We want to remark that, when considering 2^{20} lines written 2^{30} times each alone, AES-XTS' bound on the advantage would be larger than 2^{-32}.

Authenticated Encryption. If we follow the recommendation for AES-GCM in TLS 1.3 [49], one is allowed to encrypt $2^{24.5}$ records of size up to $2^{14} + 1$ bytes, while keeping the advantage below 2^{-57}. We will now investigate how many records we can encrypt using *aaa-bbb-ddd-AES*. Here, we have a block size of a maximum of $2^{14} + 1 + 16$ bytes with a 16-byte tag, which are in total approximately $2^{10.002}$ 128-bit blocks. This means that we can allow for 2^{51} records, because (from Theorem 5 and Corollary 3, with $v = 2^{10.002} \le 2^{10.1}$, $m_{\max} = 2^{10.1}$, and $q_x = 2^{51}$)

$$\begin{aligned}&\frac{(2^{10.1})^4 2^{51\cdot 3}}{5\cdot 2^{256}} + \frac{\binom{2^{10.1}}{2}2^{51}}{2^{128}} + \frac{2\binom{2^{10.1}}{2}^2 2^{51\cdot 2}}{2^{256}} + 2^{51}\binom{1}{2}\left(\frac{4(2^{10.1})^2}{2^{128}} + \frac{2^{11.1}}{2^{128}} + \frac{1}{2^{256}}\right)\\ &\qquad\qquad + q_d \max\left\{\epsilon, \frac{1}{2^{128}-1}\right\}\\ &\le \frac{1}{2^{64.2}} + \frac{1}{2^{57.8}} + \frac{1}{2^{114.6}} + q_d \max\left\{\epsilon, \frac{1}{2^{128}-1}\right\}\\ &\le 2^{-57} + q_d \max\left\{\epsilon, \frac{1}{2^{128}-1}\right\}\,,\end{aligned}$$

assuming that we use a good universal hash function such as *Polyval*, i.e., with low ϵ, and that the number of decryption queries q_d is bounded. In conclusion, with these parameters, AES-GCM can encrypt up to 362 Gigabytes, while *aaa-bbb-ddd-AES* can encrypt up to 32 Exabytes.

10 High-Level Comparison

While *bbb-ddd-AES* is the only tweakable wide blockcipher *based on a blockcipher* known to us that can reach beyond birthday bound security, we still provide a comparison with other tweakable wide blockciphers. Before doing so, we remark that we do not compare *ddd-AES* and *bbb-ddd-AES* with other constructions that do not by default allow for arbitrary-length inputs, like CTET+ [14], or that are based on tweakable blockciphers, like PIV [54], THCTR [19] (see also [1,33]), and ZCZ* [8].

10.1 General Comparison

We start with the compact Table 1, that shows the workload in terms of blockcipher calls and finite field multiplications for processing $n \cdot m$ bits of input, where n is the block size. For Table 1, we assume that the tweak is available before the rest of the data (e.g., the plaintext). This means that all processing just depending on the tweak has already been done. An example use case where this applies to is when loading data from memory and storage, where the tweak is just the location (address) that indicates which data needs to be fetched.

As we can see in Table 1, *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES* all roughly need m blockcipher calls and $2m$ multiplications. Hence, they are in line with most of the other so-called hash-encrypt-hash constructions, which is remarkable considering that *bbb-ddd-AES* is the only scheme that achieves security beyond the birthday bound. There exist hash-encrypt-hash constructions that use so-called BRW polynomials [4, 48] to instantiate their universal hash functions. This allows to do the hashing with m multiplications instead of $2m$. Although *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES* could be instantiated with such hash function, we did not do so, since this leads to more complex implementations and the benefit in terms of speed seems to be limited [9].

The other category of schemes shown in Table 1 are the encrypt-mix-encrypt constructions. Those constructions do not use universal hash functions but have two full encryption layers and a mixing layer in the middle, which does not appear in this rough comparison, as we only consider blockcipher calls and multiplications. Furthermore, those schemes need to have the inverse of the blockcipher implemented, as we can see in Table 2.

10.2 Detailed Comparison

To be more specific, Table 2 shows a more detailed comparison of the different tweakable wide blockciphers. We can see that all our schemes are inverse free

Table 1: High-level comparison of *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES* with other blockcipher based tweakable wide blockciphers, assuming that all computations that only depend on the tweak are already processed. Here, we remark that *bbb-ddd-AES* is the only scheme achieving (conditional) beyond birthday bound security.

type	scheme	blockcipher calls	multiplications	ref.
hash-encrypt-hash	*ddd-AES*	m	$2m-2$	4.2
	ddd-AES$^+$	m	$2m-2$	4.3
	bbb-ddd-AES	$m+2$	$2m-2$	4.4
	HCTR2	m	$2m-2$	[15]
	HCH	$m+1$	$2m-2$	[10]
	HSE2	m	$2m-2$	[39]
	HEH[BRW]	m	$2+2\lfloor\frac{m-1}{2}\rfloor$	[53]
	HEH[Poly]	m	$2m-2$	[53]
	XCB	$m+1$	$2m-2$	[36]
	FAST[BRW]	$m+1$	$2+2\lfloor\frac{m-1}{2}\rfloor$	[9]
	FAST[Horner]	$m+1$	$2m$	[9]
encrypt-mix-encrypt	CMC	$2m$	0	[26]
	EME	$2m+1$	0	[27]
	EME*	$2m+\lceil\frac{m}{n}\rceil$	0	[25]

and have 4 individually fully parallel layers, where the last two can be partially executed in parallel. Furthermore, we see that *bbb-ddd-AES* is indeed the only scheme in the table that can be secure beyond the birthday bound. We can also observe from Table 2 that, without tweak pre-computation, *bbb-ddd-AES* needs roughly $2m$ blockcipher calls compared to *ddd-AES* needing roughly m.

This sounds like an overkill, but we will next show why, in many cases, implementations having access to dedicated hardware support can amortize those costs (we will confirm this intuition in Sec. 11). This means that the overhead of *bbb-ddd-AES* is not that big compared to *ddd-AES*. This comparison will also make clear why we opted to process the arbitrary length tweak of *ddd-AES*$^+$ using blockcipher calls instead of processing it via a universal hash function.

In detail, in Fig. 5, we see an instance of *bbb-ddd-AES* processing a 512-bit plaintext. We can clearly see that the mask generation for $\widetilde{XORP}$ just depends on the tweak, and the blockcipher calls associated with those are colored in blue (🍃) and purple (‽). Furthermore, we also see that a single multiplication of the bottom universal hash (shown in dark blue (℧)) can be computed once the computation of the associated PRF block (shown in orange (§)) is finished. Hence, the computation of these two layers can be interleaved.

In Fig. 6a, we show the execution flow of *bbb-ddd-AES*, assuming that we have support of two engines for computing the blockcipher calls and two engines doing the finite field multiplications. For this example, we assume that the execution time of the multiplications and the calls to the blockcipher are roughly

Table 2: High-level comparison of *ddd-AES*, *ddd-AES*$^+$, and *bbb-ddd-AES* with other blockcipher based tweakable wide blockciphers. For simplicity of notation, we only consider block lengths that are a multiple m of the underlying n-bit blockcipher. Furthermore, for designs supporting arbitrary length tweaks, we also count the data in terms of multiple t of n-bit blocks. Parallel layers indicates the number of layers that can in themselves be computed in parallel, but have to be computed after each other. In brackets, we give a relaxed condition on numbers of layers that can be computed in parallel if after partial completion of one layer, the next one can already be computed.

scheme	BC calls	multiplications	inverse free	tweak length	bbb	parallel layers	ref.
ddd-AES	$m+2$	$2m$	yes	fixed	no	4 (3)	4.2
ddd-AES$^+$	$m+t+1$	$2m$	yes	arbitr.	no	4 (3)	4.3
bbb-ddd-AES	$2m+4$	$2m$	yes	fixed	yes	4 (3)	4.4
HCTR2	m	$2m+t-1$	no	arbitr.	no	4 (3)	[15]
HCH	$m+3$	$2m-2$	no	fixed	no	5 (4)	[10]
HSE2	m	$2m+t-1$	no	arbitr.	no	4 (3)	[39]
HEH[BRW]	$m+1$	$2+2\lfloor\frac{m-1}{2}\rfloor$	no	fixed	no	≥4 (3)	[53]
HEH[Poly]	$m+1$	$2m-2$	no	fixed	no	4 (3)	[53]
XCB	$m+1$	$2m+t-1$	no	arbitr.	no	4	[36]
FAST[BRW]	$m+1$	$2{+}2\lfloor\frac{m-1}{2}\rfloor{+}\lfloor\frac{t}{2}\rfloor$	yes	arbitr.	no	≥5 (4)	[9]
FAST[Horner]	$m+1$	$2m+t$	yes	arbitr.	no	5 (4)	[9]
CMC	$2m+1$	0	no	fixed	no	$m+2$	[26]
EME	$2m+1$	0	no	fixed	no	3	[27]
EME*	$2m{+}\lceil\frac{m}{n}\rceil{+}t$	0	no	arbitr.	no	3/4/5	[25]

the same. We can see that many of the masks (blue (🍃) and purple (‽) blocks) can be processed in parallel to the execution of the first universal hash function (red (♪) blocks). The processing of the part of the first PRF (shown in green (※)) that depends on the result of the first universal hash can only start after the first universal hash function can be computed entirely. The computation of the parts of the second PRF (shown in orange (§)) that depend on the first PRF can only start after the green blocks are finished. However, the computation of the last universal hash (shown in dark blue (℧)) can already start when the first blocks of the second PRF are ready. Hence, those computations are interleaved.

If we now compare the execution flow of *bbb-ddd-AES* shown in Fig. 6a with the execution flow of *ddd-AES* shown in Fig. 6b, we see that *bbb-ddd-AES* is only marginally slower than *ddd-AES* in this scenario. This is essentially because the resources saved by saving blockcipher calls for mask generation are not fully translated to saving execution time: the engines for AES are just underutilized in this case. This is also the reason why we opted to process the arbitrary length tweak for *ddd-AES*$^+$ using blockcipher calls. This allows us to make use of those potentially underutilized AES engines.

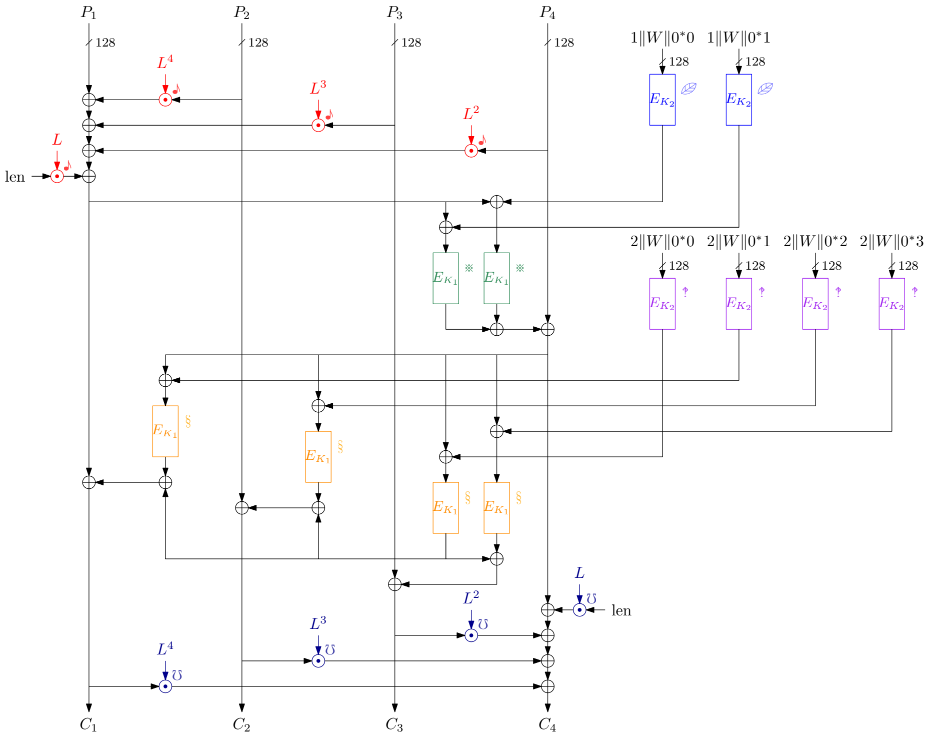
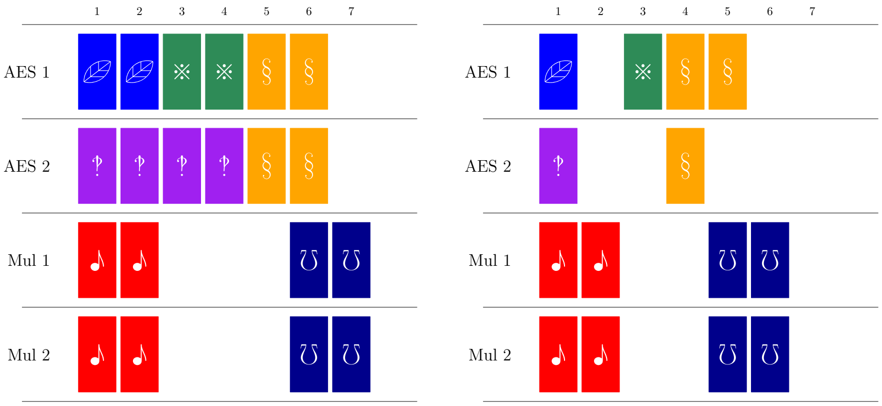


Fig. 5: *bbb-ddd-AES* processing 512-bit blocks.

11 Software Implementation

We implemented *ddd-AES* and *bbb-ddd-AES* with AES-128 in C using intrinsics to access the AES-NI instructions and carry-less multiplication instructions PCLMULQDQ. We implemented different approaches for *ddd-AES* and *bbb-ddd-AES*, essentially allowing to compute from 1 to 6 executions of AES and carry-less multiplication in parallel for most of the execution. Comparing the benchmarks of those, the ones with 3 or 4 possible parallel executions performed best, which is not surprising given the information on latency and throughput we got from the Intel® Intrinsics Guide [29]. The results of our benchmarks are given in Table 3. Note that we do not benchmark *bbb-ddd-AES* against other tweakable wide blockciphers because, as demonstrated in Section 10, there is no other blockcipher based scheme achieving security beyond the birthday bound.

Our results indicate that our postulations of Sec. 10 also translate into reality. For larger block sizes, the differences between *ddd-AES* and *bbb-ddd-AES* in performance are indeed small seeing 1.0 cycles/byte for *ddd-AES* versus 1.1 cycles/byte for *bbb-ddd-AES* considering large messages. To compare our results, we have also benchmarked CBC encryption with AES, which also provides at least a limited diffusion capability beyond one block. We observed that



(a) *bbb-ddd-AES*. (b) *ddd-AES*.

Fig. 6: Execution flow processing 512-bit blocks.

CBC encryption already needs 1.4 cycles/byte when encrypting messages of 2048 bytes measured with our toolchain, while arguably having worse properties than *bbb-ddd-AES*. In particular, CBC just provides diffusion in forward direction.

We also want to note that we expect better performance numbers on newer architectures that can make use of vectorized AES-NI and PCLMULQDQ instructions due to the high degree of parallelism of *bbb-ddd-AES*.

12 Conclusion

In this paper, we explored instances of the docked double decker construction that can make efficient use of already existing hardware that speeds up the execution of AES and GHash. We did this, so that the resulting tweakable wide blockcipher is essentially just a mode of operation for the AES blockcipher. We have also introduced a method to instantiate authenticated encryption with a tweakable wide blockcipher having a fixed length tweak, called *aaa*. This method shows good security bounds even in the case that the nonce is selected randomly. We hope that our work provides additional insights into the trade-offs of variable tweak length, security, and efficiency, and will foster the research of more tweakable wide blockcipher modes of operations.

In the process of designing the beyond birthday bound secure tweakable wide blockcipher *bbb-ddd-AES*, we also designed an efficient blockcipher based PRF called $\widetilde{XORP}$, which is able to process up to $2n$-bit inputs. We proved that this construction achieves around $2n/3$-bit security, provided tweaks are not reused too often. Since we do not have an attack matching the bound of $\widetilde{XORP}$, it remains future work to see if such an attack can be found, or if the bound can be improved.

Table 3: Performance in cycles/byte of *ddd-AES* and *bbb-ddd-AES* on an Intel® Core™ i7-10610U using RDTSC to get the value for cycles. Your costs and results may vary.

Message length (bytes)	32	48	64	96	128	256	512	1024	2048
ddd-AES x1	6	4.3	3.4	2.8	2.5	2.3	2.2	2.1	2.1
ddd-AES x2	6	3.9	3.2	2.5	2.0	1.7	1.5	1.3	1.3
ddd-AES x3	9	4.6	3.1	2.5	2.1	1.4	1.2	1.1	1.0
ddd-AES x4	7	4.3	3.5	2.6	2.3	1.6	1.3	1.1	1.0
ddd-AES x5	8	4.6	3.8	2.4	2.2	1.5	1.2	1.1	1.0
ddd-AES x6	7	4.6	3.6	2.9	2.1	1.7	1.2	1.1	1.0
bbb-ddd-AES x1	8	5.0	4.0	3.2	2.9	2.6	2.5	2.5	2.5
bbb-ddd-AES x2	9	5.1	3.9	3.0	2.6	1.9	1.6	1.4	1.3
bbb-ddd-AES x3	8	5.2	3.8	3.0	2.5	1.7	1.4	1.2	1.1
bbb-ddd-AES x4	8	5.0	4.1	3.0	2.8	1.9	1.4	1.2	1.1
bbb-ddd-AES x5	9	5.9	4.1	2.8	2.8	1.7	1.5	1.3	1.2
bbb-ddd-AES x6	9	5.2	4.4	3.3	2.6	2.0	1.4	1.3	1.2

ACKNOWLEDGEMENTS. Bart Mennink is supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099. We thank Samuel Neves for sharing his thoughts on the impossibility to replace the blockcipher calls in the masking of $\widehat{XORP}$ by finite field multiplication of input and a key, or more generally a universal hash function. We thank the anonymous reviewer of CRYPTO 2024 who pointed out a mistake in an earlier version of the proof, as well as all other reviewers for their useful feedback.

References

1. Andreeva, E., Bhati, A.S., Preneel, B., Vizár, D.: 1, 2, 3, Fork: Counter Mode Variants based on a Generalized Forkcipher. IACR Trans. Symmetric Cryptol. 2021(3), 1–35 (2021), https://doi.org/10.46586/tosc.v2021.i3.1-35
2. Bellare, M., Impagliazzo, R.: A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. Cryptology ePrint Archive, Report 1999/024 (1999), http://eprint.iacr.org/1999/024
3. Bellare, M., Krovetz, T., Rogaway, P.: Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In: Nyberg, K. (ed.) Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. Lecture Notes in Computer Science, vol. 1403, pp. 266–280. Springer (1998), https://doi.org/10.1007/BFb0054132
4. Bernstein, D.J.: Polynomial evaluation and message authentication. https://cr.yp.to/papers.html#pema (2007)
5. Bhargavan, K., Leurent, G.: On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of

the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 456–467. ACM (2016), https://doi.org/10.1145/2976749.2978423

6. Bhattacharjee, A., Dutta, A., List, E., Nandi, M.: CENCPP*: beyond-birthday-secure encryption from public permutations. Des. Codes Cryptogr. 90(6), 1381–1425 (2022), https://doi.org/10.1007/s10623-022-01045-z
7. Bhattacharya, S., Nandi, M.: Revisiting Variable Output Length XOR Pseudorandom Function. IACR Trans. Symmetric Cryptol. 2018(1), 314–335 (2018), https://doi.org/10.13154/tosc.v2018.i1.314-335
8. Bhaumik, R., List, E., Nandi, M.: ZCZ - Achieving n-bit SPRP Security with a Minimal Number of Tweakable-Block-Cipher Calls. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11272, pp. 336–366. Springer (2018), https://doi.org/10.1007/978-3-030-03326-2_12
9. Chakraborty, D., Ghosh, S., López, C.M., Sarkar, P.: FAST: Disk encryption and beyond. Adv. Math. Commun. 16(1), 185–230 (2022), https://doi.org/10.3934/amc.2020108
10. Chakraborty, D., Sarkar, P.: HCH: A New Tweakable Enciphering Scheme Using the Hash-Encrypt-Hash Approach. In: Barua, R., Lange, T. (eds.) Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4329, pp. 287–302. Springer (2006), https://doi.org/10.1007/11941378_21
11. Chen, S., Steinberger, J.P.: Tight Security Bounds for Key-Alternating Ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 327–350. Springer (2014), https://doi.org/10.1007/978-3-642-55220-5_19
12. Chen, Y.L., Davidson, M., Dworkin, M., Kang, J., Kelsey, J., Sasaki, Y., Turan, M.S., Chang, D., Mouha, N., Thompson, A.: Proposal of Requirements for an Accordion Mode: Discussion Draft for the NIST Accordion Mode Workshop 2024. https://csrc.nist.gov/pubs/other/2024/04/10/proposal-of-requirements-for-an-accordion-mode-dis/iprd (2024)
13. Cogliati, B., Dutta, A., Nandi, M., Patarin, J., Saha, A.: Proof of Mirror Theory for a Wide Range of $\xi_{\max}$. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 14007, pp. 470–501. Springer (2023), https://doi.org/10.1007/978-3-031-30634-1_16
14. Cogliati, B., Ethan, J., Lallemand, V., Lee, B., Lee, J., Minier, M.: CTET+: A Beyond-Birthday-Bound Secure Tweakable Enciphering Scheme Using a Single Pseudorandom Permutation. IACR Trans. Symmetric Cryptol. 2021(4), 1–35 (2021), https://doi.org/10.46586/tosc.v2021.i4.1-35
15. Crowley, P., Huckleberry, N., Biggers, E.: Length-preserving encryption with HCTR2. Cryptology ePrint Archive, Report 2021/1441 (2021), http://eprint.iacr.org/2021/1441

16. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002), https://doi.org/10.1007/978-3-662-04722-4
17. Daemen, J., Rijmen, V.: The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition. Information Security and Cryptography, Springer (2020), https://doi.org/10.1007/978-3-662-60769-5
18. Dai, W., Hoang, V.T., Tessaro, S.: Information-Theoretic Indistinguishability via the Chi-Squared Method. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10403, pp. 497–523. Springer (2017), https://doi.org/10.1007/978-3-319-63697-9_17
19. Dutta, A., Nandi, M.: Tweakable HCTR: A BBB Secure Tweakable Enciphering Scheme. In: Chakraborty, D., Iwata, T. (eds.) Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11356, pp. 47–69. Springer (2018), https://doi.org/10.1007/978-3-030-05378-9_3
20. Dworkin, M.: Recommendation for Block Cipher Modes of Operation Methods and Techniques (2001-12-01 2001), https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51031
21. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices (2010-01-18 2010), https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=904691
22. Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: Specification and Analysis. Cryptology ePrint Archive, Report 2017/168 (2017), http://eprint.iacr.org/2017/168
23. Gunsing, A., Daemen, J., Mennink, B.: Deck-Based Wide Block Cipher Modes and an Exposition of the Blinded Keyed Hashing Model. IACR Trans. Symmetric Cryptol. 2019(4), 1–22 (2019), https://doi.org/10.13154/tosc.v2019.i4.1-22
24. Gunsing, A., Mennink, B.: The Summation-Truncation Hybrid: Reusing Discarded Bits for Free. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12170, pp. 187–217. Springer (2020), https://doi.org/10.1007/978-3-030-56784-2_7
25. Halevi, S.: EME*: Extending EME to Handle Arbitrary-Length Messages with Associated Data. In: Canteaut, A., Viswanathan, K. (eds.) Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3348, pp. 315–327. Springer (2004), https://doi.org/10.1007/978-3-540-30556-9_25
26. Halevi, S., Rogaway, P.: A Tweakable Enciphering Mode. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2729, pp. 482–499. Springer (2003), https://doi.org/10.1007/978-3-540-45146-4_28
27. Halevi, S., Rogaway, P.: A Parallelizable Enciphering Mode. In: Okamoto, T. (ed.) Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings.

Lecture Notes in Computer Science, vol. 2964, pp. 292–304. Springer (2004), https://doi.org/10.1007/978-3-540-24660-2_23

28. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust Authenticated-Encryption AEZ and the Problem That It Solves. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 15–44. Springer (2015), https://doi.org/10.1007/978-3-662-46800-5_2
29. Intel: Intel® Intrinsics Guide Version 3.6.9. https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html (July 2024)
30. Iwata, T.: New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In: Robshaw, M.J.B. (ed.) Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4047, pp. 310–327. Springer (2006), https://doi.org/10.1007/11799313_20
31. Iwata, T., Mennink, B., Vizár, D.: CENC is Optimally Secure. Cryptology ePrint Archive, Report 2016/1087 (2016), http://eprint.iacr.org/2016/1087
32. Kampanakis, P., Campagna, M., Crocket, E., Petcher, A.: Practical Challenges with AES-GCM and the need for a new mode and wide-block cipher (Oct 2023), https://csrc.nist.gov/Presentations/2023/practical-challenges-with-aes-gcm
33. Khairallah, M.: A note on "Tweakable HCTR: A BBB Secure Tweakable Enciphering Scheme". Cryptology ePrint Archive, Report 2024/600 (2024), http://eprint.iacr.org/2024/600
34. Liskov, M.D., Rivest, R.L., Wagner, D.A.: Tweakable Block Ciphers. In: Yung, M. (ed.) Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2442, pp. 31–46. Springer (2002), https://doi.org/10.1007/3-540-45708-9_3
35. Mattsson, J.P., Smeets, B., Thormarker, E.: Proposals for Standardization of Encryption Schemes (Oct 2023), https://csrc.nist.gov/Presentations/2023/proposal-for-standardization-of-encryption-schemes
36. McGrew, D.A., Fluhrer, S.R.: The Security of the Extended Codebook (XCB) Mode of Operation. In: Adams, C.M., Miri, A., Wiener, M.J. (eds.) Selected Areas in Cryptography, 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4876, pp. 311–327. Springer (2007), https://doi.org/10.1007/978-3-540-77360-3_20
37. McGrew, D.A., Viega, J.: The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In: Canteaut, A., Viswanathan, K. (eds.) Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3348, pp. 343–355. Springer (2004), https://doi.org/10.1007/978-3-540-30556-9_27
38. Mennink, B., Neves, S.: Encrypted Davies-Meyer and Its Dual: Towards Optimal Security Using Mirror Theory. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10403, pp. 556–583. Springer (2017), https://doi.org/10.1007/978-3-319-63697-9_19

39. Minematsu, K., Matsushima, T.: Tweakable Enciphering Schemes from Hash-Sum-Expansion. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4859, pp. 252–267. Springer (2007), https://doi.org/10.1007/978-3-540-77026-8_19
40. Nachef, V., Patarin, J., Volte, E.: Feistel Ciphers - Security Proofs and Cryptanalysis. Springer (2017), https://doi.org/10.1007/978-3-319-49530-9
41. NIST: The Third NIST Workshop on Block Cipher Modes of Operation 2023. https://csrc.nist.gov/events/2023/third-workshop-on-block-cipher-modes-of-operation, accessed: 2023-12-12
42. Patarin, J.: Étude des Générateurs de Permutations Basés sur le Schéma du D.E.S. Ph.D. thesis, Université Paris 6, Paris, France (Nov 1991)
43. Patarin, J.: On Linear Systems of Equations with Distinct Variables and Small Block Size. In: Won, D., Kim, S. (eds.) Information Security and Cryptology - ICISC 2005, 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3935, pp. 299–321. Springer (2005), https://doi.org/10.1007/11734727_25
44. Patarin, J.: The "Coefficients H" Technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5381, pp. 328–345. Springer (2008), https://doi.org/10.1007/978-3-642-04159-4_21
45. Patarin, J.: Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. Cryptology ePrint Archive, Report 2010/287 (2010), http://eprint.iacr.org/2010/287
46. Patarin, J.: Mirror Theory and Cryptography. Cryptology ePrint Archive, Report 2016/702 (2016), http://eprint.iacr.org/2016/702
47. Public Comments on FIPS 197 - Advanced Encryption Standard (AES) (2021), https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/initial-comments/fips-197-initial-public-comments-2021.pdf
48. Rabin, M.O., Winograd, S.: Fast evaluation of polynomials by rational preparation. Communications on Pure and Applied Mathematics 25, 433–458 (1972)
49. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. Request for Comments (RFC) 8446 (August 2018), https://tools.ietf.org/html/rfc8446
50. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In: Lee, P.J. (ed.) Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3329, pp. 16–31. Springer (2004), https://doi.org/10.1007/978-3-540-30539-2_2
51. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. https://www.cs.ucdavis.edu/~rogaway/papers/offsets.pdf (2004), full version of [50]
52. Salowey, J., Choudhury, A., McGrew, D.: AES Galois Counter Mode (GCM) Cipher Suites for TLS. Request for Comments (RFC) 5288 (August 2008), https://tools.ietf.org/html/rfc5288
53. Sarkar, P.: Efficient tweakable enciphering schemes from (block-wise) universal hash functions. IEEE Trans. Inf. Theory 55(10), 4749–4760 (2009), https://doi.org/10.1109/TIT.2009.2027487

54. Shrimpton, T., Terashima, R.S.: Salvaging Weak Security Bounds for Blockcipher-Based Constructions. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 429–454 (2016), https://doi.org/10.1007/978-3-662-53887-6_16
55. Thompson, A., Turan, M.S.: NIST Workshop on the Requirements for an Accordion Cipher Mode 2024: Workshop Report. https://csrc.nist.gov/pubs/ir/8537/final (2024)