Design and Analysis of Lightweight Authenticated Encryption with Associated Data

A thesis submitted to Indian Statistical Institute
in partial fulfillment of the thesis requirements for the degree of
Doctor of Philosophy in Computer Science

Author:
Bishwajit Chakraborty

Supervisor:
Prof. Mridul Nandi

Applied Statistics Unit
Indian Statistical Institute
203, B. T. Road, Kolkata,

January, 2023

To all who remained anonymous in history..

Acknowledgment

This thesis results from many people's guidance, inspiration, and assistance, not just my own. At this time of recognition, I'd like to express my heartfelt gratitude to each of them.

Before anything else, I must express my gratitude to Mridul-da, my mentor and supervisor, for all the support and guidance he has given me over the years. My relationship with Mridul-da began in 2015 when I first attended his introductory probability course during the third semester of my master's program and later approached him about taking a research course under his guidance during the subsequent semester. He has continued to provide inspiration, suggestions, and direction ever since. His highly intuitive yet mathematically rigorous approach to solving technical problems has significantly influenced the direction of my research. He helped me with all the problems discussed in this thesis, and I am thankful for that. Mridul-da has been more than just a Ph.D. supervisor. In my early years as a research scholar, he confidently encouraged me to investigate areas of my interest. He vowed to help me even in those areas that weren't in his regular field of expertise. Throughout the years, I have confided in him on several occasions during small and large personal crises, and he has always been willing to help me with his tolerance, wisdom, and support. I know that without him, I would not be the person I am today, and I don't have the words to convey how I feel about it.

Every student, teacher, and non-teaching staff member at ISI has welcomed me since I first enrolled as an M. Math student in 2013. I've had the privilege of getting to know and work with some of the most intelligent people I know throughout my time at ISI. My co-authors, Ashwin Bhaiya, Nilanjan Da, Chandranan, and of course Mridul Da, deserve particular thanks. Additionally, I'd like to express my gratitude to the many people who have contributed significantly to my life over the years, including Soumya, Munmum di, Ritam da, Avik da, Ananda, Anik, Arghya, Arka, Sayantan, Suprita, Arnab, Diganta, Atanu, Mayukh, Naveen, Viswas, Ujjwal, Goonj, Avishek, Samir, Shion da. I lack the words to express my thanks for their eternal love.

I appreciate that ISI provided all the resources, particularly the computing facility. At CSSC and the assistance needed for my research. An expression of gratitude is due to the Applied Statistics Division of RFAC for their timely review and renewal of my research fellowship. Their feedback and counsel were crucial for me to move on in that direction.

Last but not least, I want to express my sincere gratitude to my family for their never-ending support and inspiration. I want to express my gratitude to my parents, Shova Chakraborty and Sudhangshu Sekhar Chakraborty, for inspiring me to pursue my hobbies. They had helped me at every stage of my life and have inspired and strengthened me when I most needed it. The wellspring of my deepest love has always been my sister (Baishakhi). She is the one who has the most faith in me.

Bishwajit Chakraborty
Kolkata, January 2023

List of Publications

This thesis is based on the following works :

1. Chakraborty, Bishwajit, Ashwin Jha, and Mridul Nandi. *"On the Security of Sponge-type Authenticated Encryption Modes."* IACR Transactions on Symmetric Cryptology (2020): 93-119. DOI: `https://doi.org/10.13154/tosc.v2020.i2.93-119`.

2. Chakraborty, Bishwajit, and Mridul Nandi. "mixFeed." Round 2 Candidate, NIST Lightweight Cryptography Standardization Process (2019). `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/mixFeed-spec-round2.pdf`.

3. Chakraborty, Bishwajit, and Mridul Nandi. "ORANGE." Round 2 Candidate, NIST Lightweight Cryptography Standardization Process (2019). `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/orange-spec-round2.pdf`.

4. Chakraborty, Bishwajit, and Mridul Nandi. *"The mF mode of authenticated encryption with associated data."* Journal of Mathematical Cryptology 16, no. 1 (2022): 73-97. DOI `https://doi.org/10.1515/jmc-2020-0054`.

5. Chakraborty, Bishwajit, Nilanjan Dutta, and Mridul Nandi. *"On the security of Full-Rate Sponge-type AEAD modes."* Submitted to Journal of Mathematical Cryptology 2022, Dated 13 July, 2022. Manuscript Id JMC.2022.0021.

Abstract

The demand for lightweight cryptographic protocols has skyrocketed in the previous decade, especially for many resource-constrained devices such as IoT end nodes and RFID tags. The goal of lightweight cryptography is to use less memory, processing resources, and power to create a less secure but adequate security solution on devices with limited resources. As a result, lightweight cryptographic protocols should be easier and faster to use than traditional ones. The National Institute of Standards and Technology (NIST) began a standardization process in 2018 for lightweight cryptographic (LWC) encryption schemes that have at least one scheme with a key length of 128 bits and can achieve security against adversaries capable of making $2^{50} - 1$ byte queries and 2^{112} computations. Running a good feedback loop on a specified lightweight primitive is a typical approach to designing such schemes. In more technical terms, such approaches process data input into fixed-sized chunks. Following initialization, the primitive output is passed through a suitable function with one of these partial data inputs to provide acceptable output and feedback for the primitive's subsequent execution. In this thesis, we start by looking at existing schemes that use pseudo-random permutations as the underlying primitive and can be thought of as different variations of the Sponge scheme. We give all such structures a new name Transform-then-Permute and do a security analysis. We show that we can reduce the security of such schemes to a graph theoretic security game called the "multi-chain security game," based on the underlying feedback function of the Transform-then-Permute design. Then, for various feedback functions, we employ various techniques to limit an adversary's advantage against this game. As a result, we developed some novel or significantly enhanced security bounds for popular permutation-based authenticated encryption techniques. Finally, we aim to design authenticated encryption techniques to process the most data possible in each protocol iteration. We construct and examine two generalized systems, Full-rate Transform-then-Permute and mF, which use a pseudo-random permutation and a tweakable block cipher as their underlying primitives, respectively. We illustrate how to instantiate these general structures to meet the NIST LwC requirement.

Contents

Chapter 1

Introduction

1.1 Lightweight Ctryptography

The lightweight cryptography trades off implementation costs, execution times, security, performance, and energy consumption on devices with limited resources. Lightweight cryptography's goal is to create a less solid but adequate security solution that can operate in resource-constrained devices using less memory, computational power, and energy. IoT end nodes and RFID tags [89] are just two examples of the wide range of low-resource devices affected by lightweight cryptography. Standard cryptographic protocols like AES [92] and SHA3 [52], which function well in computer systems, are challenging to implement due to implementation size, throughput or speed, and energy consumption. Therefore, it is easy to anticipate that lightweight cryptographic protocols will be easier to use and faster than traditional cryptographic protocols.

The need for lightweight cryptographic protocols has significantly increased over the past decade. The National Institute of Standards and Technology (NIST) has started a process to standardize lightweight cryptographic (LWC) encryption techniques to meet this need. Most of the NIST lightweight cryptography standardization process submissions fall into one of three general categories.

- Permutation Based.
- (Tweakable) block cipher based.
- Stream cipher based and others.

We tabulate the 52 first-round candidates for the NIST LwC standardization process in Table 1.1 along with some citations to their security analysis. Note that these

citations are not exhaustive. Readers can find many detailed surveys about the standardization process in [91, 107, 106].

Permutation Based	(Tweakable) block cipher based	Stream cipher based, and others.
ACE [80, 7, 74, 24, 82]	COMET [58, 72, 16, 59]	Bleep64
ASCON [47, 97]	ESTATE [102, 31]	CLAE
CiliPadi	FlexAEAD	Grain-128AEAD [62, 41]
CLX	ForkAE [10, 4]	Quartet
DryGASCON [105]	GIFT-COFB [8]	Triad
Elephant [113, 104, 25, 110]	HyENA [32, 33]	
Fountain	LAEM	
GAGE	Lilliput-AE	
Gimli [61, 27, 78, 54]	Limdolen	
HERN	LOTUS-AEAD, and LOCUS-AEAD [29, 30]	
ISAP [50, 45]	mixFeed [71, 72, 75]	
KNOT [111]	Pyjamask [56, 51]	
ORANGE [49, 70]	Qameleon	
Oribatida [23]	Remus [66, 60]	
PHOTON-Beetle	Romulus [66]	
Shamash	SAEAES [90]	
SIV-TEM-PHOTON	Saturnin [28]	
SNEIK	Simple	
SPARKLE [12, 11, 81]	SIV-Rijndael	
SPIX [73]	SKINNY-AEAD [13]	
SpoC [73]	Spook [88, 15]	
Subterranean 2.0 [77, 63, 103]	SUNDAE-GIFT [40]	
Sycon	Thank Goodness It's Friday (TGIF)	
WAGE [3]	TinyJAMBU [109, 100]	
Xoodyak [112, 79]	TRIFLE	
Yarará		

Table 1.1: Classification of NIST LwC candidates.

1.2 Sponge-based AEAD Schemes

Bertoni et al. first suggested the Sponge function as a method of operation for variable output length hash functions at the ECRYPT Hash Workshop [17]. Since several of the proposals in the NIST's SHA-3 competition were based on the Sponge paradigm, it attracted immediate attention. Most notably, of the five finalists, Keccak [22] emerged victorious, followed by JH, [108]. In due course, the Sponge mode found use in message authentication [17, 21], pseudorandom sequence generation [19], and the "duplex mode"

for authenticated encryption [20]. Twelve Sponge-based entries were made specifically for the recently finished CAESAR competition to develop authenticated encryption with associated data (AEAD) systems. The lightweight applications (resource-constrained settings) use-case winner Ascon [46] also takes advantage of the duplex form of authenticated encryption.

At a very high level, Sponge-type constructions consist of a b-bit state, split into a c-bit inner state, called the capacity, and an r-bit outer state, called the rate, where $b = c + r$. Traditionally, in Sponge-like modes, data absorption and squeezing are done via the rate part, i.e., r bits at a time. SpoC [2], a round 2 submission to the NIST LwC standardization process, is a notable exception, where the absorption is done via the capacity part, and the squeezing is done via the rate part. In [18], Bertoni et al. proved that the Sponge construction is indifferentiable from a random oracle with a birthday-type bound in the capacity. While it is well-known that this bound is tight for hashing, for keyed applications of the Sponge, especially authenticated encryption schemes, such as duplex mode, the security could be significantly higher.

1.2.1 Existing Security Bounds for Sponge-type AEAD Schemes

Sponge-type authenticated encryption is mostly done via the duplex construction [20]. The duplex mode is a stateful construction that consists of an initialization interface and a duplexing interface. Initialization creates an initial state using the underlying permutation π, and each duplexing call to π absorbs and squeezes r bits of data. The security of Sponge-type AEAD modes can be represented and understood in terms of two parameters, namely the data complexity D (total number of initialization and duplexing calls to π), and the time complexity T (total number of direct calls to π).

Initially, Bertoni et al. [20] proved that duplex is as strong as Sponge, i.e. secure up to $DT \ll 2^c$. Mennink et al. [87] introduced the full-state duplex and proved that this variant is secure up to $DT \ll 2^\kappa$, $D \ll 2^{c/2}$, where κ is the key size. Jovanovic et al. [68] proved privacy up to $D \ll \min\{2^{b/2}, 2^\kappa\}$, $T \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^\kappa\}$, and integrity up to $DT \ll 2^c$, $D \ll \min\{2^{c/2}, 2^\kappa, 2^\tau\}$, $T \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^\kappa\}$, where τ denotes the tag size. Note that integrity has an additional restriction that $D \ll 2^{c/2}$, where D is dominated by the decryption data complexity. Daemen et al. [43] gave a generalization of duplex that has built-in multi-user security. Very recently, a tight privacy analysis [69] is provided. However, one of the dominating restrictions present in all of the existing integrity analysis of duplex authenticated encryption is $DT \ll 2^c$. Moreover, no forgery attack with a matching bound is known. A recent variant of duplex mode, called the Beetle mode of operation [35], modifies the duplexing phase by introducing a combined feedback based absorption/squeezing, similar to the feedback paradigm of CoFB [36]. In [35], Chakraborti et al. showed that feedback based duplexing actually helps in improving the security bound, mainly to get rid of the condition $DT \ll 2^c$. They showed privacy up to $DT \ll 2^b$, $D \ll 2^{b/2}$, $T \ll 2^c$, and integrity up to $D \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^r\}$, $T \ll \min\{2^{c-\log_2 r}, 2^r, 2^{b/2}\}$, with the assumptions that $\kappa = c$ and $\tau = r$.

From all the available analysis, it is evident the security of a Sponge type design, increasing r degrades the security, and if $2^r \geq \min\{2^{b-\log T}, 2^{b-\log D}\}$ then all the existing constructions become insecure. ORANGE-Zest [39] is our round 2 submission to the ongoing NIST LwC standardization process where some extra-state is induced in the encryption/decryption protocol to construct a full-rate Sponge type AEAD scheme. Dobraunig et al. [49] showed that the ORANGE-Zest mode is insecure by constructing

an adversary which forges successfully after making only two encryption queries.

Sponge-type AEAD in the Context of NIST LwC Standardization Process

In it's report [106], NIST classifies all the 17 permutation based lightweight AEAD candidates selected for round 2 of it's LwC standardization process as follows :

- *Classical sponge with public permutation :* DryGASCON, Gimli, KNOT, Subterranean 2.0, Xoodyak
- *Classical sponge with stronger initialization and finalization :* ACE, ASCON, SPIX, Spook, WAGE
- *Classical sponge with keyed permutation :* SAEAES, TinyJAMBU
- *Modified sponge with public permutation :* ORANGE, Oribatida, PHOTON-Beetle, SPARKLE, SpoC.

Out of these, Xoodyak, ASCON, TinyJAMBU, PHOTON-Beetle and SPARKLE were further promoted as the finalists.

According to NIST's call for proposals, an AEAD scheme must have one *primary* member with a key length of 128 bits and be secure up to 2^{112} computations and $2^{50}-1$ byte queries. A traditional Sponge-based scheme must have a capacity size of at least 160-bit. All Sponge-based submissions to NIST LwC standardization process use at least 192-bit capacity, except the round 1 submission CLX. Mege [84], in the 1st round; official comments described how to mount an attack on the primary variant of CLX.

On the other hand, the known bound for Beetle imposes certain limitations on the state size and rate. Specifically, Beetle-based schemes require approximately 120-bit

capacity and approximately 120-bit rate, i.e., a permutation size of at least 240 bits to achieve NIST LwC requirements.

In light of the ongoing NIST LwC standardization process, it is an interesting problem to investigate whether these limitations can be relaxed.

1.3 (Tweakable) Block Cipher-based AEAD Schemes

A block cipher is a cryptographic primitive consisting of two algorithms namely (E, D) such that the deterministic function E takes a fixed length key and a fixed length (also called a block) message as input to produce a single block of ciphertext output. In notation, $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ is a function such that, for all $K \in \{0,1\}^\kappa$, E_K is a permutation. For any given $K \in \{0,1\}^\kappa$ and any $M, C \in \{0,1\}^n$, we have $D(K,C) = M$ if and only if $E(K,M) = C$.

A tweakable block cipher (TBC) [76] is a deterministic function $\tilde{E}$ which takes a fixed length tweak along with a fixed length key and a block of message to output a single block of ciphertext, such that it acts as a family of block ciphers. In notation, $\tilde{E} : \{0,1\}^t \times \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ is a function such that there exists a family of block ciphers $\{E_{tw}\}$ such that for, each $tw \in \{0,1\}^t$, $\tilde{E}(tw, K, M) = E_{tw}(K, M)$.

1.3.1 Security of TBC-based AEAD Modes.

For many AEAD modes [38, 64, 65, 67, 57] with an underlying TBC $\tilde{E}$ and a given key K, the encryption algorithm works as follows. It takes an initial input IV and then computes $Y_0 = \tilde{E}(tw_0, K, IV)$ where the tweak tw_0 depends on the nonce and perhaps,

some other parameters. To process the ith block of message (or associated data), it uses a feedback function FB, which takes Y_{i-1} and M_i as inputs and X_i and C_i as outputs (only X_i in case of AD). It then outputs $Y_i = \tilde{E}(tw_0, K, X_i)$ which is used to process the next block of associated data or message. After processing the entire message, the next TBC output is used as the input for the tag generation protocol. In an AEAD scheme with appropriate feedback function FB, one can bound the privacy and the forgery advantage of any adversary as

$$\mathbf{Adv}^{\text{priv}}_{AEAD}(T, D) \leq \mathbf{Adv}^{\text{tprp}}_{\tilde{E}}(T, D) \tag{1.1}$$

$$\mathbf{Adv}^{\text{forge}}_{AEAD}(T, D) \leq \mathbf{Adv}^{\text{tprp}}_{\tilde{E}}(T, D) + \mathcal{O}(\frac{D}{2^n}) \tag{1.2}$$

where T, D are respectively the time and data complexity of the adversaries playing the security games and n is the TBC state size. Hence, the security of such a TBC-based AEAD scheme can be bounded by bounding the TPRP advantage of the underlying TBC.

A dedicated TBC construction is conjectured to have security; hence, we can instantiate the mode with such a secure TBC. In addition to the dedicated constructions, there are some known constructions of TBC based on a block cipher. For example, XEX [99] has birthday-bound security.

AEAD modes such as Remus-N [64] and mixFeed [38] use explicit block cipher based TBC constructions as their underlying TBC. Given a block cipher E and a feedback function ρ, the ICE1 TBC (used in Remus-N1) uses a key derivation function KDF which takes inputs K and N and outputs $L = \rho(E_K(N))$. Given input X and tweak (N, ω, i), it then outputs $E_{K'}(X)$, where $K' = 2^i L \oplus \omega$. As described in [44], an

adversary can make D queries with a fixed input 0^n and varying the tweaks to get D outputs $Y_1, \ldots, Y_D$. Notice that ICE1 uses two block cipher calls, the first one is used as the key derivation function and the second one is used as the encryption function with the key which is derived from the key derivation output of the first. So, let $K_1, \ldots, K_D \in \{0,1\}^\kappa$ be those intermediate keys. Now, if the adversary pre-computes $Y'_1, \ldots, Y'_T$ by making T primitive block cipher calls with input 0^n and keys $K'_1, \ldots, K'_T$ then the probability that $Y_i = Y'_j$ and hence $K_i = K'_j$ for some $1 < i < D, 1 < j < T$ is bounded by $\frac{DT}{2^\kappa}$. Hence

$$\mathbf{Adv}^{\text{tprp}}_{\mathsf{ICE1}}(T, D) \approx \frac{DT}{2^\kappa}.$$

Security of TBC-based AEAD in the Context of NIST LwC Standardization Process

Note that while constructing any TBC, a general objective is to achieve security with T close to 2^κ. For instance, according to NIST [93], $T \geq 2^{112}$ for $n \geq 128$ and $T \geq 2^{224}$ when $n = 256$. Hence, a bound such as the above doesn't provide adequate security as it limits the size of D. Accordingly, given $D \geq 2^{50}$, ICE1 doesn't satisfy NIST requirements. The authors of [44] also extended the above attack to Remus-N1. ICE2 (used in Remus-N2) provides higher security of the form $DT/2^{2\kappa}$. Still, it uses an auxiliary key (which can be viewed as a combination of XEX and ICE1) which costs some additional state to hold this auxiliary key. This cost motivates the following question.

> *Can we have an instantiation of TBC based on block cipher without using any auxiliary key but still provide adequate security (for instance, up to the NIST desired level) ?*

At a glance, it seems impossible to design such a TBC. However, we show that such security is possible for the AEAD mode based on a TBC, which does not use any auxiliary key. We must have new reductions than what we usually have (as mentioned in Eq.1.1-1.2). The distinguishing attack for Remus-N1 [64] requires fixing the associated data block as it is the input of the underlying block cipher, which can be avoided if the first block is defined to be nonce. This principle is adapted in mixFeed [38]. Both mixFeed [38] and Remus [64] are similar in nature (as described above) with two main differences. First, they use different feedback functions, which do not affect security. The second difference is that unlike Remus, mixFeed processes the nonce before processing the associated data and hence uses one extra TBC call. In the nonce-respecting model, the inputs of the block cipher vary in the case of mixFeed. Hence, the attack due to [44] can be avoided.

1.4 Our Contributions

In this thesis, inspired by the NIST LwC requirements, we extend a long line of research on the security of Sponge-type and TBC-based AEAD schemes.

In Chapter 4, we study Sponge-type AEAD construction with a generalization of the feedback function used in the duplexing interface, that encompasses the feedback used in Sponge-duplex, Beetle, SpoC etc. We call it the Transform-then-Permute construction. We show that the AEAD security of this Transform-then-Permute construction is bounded by the adversary's ability to construct a special data structure, called the *multi-chains*. In particular, we show that for a specific class of feedback functions containing the Beetle and SpoC modes, optimal AEAD security is achieved. Further, we

derive improved security bound for generic Sponge-duplex schemes. We also show a matching attack exploiting the multi-chains. As a corollary of this, we give

1. a security proof validating the security claims of SpoC,

2. an improved and tight bound for Beetle.

3. an improved (not tight) bound for general Sponge-duplex.

In Chapter 5, we introduced a full-rate sponge-type general construction inspired by the Transform-then-Permute construction with the extra state. We dubbed it the Full-rate Transform-then-Permute construction (frTtP in short). We show that the extra state compensates for the increase in the size of the rate part and makes the construction secure. We further show with a suitable initiation of this extra-state, one can achieve a full-rate Sponge type mode with security up to $D \ll 2^{c/2}, T \ll 2^c$. We further introduce our full-rate Sponge-based AEAD scheme called ORANGE-Zest which is a 2nd round candidate in the NIST LwC standardization process. We show that the weakness in the full-rate construction ORANGE-Zest is due to an improper initialization protocol. With a proper initialization function, maintaining the properties mentioned in our theorem statement, one can get a secure full-rate Sponge-type AEAD scheme using the ORANGE-Zest feedback function. In addition, we have also considered some full-rate feedback functions used in block-cipher based constructions such as CoFB [36] and HyENA [34], and show that with some proper modifications these feedback functions can also be used to construct secure full-rate Sponge-type modes.

Finally, in Chapter 6, we formalize the TBC-based AEAD mode, which is used in mixFeed, and we call it mF. We reduce the security of the mF mode in terms of the

security of the underlying TBC against two newly introduced input restricted adversaries, namely (1) the μ-respecting TPRP adversary and (2) (μ, λ)-multi commitment prediction adversary. These two notions are non-standard but weaker than standard TPRP security notions because an adversary is μ-respecting if it can make at most μ queries with the same input. Consider a nonce-respecting AEAD scheme where the nonce is processed in the beginning. Since the initial TBC input is not repeated and the other TBC inputs depend on their previous TBC outputs, we can consider the security of the underlying TBC in terms of μ-respecting adversaries with small μ. In the case of mF mode in nonce-respecting setup, we can choose μ to be about n to achieve the following bounds.

$$\mathbf{Adv}^{\text{priv}}_{\mathsf{mF}}(T, D) \leq \mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}(T, D) + \mathcal{O}\left(\frac{\mu^2 D}{2^n}\right) \tag{1.3}$$

$$\begin{aligned}\mathbf{Adv}^{\text{forge}}_{\mathsf{mF}}(T, D) \leq \mathbf{Adv}^{(\mu,D)\text{-mcp}}_{\tilde{E}}(T, D) + 2 \cdot \mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}(T, D) \\ + \mathcal{O}\left(\frac{\mu^2 D}{2^n}\right) + \frac{2D}{2^{\frac{n}{2}}}.\end{aligned} \tag{1.4}$$

where, the new security advantage terms $\mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}$ and $\mathbf{Adv}^{(\mu,D)\text{-mcp}}_{\tilde{E}}$ respectively denote the adversarial advantage of any μ-respecting TPRP adversary and any (μ, D)-multi commitment prediction adversary. For more detailed definitions, see Section 6.2.

We then study these two security notions for two instantiations of TBC. $\mathsf{mixFeed}$ is one such instantiation. For $\mathsf{mixFeed}$, we claimed the scheme's security under the assumption that the AES key scheduling algorithm has a small number of short permutation cycles. Hence, the probability of finding a key in those cycles is also small.

Khairallah [72] observed that if this assumption is violated, it may lead to weak-key attacks on mixFeed. Later, Leurent et al. [75] confirmed this observation by explicitly finding a large number of short cycles in the AES key scheduling algorithm. We try to interpret this weakness in our general notations and conclude that the weakness is only due to the use of the AES key schedule in the Key updation function. Our second instantiation of TBC based on block cipher using primitive element multiplication (we call the overall AEAD $\mathsf{mF}_{\mathsf{prim}}$) achieves the bounds,

$$\mathbf{Adv}_{\tilde{E}}^{\mu\text{-tprp}}(T, D) \approx \frac{\mu T}{2^n}. \tag{1.5}$$

$$\mathbf{Adv}_{\tilde{E}}^{(\mu,D)\text{-mcp}}(T, D) \approx \frac{\mu T}{2^n} + \frac{D}{2^{\frac{n}{2}}}. \tag{1.6}$$

Plugging these results in our previous results, we obtain that the $\mathsf{mF}_{\mathsf{prim}}$ mode is well secured within the NIST requirements.

Chapter 2

Preliminaries

2.1 Notational Setup

We start by setting up a few mathematical notations which will be followed throughout this thesis. For any $n \in \mathbb{N}$, $(n]$ (res. $[n]$) signifies the set $\{1, 2, \ldots, n\}$ (res. $\{0, 1, \ldots, n\}$). $\{0,1\}^n$ denotes the set of bit strings of length n, $\{0,1\}^* := \bigcup_{n \geq 0}\{0,1\}^n$, and $\mathsf{Perm}(n)$ signifies the set of all permutations over $\{0,1\}^n$. We say that the two distinct strings $a = a_1 \ldots a_m$ and $b = b_1 \ldots b_{m'}$ have a common prefix of length $n \leq \min\{m, m'\}$ if $a_i = b_i$ for all $i \in (n]$, and $a_{n+1} \neq b_{n+1}$. $\lceil x \rceil_n$ (res. $\lfloor x \rfloor_n$) designates the most (res. least) significant n bits of any bit string x with $|x| \geq n$. For any rn bit string $a \in \{0,1\}^{rn}$, $(a_1, \ldots, a_n) \xleftarrow{r} a$ denotes the parsing of a into r bit strings $a_1, \ldots, a_n$ such that $a = a_n \| \cdots \| a_1$. We define the falling factorial $(n)_k := n(n-1)\cdots(n-k+1)$.

For any finite set $\mathcal{X}$, $(\mathcal{X})_q$ signifies the set of all q-tuples with distinct elements from $\mathcal{X}$. $\mathsf{X} \leftarrow_{\$} \mathcal{X}$ signifies the uniform sampling of X from $\mathcal{X}$, which is independent of all other previously sampled random variables. An uniform sampling of t random variables $\mathsf{X}_1, \ldots, \mathsf{X}_t$ from $\mathcal{X}$ without replacement is denoted by $(\mathsf{X}_1, \ldots, \mathsf{X}_t) \xleftarrow{\mathsf{wor}} \mathcal{X}$.

2.2 Mathematical Background

Proposition 1. *[83] Given any $m \times n$ matrix A and $n \times l$ matrix B with entries in a field $\mathbb{F}$,*

$$\mathsf{rank}(A \cdot B) \leq \min\{\mathsf{rank}(A), \mathsf{rank}(B)\}.$$

Corollary 1. *For any $n \times n$ square matrix A with entries in a field $\mathbb{F}$,*

$$\mathsf{rank}(A^i) \geq \mathsf{rank}(A^n) \ \forall i \in \mathbb{N}.$$

Proposition 2. *For any $m \times m$ matrix A and $n \times m$ matrix B with entries in a field $\mathbb{F}$,*

$$\mathsf{rank}\left(\begin{bmatrix} A \\ B \end{bmatrix}\right) \leq \mathsf{rank}(A) + \mathsf{rank}(B).$$

Theorem 1. *(Sylvester rank inequality)[83] For any $m \times n$ matrix A and $n \times k$ matrix B with entries in a field $\mathbb{F}$,*

$$\mathsf{rank}(AB) \geq \mathsf{rank}(A) + \mathsf{rank}(B) - n.$$

Proposition 3. *(Markov's inequality)[53] If X is a nonnegative random variable and $a > 0$,*

$$\Pr[X > a] \leq \frac{\mathsf{Ex}[X]}{a}.$$

2.3 Results on Multicollision

In this section, we briefly revisit some valuable results on the expected value of maximum multicollision in some specific types of samples. This problem has seen a lot of interest (see, for instance, [55, 6, 101, 96]) in the context of the complexity of hash table[1] probing. However, most of the results available in the literature are given in asymptotic forms. We state some relevant results in a more concrete form, following similar proof strategies and probability calculations as before. Moreover, we also extend these results for samples that, although not uniform, have high entropy, almost close to uniform.

2.3.1 Expected Maximum Multicollision in a Uniform Random Sample

Let $\mathsf{X}_1, \ldots, \mathsf{X}_q \leftarrow_{\$} \mathcal{D}$ where $|\mathcal{D}| = N$ and $N \geq 2$. We denote the maximum multicollision random variable for the sample as $\mathsf{mc}_{q,N}$. More precisely, $\mathsf{mc}_{q,N} = \max_a |\{i : \mathsf{X}_i = a\}|$. For any integer $\rho \geq 2$,

$$\begin{aligned}
\Pr\left[\mathsf{mc}_{q,N} \geq \rho\right] &\leq \sum_{a \in \mathcal{D}} \Pr\left[|\{i : \mathsf{X}_i = a\}| \geq \rho\right] \\
&\leq N \cdot \frac{\binom{q}{\rho}}{N^{\rho}} \\
&\leq N \cdot \frac{q^{\rho}}{N^{\rho}\rho!} \\
&\leq N \cdot \left(\frac{qe}{\rho N}\right)^{\rho}.
\end{aligned}$$

[1] A famous data structure used for efficient searching applications.

We justify the inequalities in the following way: The first inequality is due to the union bound. If there are at least ρ indices for which X_i takes value a, we can choose the first ρ indices in $\binom{q}{\rho}$ ways. This justifies the second inequality. The last inequality follows from the simple observation that $e^\rho = \sum_{i \geq 0} \rho^i / i! \geq \rho^\rho / \rho!$. Thus, we have

$$\Pr\left[\mathsf{mc}_{q,N} \geq \rho\right] \leq N \cdot \left(\frac{qe}{\rho N}\right)^\rho. \tag{2.1}$$

For any positive integer-valued random variable Y bounded above by q, we define another random variable Y' as

$$\mathsf{Y}' = \begin{cases} \rho - 1 & \text{if } \mathsf{Y} < \rho \\ q & \text{otherwise.} \end{cases}$$

Clearly, $\mathsf{Y} \leq \mathsf{Y}'$ and

$$\mathsf{Ex}\left[\mathsf{Y}\right] \leq (\rho - 1) + q \cdot \Pr\left[\mathsf{Y} \geq \rho\right].$$

Using Eq. (2.1), and the above relation we can prove the following results for the expected value of maximum multicollision. We write $\mathsf{mcoll}(q, N)$ to denote $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right]$. So from the above relation,

$$\mathsf{mcoll}(q, N) \leq (\rho - 1) + qN \cdot \left(\frac{qe}{\rho N}\right)^\rho \tag{2.2}$$

for all positive ρ. We use this relation to prove an upper bound of $\mathsf{mcoll}(q, N)$ by plugging in some suitable value for ρ.

Proposition 4. *For $N \geq 4,\ n = \log_2 N$,*

$$\mathsf{mcoll}(q, N) \leq \begin{cases} \frac{4 \log_2 q}{\log_2 \log_2 q} & \text{if } 4 \leq q \leq N \\ 5n \left\lceil \frac{q}{nN} \right\rceil & \text{if } N < q \end{cases}$$

Proof. We first prove the result when $q = N$. A simple algebra shows that for $n \geq 2$, $\left(\frac{e \log_2 n}{4n}\right) \leq n^{-\frac{1}{2}}$. In other words, $\left(\frac{e}{\rho}\right)^{\rho} \leq N^{-2}$ where $\rho = 4n/\log_2 n$. So

$$\mathsf{mcoll}(q, N) \leq \rho - 1 + N^2 \cdot \left(\frac{e}{\rho}\right)^{\rho} \leq \rho.$$

When $q < N$, we can simply bound $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right] \leq \mathsf{Ex}\left[\mathsf{mc}_{q,q}\right] \leq \frac{4 \log_2 q}{\log_2 \log_2 q}$.

For $N < q \leq Nn$, we choose $\rho = 4n$. Now,

$$\begin{aligned} \mathsf{mcoll}(q, N) &\leq 4n - 1 + nN^2 \times \left(\frac{e}{4}\right)^{4n} \\ &\leq 4n - 1 + nN^2/4^n \leq 5n. \end{aligned}$$

When $q \geq nN$, we can group them into $\lceil q/nN \rceil$ samples each of size exactly nN (we can add more samples if required). This would prove the result when $q \geq nN$. $\square$

Remark 1. *Note that, similar bound as in Proposition 4 can be achieved in the case of non-uniform sampling. For example, when we sample $\mathsf{X}_1, \ldots, \mathsf{X}_q \xleftarrow{wor} \{0,1\}^b$ and then define $\mathsf{Y}_i = \lceil \mathsf{X}_i \rceil_r$ for some $r < b$. In this case, we have*

$$\Pr\left[\mathsf{Y}_{i_1} = a, \cdots, \mathsf{Y}_{i_\rho} = a\right] \leq \frac{(2^{b-r})_\rho}{(2^b)_\rho} \leq \frac{1}{2^{r\rho}}.$$

This can be easily justified as we have to choose the remaining $b - r$ bits distinct (as

$\mathsf{X}_1, \ldots, \mathsf{X}_q$ *must be distinct). So, the same bound as given in Proposition 4 can be applied for this distribution.*

2.3.2 A Special Example of Non-Uniform Random Sample

In this thesis, we consider the following non-uniform random samples. Let $x_1, \ldots, x_q$ be distinct and $y_1, \ldots, y_q$ be distinct b bits. Let Π denote the random permutation over b bits, $\Pi^2 := \Pi \circ \Pi$ denotes the composition of Π with itself. We define $\mathsf{Z}_{i,j} = \Pi(x_i) \oplus \Pi^{-1}(y_j)$. Now, for all distinct $i_1, \ldots, i_\rho$, distinct $j_1, \ldots, j_\rho$ and $a \in \{0,1\}^b$, we want to bound $\Pr\left[\mathsf{Z}_{i_1,j_1} = a, \cdots, \mathsf{Z}_{i_\rho,j_\rho} = a\right]$. By abuse of notations we write both i_k and j_k as k.

Let $N := 2^b$. We can assume $a = 0^b$. Since otherwise, we consider $\Pi'(x) = \Pi(x) \oplus a$ which is also a random permutation and consider $y'_i = y_i \oplus a$ instead of y_i, $\forall 1 \leq i \leq \rho$. Note that y'_i's are clearly distinct. So the problem reduces to bounding

$$\begin{aligned}\theta &:= \Pr\left[\Pi^2(x_1) = y_1, \cdots, \Pi^2(x_\rho) = y_\rho\right] \\ &= \sum_{c^\rho} \Pr\left[\Pi(x_1) = c_1, \Pi(c_1) = y_1, \cdots, \Pi(x_\rho) = c_\rho, \Pi(c_\rho) = y_\rho\right]\end{aligned}$$

We say that c^ρ valid if $c_i = x_j$ if and only if $c_j = y_i$. The set of all such valid tuples is denoted as V. For any valid c^ρ, define $S := \{x_1, \ldots, x_\rho\} \cup \{c_1, \ldots, c_\rho\}$. Then, $\Pr\left[\Pi(x_1) = c_1, \Pi(c_1) = y_1, \cdots, \Pi(x_\rho) = c_\rho, \Pi(c_\rho) = y_\rho\right] = \frac{1}{(N)_{|S|}}$. On the other hand, if c^ρ is not valid then the above probability is zero. Let V_s be the set of all valid tuples for which $|S| = s$.

If $|S| = 2\rho - k$, then we must have exactly k many pairs $(i_1, j_1), \ldots, (i_k, j_k)$ such

that $c_i = x_j$. Now the number of ways this k-many pairs can be chosen is bounded by ρ^{2k}. The remaining $\rho - k$ many c_i's can be chosen in $(N-k)_{\rho-k}$ ways. Hence, $|V_{2\rho-k}| \leq \rho^{2k}(N-k)_{\rho-k}$.

$$\begin{aligned}
\Pr\left[\Pi^2(x_i) = y_i \ \forall 1 \leq i \leq \rho\right] &= \sum_{s=\rho}^{2\rho} \sum_{c^\rho \in V_s} \Pr\left[\Pi(x_i) = c_i, \Pi(c_i) = y_i \ \forall 1 \leq i \leq \rho\right] \\
&\leq \sum_{s=\rho}^{2\rho} \frac{|V_s|}{(N)_s} \leq \sum_{k=0}^{\rho} \frac{|V_{2\rho-k}|}{(N)_{2\rho-k}} \\
&\leq \sum_{k=0}^{\rho} \frac{\rho^{2k}(N-k)_{\rho-k}}{(N)_{2\rho-k}} \leq \sum_{k=0}^{\rho} \frac{\rho^{2(\rho-k)}}{(N-2\rho)^\rho} \\
&\leq \left(\sum_{k=0}^{\rho} \frac{1}{\rho^{2k}}\right) \cdot \left(\frac{\rho^2}{N-2\rho}\right)^\rho \leq 2 \cdot \left(\frac{\rho^2}{N-2\rho}\right)^\rho
\end{aligned}$$

Since the sample space $\{(x_i, y_j)\}_{i,j\in[q]}$ is of size q^2, we denote the maximum multicollision random variable for the sample as $\mathsf{mc}'_{q^2,N}$. Then we have by a similar analysis as in the previous section,

$$\Pr\left[\mathsf{mc}'_{q^2,N} \geq \rho\right] \leq 2N \cdot \binom{q^2}{\rho} \cdot \left(\frac{\rho^2}{N-2\rho}\right)^\rho \leq 2N\left(\frac{q^2 e\rho}{N-2\rho}\right)^\rho.$$

We write $\mathsf{mcoll}'(q^2, N)$ to denote $\mathsf{Ex}\left[\mathsf{mc}'_{q^2,N}\right]$. So from the above relation,

$$\mathsf{mcoll}'(q^2, N) \leq (\rho - 1) + 2q^2N \cdot \left(\frac{q^2 e\rho}{N-2\rho}\right)^\rho$$

Proposition 5. *For $b > 16$,*

$$\mathsf{mcoll}'(q^2, N) \leq \frac{4b}{\log_2 b}\left\lceil \frac{b^2q^2}{N} \right\rceil.$$

Proof. Let $b^2q^2 \leq N$. Since $N > 2^{16}$, $\rho = \frac{4b}{\log_2 b} \implies q^2 \leq \frac{N-2\rho}{\rho^2}$. Hence, $2q^2N \cdot \left(\frac{q^2e\rho}{N-2\rho}\right)^\rho \leq N^2 \cdot \left(\frac{e}{\rho}\right)^\rho$. Now, $\left(\frac{e}{\rho}\right)^\rho \leq \left(\frac{e}{4}\right)^{4b} \leq \frac{1}{N^2} \implies N^2 \cdot \left(\frac{e}{\rho}\right)^\rho \leq 1$.

Now for $q^2 \geq \frac{N}{b^2}$ we can group the q^2 samples into $\left\lceil \frac{b^2q^2}{N} \right\rceil$ groups each of size exactly $\frac{N}{b^2}$ (we can add more samples if required). This would prove the bounds. □

2.3.3 A Generalization of the Non-Uniform Random Sample

Here, we study a generalization of the above problem, which will be useful when a non-invertible linear function is sandwiched between two random permutation calls. For example, this happens in case of PHOTON-Beetle [9] and duplex [20].

Let $x_1, \ldots, x_q$ be distinct and $y_1, \ldots, y_q$ be distinct b-bit strings. Let Π denote the random permutation over b bits, $L : \{0,1\}^b \to \{0,1\}^b$ be a linear function with rank $\mathsf{rank}(L)$, and $\varphi := \Pi \circ L \circ \Pi$. We define $\mathsf{Z}_{i,j} = L(\Pi(x_i)) \oplus \Pi^{-1}(y_j)$. Now, for all distinct $i_1, \ldots, i_\rho$, distinct $j_1, \ldots, j_\rho$ and $a \in \{0,1\}^b$, we want to bound $\Pr\left[\mathsf{Z}_{i_1,j_1} = a, \cdots, \mathsf{Z}_{i_\rho,j_\rho} = a\right]$. By a slight abuse of notations we write both i_k and j_k as k.

Let $N := 2^b$. We can assume $a = 0^b$. Since otherwise, we consider $\Pi'(x) = \Pi(x \oplus a)$ which is also a random permutation and consider $x'_i = x_i \oplus a$ instead of x_i, $\forall 1 \leq i \leq \rho$. Note that x'_i's are clearly distinct. So the problem reduces to bounding

$$\begin{aligned}\theta &:= \Pr\left[\varphi(x_1) = y_1, \cdots, \varphi(x_\rho) = y_\rho\right] \\ &= \sum_{c^\rho} \Pr\left[\Pi(x_1) = c_1, \Pi(L(c_1)) = y_1, \cdots, \Pi(x_\rho) = c_\rho, \Pi(L(c_\rho)) = y_\rho\right]\end{aligned}$$

For all $1 \leq i \leq \rho$, let $d_i = L(c_i)$. We say that c^ρ *valid* if $d_i = x_j$ if and only if $c_j = y_i$. The

set of all such valid tuples is denoted as V. For any valid c^ρ, define $S := \{x_1, \ldots, x_\rho\} \cup \{d_1, \ldots, d_\rho\}$. Then, $\Pr[\Pi(x_1) = c_1, \Pi(d_1) = y_1, \cdots, \Pi(x_\rho) = c_\rho, \Pi(d_\rho) = y_\rho] = \frac{1}{(N)_{|S|}}$. On the other hand, if c^ρ is not valid then the above probability is zero. Let V_s be the set of all valid tuples for which $|S| = s$.

We say that d_i is *old* if $d_i = x_j$ for $1 \leq i, j \leq \rho$. If $|S| = 2\rho - k$, then we must have exactly k many pairs $(i_1, j_1), \ldots, (i_k, j_k)$ such that $d_i = x_j$. Now, the number of ways these k-many pairs can be chosen is bounded by ρ^{2k}. This fixes all old d_i values. Then, the number of c_i values corresponding to old d_i values is bounded by at most $\rho^{2k}\tilde{N}^k$ where $\tilde{N} = 2^{b-\mathsf{rank}(L)}$, as once we fix the $b - \mathsf{rank}(L)$ bits there is a unique solution for $L(c_i) = d_i$. Once we fix the c_i values corresponding to the old d_i values, then the remaining $\rho - k$ many c_i's can be chosen in $(N-k)_{\rho-k}\tilde{N}^{\rho-k}$ ways. Hence, $|V_{2\rho-k}| \leq \rho^{2k}(N-k)_{\rho-k}\tilde{N}^\rho$.

$$\begin{aligned}
\Pr[\varphi(x_i) = y_i \ \forall 1 \leq i \leq \rho] &= \sum_{s=\rho}^{2\rho} \sum_{c^\rho \in V_s} \Pr[\Pi(x_i) = c_i, \Pi(d_i) = y_i, \ \forall 1 \leq i \leq \rho] \\
&\leq \sum_{s=\rho}^{2\rho} \frac{|V_s|}{(N)_s} \leq \sum_{k=0}^{\rho} \frac{|V_{2\rho-k}|}{(N)_{2\rho-k}} \\
&\leq \sum_{k=0}^{\rho} \frac{\rho^{2k}(N-k)_{\rho-k}\tilde{N}^\rho}{(N)_{2\rho-k}} \leq \sum_{k=0}^{\rho} \frac{\rho^{2(\rho-k)}\tilde{N}^\rho}{(N-2\rho)^\rho} \\
&\leq \left(\sum_{k=0}^{\rho} \frac{1}{\rho^{2k}}\right) \cdot \left(\frac{\rho^2\tilde{N}}{N-2\rho}\right)^\rho \leq 2 \cdot \left(\frac{\rho^2\tilde{N}}{N-2\rho}\right)^\rho
\end{aligned}$$

Since the sample space $\{(x_i, y_j)\}_{i,j\in[q]}$ is of size q^2, we denote the maximum multicollision random variable for the sample as $\widetilde{\mathsf{mc}}_{q^2,N,\mathsf{rank}(L)}$. Then, we have by a similar analysis as

in the previous section,

$$\Pr\left[\widetilde{\mathsf{mc}}_{q^2,N,\mathsf{rank}(L)} \geq \rho\right] \leq 2N \cdot \binom{q^2}{\rho} \cdot \left(\frac{\rho^2 \widetilde{N}}{N-2\rho}\right)^{\rho} \leq 2N\left(\frac{q^2 e\rho \widetilde{N}}{N-2\rho}\right)^{\rho}.$$

We write $\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L))$ to denote $\mathsf{Ex}\left[\widetilde{\mathsf{mc}}_{q^2,N,\mathsf{rank}(L)}\right]$. So from the above relation,

$$\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L)) \leq (\rho - 1) + 2q^2 N \cdot \left(\frac{q^2 e\rho \widetilde{N}}{N-2\rho}\right)^{\rho}.$$

Finally, we have the following upper bound on $\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L))$.

Proposition 6. *For $b > 16$,*

$$\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L)) \leq \frac{4b}{\log_2 b}\left\lceil \frac{b^2 q^2}{2^{\mathsf{rank}(L)}} \right\rceil.$$

Proof. Let $b^2q^2 \leq 2^{\mathsf{rank}(L)}$. Since $N > 2^{16}$, $\rho = \frac{4b}{\log_2 b} \implies q^2 \leq \frac{N-2\rho}{\rho^2 \widetilde{N}}$. Hence, $2q^2N \cdot \left(\frac{q^2 e\rho}{N-2\rho}\right)^{\rho} \leq N^2 \cdot \left(\frac{e}{\rho}\right)^{\rho}$. Now, $\left(\frac{e}{\rho}\right)^{\rho} \leq \left(\frac{e}{4}\right)^{4b} \leq \frac{1}{N^2} \implies N^2 \cdot \left(\frac{e}{\rho}\right)^{\rho} \leq 1$.

Now, for $b^2q^2 > 2^{\mathsf{rank}(L)}$ we can group the q^2 samples into $\left\lceil \frac{b^2q^2}{2^{\mathsf{rank}(L)}} \right\rceil$ groups each of size exactly $\frac{2^{\mathsf{rank}(L)}}{b^2}$ (we can add more samples if required). This would prove the bounds. □

Remark 2. *Note that, for* $\mathsf{rank}(L) = b$, $\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L)) = \mathsf{mcoll}'(q^2, N)$.

2.3.4 Multicollisions in Context of the Analysis of Sponge-type AEAD

In a later chapter, we will use the bound on the expected number of multicollisions to give tight security bound for Transform-then-Permute and some of its instantiations.

Here, we note that multicollisions have been previously studied in context with the duplex mode, most notably in [43] and [69]. However, there is a fundamental difference between our approach and the previously used strategies in [43, 69]. In the following, r, c and b have their usual meaning in context of Sponge, i.e., $b = r + c$.

In [43], the authors try to upper bound a parameter called the multicollision limiting function $\nu^q_{r,c}$. Assume we distribute q balls into 2^r bins, one at a time, where the bin for each ball is selected uniformly at random and independent of other choices. Then, $\nu^q_{r,c}$ is defined as the smallest natural number x such that $\Pr[\mathsf{mc}_{q,2^r} > x] < x/2^c$. On a closer inspection of the proof, one can see that the $\nu^q_{r,c}$ is dependent upon b and $\lambda = q2^{-r}$. The authors derive bounds for $\nu^q_{r,c}$, for three cases, viz. $\lambda < 1$, $\lambda = 1$, and $\lambda > 1$.

In [69], the authors upper bound $\Pr[\mathsf{mc}_{q,2^r} > \rho]$ to q/S, where $S = \min\{2^{b/2}, 2^c\}$ and ρ is viewed as a function of r and c. Basically, based on the value of r and c, they derive choices for ρ, such that the desired probability is bounded by q/S. To derive sharp bounds on ρ for various choices of r and c, they employ a detailed analysis involving Sterling's approximation and Lambert W function.

In contrast to the above strategies, we are interested in good estimates for the expectation of $\mathsf{mc}_{q,2^r}$ depending upon the relationship between q and 2^r. Further, our analysis is much more straightforward.

2.4 Security Definitions of AEAD

Authenticated Encryption with Associated Data: An authenticated encryption scheme with associated data functionality, or AEAD in short [59], is a tuple of deterministic algorithms $\mathsf{AEAD} = (\mathsf{enc}, \mathsf{dec})$, defined over the *key space* $\mathcal{K}$, *nonce space* $\mathcal{N}$, *associated data space* $\mathcal{A}$, *message space* $\mathcal{M}$, *ciphertext space* $\mathcal{C}$, and *tag space* $\mathcal{T}$, where:

$$\mathsf{enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T} \quad \text{and} \quad \mathsf{dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}.$$

Here, enc and dec are called the encryption and decryption algorithms, respectively, of the AEAD. Further, it is required that $\mathsf{dec}(K, N, A, \mathsf{enc}(K, N, A, M)) = M$ for any $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$. For all key $K \in \mathcal{K}$, we write $\mathsf{enc}_K(\cdot)$ and $\mathsf{dec}_K(\cdot)$ to denote $\mathsf{enc}(K, \cdot)$ and $\mathsf{dec}(K, \cdot)$, respectively. In this paper, we have $\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T} \subseteq \{0,1\}^*$ and $\mathcal{C} = \mathcal{M}$, so we use $\mathcal{M}$ instead of $\mathcal{C}$ wherever necessary.

2.4.1 Privacy

The *privacy advantage* [98] of an adversary $\mathscr{A}$ over an AEAD is defined as $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{AEAD}}(\mathscr{A}) = |\Pr[\mathscr{A}^{\mathsf{enc}_K} = 1] - \Pr[\mathscr{A}^{\$} = 1]|$, where \$ returns a random output string that is the same length as the output size of enc_K The AEAD_K's *privacy advantage* is given by

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{AEAD}}(q, \sigma, t) = \max_{\mathscr{A}} \mathbf{Adv}^{\mathrm{priv}}_{\mathsf{AEAD}}(\mathscr{A})$$

where the total number of blocks in all of the encryption queries is at most σ and the maximum is calculated over all of the nonce-respecting adversaries $\mathscr{A}$ that are running in time t and making at most q encryption queries.

2.4.2 Forgery

We say that a nonce-respecting oracle adversary $\mathscr{A}^{\mathsf{enc}_K,\mathsf{dec}_K}$ forges $\mathsf{AEAD} = (\mathsf{enc}_K, \mathsf{dec}_K)$ if $\mathscr{A}$ is able to make a fresh and valid query (N, A, C, T) to dec_K. By fresh query, we mean that the adversary does not make any previous query (N, A, M) to enc_K such that $\mathsf{enc}_K(N, A, M) = (C, T)$. We say a decryption query is valid, if $\mathsf{dec}_K(N, A, C, T) \neq \perp$. The *forging advantage* [98] of an adversary $\mathscr{A}$ is written as

$$\mathbf{Adv}^{\text{forge}}_{\mathsf{AEAD}}(\mathscr{A}) = \Pr\left[\mathscr{A}^{\mathsf{enc}_K,\mathsf{dec}_K} \text{ forges}\right]$$

and we write

$$\mathbf{Adv}^{\text{forge}}_{\mathsf{AEAD}}(q, \sigma, t) = \max_{\mathscr{A}} \mathbf{Adv}^{\text{forge}}_{\mathsf{AEAD}}(\mathscr{A})$$

where the maximum is taken over all adversary $\mathscr{A}$ running in time t, making at most q_e nonce-respecting encryption queries with maximum σ_e blocks and making at most q_d decryption queries with maximum σ_d blocks. We define $q = q_e + q_d$ and $\sigma = \sigma_e + \sigma_d$. Note that the decryption queries are not necessarily nonce-respecting i.e. nonce can be repeated in the decryption queries and, an encryption query and a decryption query can use the same nonce. However, all nonces used in encryption queries are distinct.

2.4.3 AEAD Security in the Random Permutation Model

Let $\Pi \xleftarrow{\$} \mathsf{Perm}(b)$, Func denote the set of all functions from $\mathcal{N} \times \mathcal{A} \times \mathcal{M}$ to $\mathcal{M} \times \mathcal{T}$ such that for any input $(*, *, M)$ the output is of length $|M| + t$ for some predefined constant t and $\Gamma \xleftarrow{\$} \mathsf{Func}$. Let $\perp$ denote the degenerate function from $(\mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$ to $\{\perp\}$. For simplicity, we simply refer to the oracle that corresponds to a function (such as enc, Π etc.) by its own name. The superscript $\pm$ designates a two-way access to Π.

Definition 1. *Let* AE_Π *be an AEAD scheme, based on the random permutation* Π*, defined over* $(\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$*. The AEAD advantage of any nonce respecting adversary* $\mathscr{A}$ *against* AEAD_Π *is defined as,*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{AEAD}_\Pi}(\mathscr{A}) := \left| \Pr_{\substack{K \xleftarrow{\$} \mathcal{K} \\ \Pi^{\pm}}} \left[\mathscr{A}^{\mathsf{enc}_K, \mathsf{dec}_K, \Pi^{\pm}} = 1 \right] - \Pr_{\Gamma, \Pi^{\pm}} \left[\mathscr{A}^{\Gamma, \perp, \Pi^{\pm}} = 1 \right] \right|. \tag{2.3}$$

Note that $\mathscr{A}^{\mathsf{enc}_K, \mathsf{dec}_K, \Pi^{\pm}}$ denotes $\mathscr{A}$'s response after its interaction with enc_K, dec_K, and $\Pi^{\pm}$, respectively. Similarly, $\mathscr{A}^{\Gamma, \perp, \Pi^{\pm}}$ denotes $\mathscr{A}$'s response after its interaction with Γ, $\perp$, and $\Pi^{\pm}$.

2.5 Security Definitions of Tweakable Block Cipher

TWEAKABLE BLOCK CIPHER: A tweakable block cipher, or TBC in short, is a deterministic algorithm $\tilde{E}$, defined over the *key space* $\mathcal{K}$, *tweak space* $\mathcal{T}$, *message cum ciphertext space* $\mathcal{M}$ as

$$\tilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}.$$

For all key $K \in \mathcal{K}$, we write $\tilde{E}_K(\cdot)$ to denote $\tilde{E}(K, \cdot)$. A TBC with $\mathcal{M} = \{0,1\}^n$ is called an n-bit TBC. In this thesis, we have $\mathcal{K}, \mathcal{T} \subseteq \{0,1\}^*$.

2.5.1 TPRP-Security

Let $\tilde{E}$ be an n-bit tweakable block cipher with tweak space $\mathcal{T}$. The *TPRP-advantage* [76] of $\tilde{E}$ against an oracle adversary $\mathscr{A}$ is defined as

$$\mathbf{Adv}^{\mathrm{tprp}}_{\tilde{E}}(\mathscr{A}) = |\Pr\left[\mathscr{A}^{\tilde{E}_K} = 1\right] - \Pr\left[\mathscr{A}^{\tilde{\Pi}} = 1\right]|$$

where $\tilde{\Pi}$ is chosen uniformly from the set of all functions $\tilde{\pi} : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$, where, for every $tw \in \mathcal{T}$, $\tilde{\pi}(tw, \cdot)$ is a permutation on $\{0,1\}^n$. We call $\tilde{\Pi}$ a tweakable random permutation. We write $\mathbf{Adv}^{\mathrm{tprp}}_{\tilde{E}}(q,t) = \max_{\mathscr{A}} \mathbf{Adv}^{\mathrm{tprp}}_{\tilde{E}}(\mathscr{A})$ where maximum is taken over all adversaries $\mathscr{A}$ running in time t making at most q queries.

2.6 Coefficient H Technique

Consider any deterministic yet computationally bounded adversary $\mathscr{A}$ using a black box type interaction with one of two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$ and trying to differentiate between them. The query-response tuple associated with $\mathscr{A}$'s interaction with its oracle is called its transcript. A transcript ω may also contain any other information that the oracle decides to reveal to the distinguisher at the end of the game's query-response phase. This expanded definition of transcript will be taken into consideration. Suppose Θ_1 (res. Θ_0) denotes the random transcript variable for $\mathscr{A}$'s interaction with $\mathcal{O}_1$ (res. $\mathcal{O}_0$). The *interpolation probability* of ω with regard to $\mathcal{O}$ is the probability of obtaining a given transcript ω in the security game with an oracle $\mathcal{O}$. Since $\mathscr{A}$ is deterministic, this probability only depends on the transcript ω and the oracle $\mathcal{O}$. A transcript ω is said to be *attainable* if $\Pr[\Theta_0 = \omega] > 0$. We now state the coefficient H technique(or simply the H-technique), a simple yet powerful tool developed by Patarin [94] in form of a theorem. A proof of this theorem can be found in a number of papers including [95, 42, 86].

Theorem 2 (H-technique [94, 95])**.** *Let Ω be the set of all transcripts that are attainable. For some $\epsilon_{\mathsf{bad}}, \epsilon_{\mathsf{ratio}} > 0$, suppose there is a set $\Omega_{\mathsf{bad}} \subseteq \Omega$ satisfying the following:*

- $\Pr[\Theta_0 \in \Omega_{\mathsf{bad}}] \leq \epsilon_{\mathsf{bad}}$;
- *For any $\omega \notin \Omega_{\mathsf{bad}}$, ω is attainable and*

$$\frac{\Pr[\Theta_1 = \omega]}{\Pr[\Theta_0 = \omega]} \geq 1 - \epsilon_{\mathsf{ratio}}.$$

Then the distinguishing advantage for any adversary $\mathscr{A}$ can be bounded as

$$\mathbf{Adv}_{\mathcal{O}_1}(\mathscr{A}) \leq \epsilon_{\mathsf{bad}} + \epsilon_{\mathsf{ratio}}.$$

Chapter 3

Multi-Chain Graphs

3.1 Introduction

In this chapter, we consider a security game that we call a multi-chain security game. In this game, an adversary $\mathscr{A}$ interacts with a random permutation and its inverse. Its goal is to construct multiple walks having the same labels. We first need to describe some notations which would be required to define the security game.

3.2 Graph Structure and Multi-chain

Let $\mathcal{L} = ((u_1, v_1), \ldots, (u_t, v_t))$ be a list of pairs of b-bit elements such that $u_1, \ldots u_t$ are distinct and $v_1, \ldots, v_t$ are distinct. For any such list of pairs, we write $\mathsf{domain}(\mathcal{L}) = \{u_1, \ldots, u_t\}$ and $\mathsf{range}(\mathcal{L}) = \{v_1, \ldots, v_t\}$.

Let L be a linear function over b bits. Given such a list we define a labeled directed graph $\mathcal{G}^L_{\mathcal{L}}$ which we call the chain graph over the set of vertices $\mathsf{range}(\mathcal{L}) \subseteq \{0,1\}^b$ as follows: A directed edge $v_i \to v_j$ with label x (also denoted as $v_i \xrightarrow{x} v_j$) is in the graph if $L(v_i) \oplus x = u_j$. We can similarly extend this to a label walk $\mathcal{W}$ from a node w_0 to w_k as

$$\mathcal{W} : w_0 \xrightarrow{x_1} w_1 \xrightarrow{x_2} w_2 \cdots \xrightarrow{x_k} w_k.$$

We simply denote it as $w_0 \xrightarrow{x} w_k$ where $x = (x_1, \ldots, x_k)$. Here k is the length of the walk. We simply denote the directed chain graph $\mathcal{G}^L_{\mathcal{L}}$ by $\mathcal{G}_{\mathcal{L}}$ wherever the linear function L is understood from the context.

Definition 2. *Let L be a fixed linear function over b bits. Let $r, \tau \leq b$ be some parameters. We say that a set of labeled walks $\{\mathcal{W}_1, \ldots, \mathcal{W}_p\}$ forms a* multi-chain with

a label $x := (x_1, \ldots, x_k)$ *in the graph* $\mathcal{G}_{\mathcal{L}}$ *if for all* $1 \leq i \leq p$, $\mathcal{W}_i : v_0^i \xrightarrow{x} v_k^i$ *and* $\lceil u_0^1 \rceil_r = \cdots = \lceil u_0^p \rceil_r$ *and* $\lceil v_k^1 \rceil_\tau = \cdots = \lceil v_k^p \rceil_\tau$. *We also say that the multi-chain is of length* k. *The labeled walks* $\mathcal{W}_i$ *are also called chains in this context.*

Note that if $\{\mathcal{W}_1, \ldots, \mathcal{W}_p\}$ is a multi-chain then so is any subset of it. Also, there can be a different set of multi-chains depending on the starting and ending vertices and different $x = (x_1, \ldots, x_k)$. Let W_k denote the maximum order of all such multi-chains of length k. For a fixed linear function L, W_k is completely determined by $\mathcal{L}$. Now we describe how the list $\mathcal{L}$ is being generated through an interaction of an adversary $\mathcal{A}$ and a random permutation.

3.2.1 Multi-Chain Security Game

Consider an adversary $\mathscr{A}$ interacting with a b-bit random permutation $\Pi^{\pm}$. Suppose, the adversary $\mathscr{A}$ makes at most t interactions with $\Pi^{\pm}$. Let (x_i, dir_i) denote ith query where $x_i \in \{0,1\}^b$ and dir_i is either $+$ or $-$ (representing forward or inverse query). If $\mathsf{dir}_i = +$, it gets response y_i as $\Pi(x_i)$, else the response y_i is set as $\Pi^{-1}(x_i)$. After t interactions, we define a list $\mathcal{L}$ of pairs (u_i, v_i) where $(u_i, v_i) = (x_i, y_i)$ if $\mathsf{dir}_i = +$, and $(u_i, v_i) = (y_i, x_i)$ otherwise. So we have $\Pi(u_i) = v_i$ for all i. We call the tuple of triples $\theta := ((u_1, v_1, \mathsf{dir}_1), \ldots, (u_t, v_t, \mathsf{dir}_t))$ the transcript of the adversary $\mathscr{A}$ interacting with $\Pi^{\pm}$. We also write $\theta' = ((u_1, v_1), \ldots, (u_t, v_t))$ which only stores the information about the random permutation. For the sake of simplicity we assume that the adversary makes no redundant queries and so all $u_1, \ldots u_t$ are distinct and $v_1, \ldots, v_t$ are distinct. For a linear function L consider the directed chain graph $\mathcal{G}_{\theta'}$. For any k, we have already

defined W_k. Now we define the maximum multi-chain advantage as

$$\mu_t = \max_{\mathscr{A}} \max_{k} \mathsf{Ex}\left[\frac{\mathsf{W}_k}{k}\right].$$

3.2.2 Bounding μ_t for Invertible L Functions

In this subsection, we derive concrete bounds for μ_t under a special assumption that the underlying linear function is invertible.

Theorem 3. *If the linear function L is invertible, then we have*

$$\mu_t \leq \mathsf{mcoll}(t, 2^\tau) + \mathsf{mcoll}(t, 2^r) + \mathsf{mcoll}'(t^2, 2^b). \tag{3.1}$$

Proof of Theorem 3: We first make the following observation which is straightforward as L is invertible.

OBSERVATION 1: If $v_i \xrightarrow{x} v_k$ and $v_j \xrightarrow{x} v_k$ then $v_i = v_j$.

We now describe some more notations related to multi-chains:

1. Let $\mathsf{W}^{\mathsf{fwd},a}$ denote the size of the set $\{i : \mathsf{dir}_i = +, \lceil v_i \rceil_\tau = a\}$ and $\max_a \mathsf{W}^{\mathsf{fwd},a}$ is denoted as $\mathsf{W}^{\mathsf{fwd}}$. This denotes the maximum multi-collision among τ most significant bits of forward query responses.

2. Similarly, we define the multi-collision for backward query responses as follows: Let $\mathsf{W}^{\mathsf{bck},a}$ denote the size of the set $\{i : \mathsf{dir}_i = -, \lceil u_i \rceil_r = a\}$ and $\max_a \mathsf{W}^{\mathsf{bck},a}$ is denoted as $\mathsf{W}^{\mathsf{bck}}$.

3. In addition to the multicollisions in forward only and backward only queries, we

consider multicollisions due to both forward and backward queries. Let $\mathsf{W}^{\mathsf{mitm},a}$ denote size of the set $\{(i,j) : \mathsf{dir}_i = +,\ \mathsf{dir}_j = -,\ L(v_i) \oplus u_j = a\}$ and $\max_a \mathsf{W}^{\mathsf{mitm},a}$ is denoted as $\mathsf{W}^{\mathsf{mitm}}$.

Now, we state an intermediate result which is the main step of the proof.

Lemma 1. *For all possible interactions, we have*

$$\mathsf{W}_k \leq \mathsf{W}^{\mathsf{fwd}} + \mathsf{W}^{\mathsf{bck}} + k \cdot \mathsf{W}^{\mathsf{mitm}}.$$

Proof. We can divide the set of multi-chains into three sets:

Forward-only chains: Each chain is constructed by Π queries only. By definition, the size of such multi-chain is at most $\mathsf{W}^{\mathsf{fwd}}$.

Backward-only chains: Each chain is constructed by Π^{-} queries only. By definition, the size of such multi-chain is at most $\mathsf{W}^{\mathsf{bck}}$.

Forward-backward chains: Each chain is constructed by using both Π and Π^{-} queries. Let us denote the size of such multi-chain by $\mathsf{W}_k^{\mathsf{fwd\text{-}bck}}$.

Then, we must have

$$\mathsf{W}_k \leq \mathsf{W}^{\mathsf{fwd}} + \mathsf{W}^{\mathsf{bck}} + \mathsf{W}_k^{\mathsf{fwd\text{-}bck}}.$$

Now, we claim that $\mathsf{W}_k^{\mathsf{fwd\text{-}bck}} \leq k \cdot \mathsf{W}^{\mathsf{mitm}}$. Suppose $\mathsf{W}_k^{\mathsf{fwd\text{-}bck}} = w$. Then, it is sufficient to show that there exist an index $j \in [k]$, such that the size of the set $\{i : (\mathsf{dir}_{j-1}^i, \mathsf{dir}_j^i) \in \{(+,-),(-,+)\},\ L(v_{j-1}^i) \oplus u_j^i = x_j\} \geq \lceil w/k \rceil$. This can be easily argued by pigeonhole principle, given Observation 1. The argument works as follows:

For each of the individual chain W_i, we have at least one index $j \in [k]$ such that $(\mathsf{dir}^i_{j-1}, \mathsf{dir}^i_j) \in \{(+,-),(-,+)\}$. We put the i-th chain in a bucket labeled j, if $(\mathsf{dir}^i_{j-1}, \mathsf{dir}^i_j) \in \{(+,-),(-,+)\}$. Note that, it is possible that the i-th chain can co-exist in multiple buckets. But more importantly, it will exist in at least one bucket. As there are k buckets and w chains, by pigeonhole principle, we must have one bucket $j \in [k]$, such that it holds at least $\lceil w/k \rceil$ chains. □

Now we complete the proof of Theorem 3. Observe that $\mathsf{W}^{\mathsf{fwd}}$ and $\mathsf{W}^{\mathsf{bck}}$ are the random variables corresponding to the maximum multicollision in a truncated random permutation sample of size t, and corresponds to Remark 1 of subsection 2.3.1. Further, if we denote $x_i := u_i$ and $y_i := L(v_i)\ \forall i \in [t]$ then using Observation 1, $\mathsf{W}^{\mathsf{mitm}}$ is the random variable corresponding to the maximum multicollision in a sum of random permutation sample of size t^2, i.e., the special distribution in subsection 2.3.2. Now, using linearity of expectation, we have

$$\begin{aligned}\mu_t &\leq \mathsf{Ex}\left[\mathsf{W}^{\mathsf{fwd}}\right] + \mathsf{Ex}\left[\mathsf{W}^{\mathsf{bck}}\right] + \mathsf{Ex}\left[\mathsf{W}^{\mathsf{mitm}}\right] \\ &\leq \mathsf{mcoll}(t, 2^\tau) + \mathsf{mcoll}(t, 2^r) + \mathsf{mcoll}'(t^2, 2^b).\end{aligned}$$

3.2.3 Bounding μ_t for Non-invertible L Functions

In the case of invertible functions, Observation 1 facilitates a fairly simple argument in favor of an upper bound on μ_t in terms of some multicollision sizes. However, the same observation does not apply to non-invertible functions. Specifically, Lemma 1 is not guaranteed to hold. For example, now the adversary can try to create a binary tree-like structure using forward queries only. We have to accommodate such attack strategies

to upper bound μ_t.

Let $\texttt{Collapse}$ denote the event that there exists distinct i and j, such that $\mathsf{dir}_i = \mathsf{dir}_j = +$, and $L(y_i) = L(y_j)$. We say that a transcript $\mathcal{L}$ is *collapse-free* if the event $\neg\texttt{Collapse}$ holds. The following result is a variant of lemma 1 for collapse-free transcripts.

Lemma 2. *For all possible collapse-free transcripts, we have*

$$\mathsf{W}_k \leq \mathsf{W}^{\mathsf{fwd}} + \mathsf{W}^{\mathsf{bck}} + k \cdot \mathsf{W}^{\mathsf{mitm}}.$$

Proof. We can again divide the set of multi-chains into three sets:

Forward-only chains: Each chain is constructed by Π queries only. We collect all such chains into a list FWD.

Backward-only chains: Each chain is constructed by Π^- queries only. We collect all such chains into list BCK.

Forward-backward chains: Each chain is constructed by using both Π and Π^- queries.

For the set of forward-backward chains, consider the smallest index j such that for two[1] distinct chains $\mathcal{W}_i$ and $\mathcal{W}_{i'}$ we have $v^i_j = v^{i'}_j$, i.e. the two chains merge. Since, the transcript is collapse-free, we must have $\mathsf{dir}^i_{j-1} = -$ or $\mathsf{dir}^{i'}_{j-1} = -$, or both. Now, we may have two cases:

[1]We may have more than two distinct chains merging at the same index. For brevity we consider only two. The general case can be handled in exactly the same manner.

1. Without loss of generality assume that only $\mathsf{dir}^i_{j-1} = -$. Now, if we traverse back along the walk $\mathcal{W}_i$ from vertex v^i_j, then either we get all backward edges (i.e. $\mathsf{dir}^i_{j'} = -$ for all $j' < j$), or there exists a $j' < j$ such that $\mathsf{dir}^i_{j'} = +$ and $\mathsf{dir}^i_{j'+1} = -$. In the first case we insert $\mathcal{W}_i$ in BCK, and in the second case we collect $\mathcal{W}_i$ in list MITM.

2. Suppose $\mathsf{dir}^i_{j-1} = -$ and $\mathsf{dir}^{i'}_{j-1} = -$. In this case, we traverse both $\mathcal{W}_i$ and $\mathcal{W}_{i'}$ and collect them in either BCK or MITM using the preceding argumentation.

We follow similar approach for all indices (in increasing order) where two or more chains merge collecting chains in either BCK or MITM. Once, we have exhausted all merging indices, we are left with some uncollected chains. We claim that these chains are disjoint of each other. This is easy to argue as for any pair of merged chains the chains with backward edge are already collected before. So all that is remaining is a collection of disjoint chains. Further, each of these chains must contain an index j such that $(\mathsf{dir}_j, \mathsf{dir}_{j+1}) \in \{(+,-),(-,+)\}$. We collect all these remaining chains in the list MITM'. Thus, we have

$$\mathsf{W}_k = |\mathsf{FWD}| + |\mathsf{BCK}| + |\mathsf{MITM}| + |\mathsf{MITM}'|.$$

By using the collapse-free property of $\mathcal{L}$ we get $|\mathsf{FWD}| \leq \mathsf{W}^{\mathsf{fwd}}$, and $|\mathsf{BCK}| \leq \mathsf{W}^{\mathsf{bck}}$ by definition. Further, by using the Pigeonhole argument used in the proof of Lemma 1, we get $|\mathsf{MITM}| + |\mathsf{MITM}'| \leq k \cdot \mathsf{W}^{\mathsf{mitm}}$. □

Finally, we get the following upper bound on μ_t for non-invertible L functions.

Theorem 4. *If the linear function L is non-invertible and $\mathcal{L}$ is collapse-free, then we*

have

$$\mu_t \leq \mathsf{mcoll}(t, 2^\tau) + \mathsf{mcoll}(t, 2^r) + \widetilde{\mathsf{mcoll}}(t^2, 2^b, \mathsf{rank}(L)). \tag{3.2}$$

Proof. As before $\mathsf{Ex}\left[\mathsf{W}^{\mathsf{fwd}}\right] \leq \mathsf{mcoll}(t, 2^r)$, $\mathsf{Ex}\left[\mathsf{W}^{\mathsf{bck}}\right] \leq \mathsf{mcoll}(t, 2^\tau)$. Further, $\mathsf{W}^{\mathsf{mitm}}$ is the multicollision random variable $\widetilde{mc}_{t^2,2^b,\mathsf{rank}(L)}$ defined in subsection 2.3.3. Thus, $\mathsf{Ex}\left[\mathsf{W}^{\mathsf{mitm}}\right] \leq \widetilde{\mathsf{mcoll}}(t^2, 2^b, \mathsf{rank}(L))$. The result follows from linearity of expectation. □

Remark 3. *Theorem 4 has a limited applicability. Specifically, it holds only when $\mathcal{L}$ is collapse-free. A straightforward upper bound on* $\Pr[\texttt{Collapse}]$ *is $t^2/2^{\mathsf{rank}(L)}$, where t denotes the size of $\mathcal{L}$ and $\mathsf{rank}(L)$ denotes the rank of linear function L. At times this bound is weaker than the bound achievable from a more straightforward approach of using the loose upper bound of $\mu_t \leq t$.*

3.3 Multi-chain Security Game with Extra State

In this section, we re-define the multi-chain graph structure to incorporate an extra state of size c bits and call it a *c-extended* multi-chain graph. Then we define an adversary $\mathscr{A}$ which interacts with a random permutation and its inverse. Its goal is to construct *c-extended* multi-chains. We start by describing a few notations which would be required to define the security game.

Labeled Walk: Let $\mathcal{L} = ((u_1, v_1), \ldots, (u_t, v_t))$ be a list of pairs of b-bit elements such that $u_1, \ldots, u_t$ are distinct and $v_1, \ldots, v_t$ are distinct. For any such list of pairs, we write $\mathsf{domain}(\mathcal{L}) = \{u_1, \ldots, u_t\}$ and $\mathsf{range}(\mathcal{L}) = \{v_1, \ldots, v_t\}$.

Let L be a linear function over $b + c$ bits with the transformation matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$.

Given such a list we define a labeled directed graph $G^L_{\mathcal{L}}$ over the set of vertices $\mathsf{range}(\mathcal{L}) \times \{0,1\}^c \subseteq \{0,1\}^{b+c}$ as follows: A directed edge $(v_i, s_i) \to (v_j, s_j)$ with label x (also denoted as $(v_i, s_i) \xrightarrow{x} (v_j, s_j)$) is in the graph if $L((v_i, s_i)) \oplus (x, 0^c) = (u_j, s_j)$. We call this a *c-extended multi-chain graph*. We can similarly extend this to a label walk $\mathcal{W}$ from a node (w_0, s_0) to (w_k, s_k) as

$$\mathcal{W} : (w_0, s_0) \xrightarrow{x_1} (w_1, s_1) \xrightarrow{x_2} (w_2, s_2) \cdots \xrightarrow{x_k} (w_k, s_k).$$

We simply denote it as $(w_0, s_0) \xrightarrow{x} (w_k, s_k)$ where $x = (x_1, \ldots, x_k)$. Here k is the length of the walk. We simply denote the directed graph $G^L_{\mathcal{L}}$ by $G_{\mathcal{L}}$ wherever the linear function L is understood from the context.

Definition 3. *Let G be a c-extended multi-chain graph as defined above. given any fixed level $(x_1, \ldots, x_l)$, we say the set of l length walks $\{\mathcal{W}_i : (u^i_0, s^i_0) \xrightarrow{(x_1,\ldots,x_l)} (u^i_l, s^i_l)\}$ form a c-extended multi-chain if and only if $u^i_l = u^j_l$ for all $i \neq j$.*

Notice that if $\mathcal{W}$ is a *c-extended* multi-chain then so is any subset of $\mathcal{W}$. Consider the set of all *c-extended* multi-chains in G of length k. Let W_k denote the size of the largest of all such *c-extended* multi-chains of length k.

3.3.1 Adversarial Game

Consider an adversary $\mathscr{A}$ interacting with a random permutation Π. Let the query transcript be of the form $\Theta = \{(U_i, W_i, \mathsf{dir}_i)\}_{i,q_p}$ where $\mathsf{dir}_i = +$ if the i th query is a forward one and $\mathsf{dir}_i = -$ otherwise. Consider a linear function $L = \{0,1\}^{b+c} \to \{0,1\}^{b+c}$ and the graph G^L_Θ.

Define

$$\mu_{q_p} := \max_{k>0} \mathsf{Ex}\left[\frac{|W_k|}{k}\right].$$

Our objective is to upper bound μ_{q_p} in G^L_Θ i.e. to restrict the adversary $\mathscr{A}$'s ability to generate *c-extended* multi-chains. To bound μ_{q_p}, we first define an event `FBAD`.

`FBAD` : Let S denote the set of all pairs of vertices in G_Θ such that there is a collision in $\mathsf{range}(\Theta)$. In notation,

$$S : \{\{(v_i, s_1), (v_j, s_2)\} \mid v_i, v_j \in \mathsf{range}(\Theta);\ s_1, s_2 \in \{0,1\}^s;\ L(v_1, s_1) = L(v_2, s_2)\}$$

define `FBAD` to be the event that $|S| > n$.

Proposition 7. $\mathsf{Ex}\left[|S|\right] \leq \frac{T^2}{2^{\mathsf{rank}(L)-2c+1}}$.

Proof. For each $i \neq i' \in [1, T]$ define

$$I_{i,i'} = \begin{cases} 1 \text{ if } \{(V_i, s_1), (V_j, s_2)\} \in S \\ 0 \text{ otherwise.} \end{cases}$$

Then

$$\begin{aligned}
\mathsf{Ex}\left[|S|\right] &= \mathsf{Ex}\left[\sum_{\substack{V_i,V_j\in\mathsf{range}(\Theta)\\ s_1,s_2\in\{0,1\}^c}} I_{\{i,i'\}}\right] \\
&= \sum_{\substack{V_i,V_j\in\mathsf{range}(\Theta)\\ s_1,s_2\in\{0,1\}^c}} \mathsf{Ex}\left[I_{\{i,i'\}}\right] \\
&\leq \sum_{\substack{V_i,V_j\in\mathsf{range}(\Theta)\\ s_1,s_2\in\{0,1\}^c}} \Pr\left[L(V_i,s_1)=L(V_j,s_2)\right] \\
&\leq \frac{T^2 2^{2c}}{2^{\mathsf{rank}(L)+1}}.
\end{aligned}$$

□

Corollary 2.

$$\Pr\left[\texttt{FBAD}\right] \leq \frac{T^2}{n2^{\mathsf{rank}(L)-2c+1}}.$$

Proof. This follows from Proposition 3 and Proposition 7. □

Proposition 8. *Consider the c-extended multi-chain graph G^L_Θ. If event* FBAD *does not hold, then*

$$\mu_{q_p} \leq n + 2^c.$$

Proof. We proceed through induction on k to show $W_k \leq nk + 2^c$. The proposition follows from the definition of μ_{q_p}.

Given any $(x_1,\ldots,x_k) \in \{0,1\}^{bk} and (v,s) \in V(G^L_\Theta)$ consider the set of walks $\mathcal{M}^k_{(v,s)} := \{(u_i,s_i) \overset{(x_1,\ldots,x_k)}{\longrightarrow} (v,s) | u_i \in V(G^L_\Theta)\}$. Clearly every *c-extended* multi-chain

of length k is a subset of some $\bigcup_{s\in\{0,1\}^c} \mathcal{M}^k_{(v,s)}$. Hence

$$W_k \leq \max_{v\in \mathsf{range}(\Theta)} \sum_{s\in\{0,1\}^c} |\mathcal{M}^k_{(v,s)}|.$$

First suppose $k = 1$. Note that an edge $(u_i, s_i) \xrightarrow{x} (v, s) \in M^1_{v,s}$ if and only if

$$L \cdot \begin{bmatrix} u_i \\ s_i \end{bmatrix} \oplus \begin{bmatrix} x \\ 0 \end{bmatrix} = \begin{bmatrix} v \\ s \end{bmatrix} \tag{3.3}$$

Since `FBAD` doesn't occur, for any $v \in \mathsf{range}(\Theta)$, $\sum_{\substack{s\in\{0,1\}^c \\ |\mathcal{M}_{(v,s)}|>1}} (|\mathcal{M}^1_{(v,s)}| - 1) \leq n$.

Hence

$$W_1 \leq \sum_{s\in\{0,1\}^c} 1 + \sum_{\substack{s\in\{0,1\}^c \\ |\mathcal{M}^1_{(v,s)}|>1}} (|\mathcal{M}^1_{(v,s)}| - 1) \leq n + 2^c.$$

Now suppose $W_{k-1} \leq n(k-1)+2^c$. Given $(v, s) \in V(G^L_\Theta)$ for every walk $(u_i, s_i) \xrightarrow{(x_1,\ldots,x_k)} (v, s)$ there exist an unique $(w_i, t_i) \in V(G^L_\Theta)$ such that $(u_i, s_i) \xrightarrow{x_1} (w_i, t_i)$ and $(w_i, t_i) \xrightarrow{(x_2,\ldots,x_k)} (v, s)$. Hence,

$$\begin{aligned} W_k &\leq W_{k-1} + \sum_{\substack{(w_i,t_i)\in V(G^L_\Theta) \\ |\mathcal{M}^1_{(w_i,t_i)}|>1}} (|\mathcal{M}^1_{(w_i,t_i)}| - 1) \\ &\leq W_{k-1} + n && \text{[Since } \texttt{FBAD} \text{ does not hold.]} \\ &\leq kn + 2^c. \end{aligned}$$

□

Corollary 3. *Consider the c-extended multi-chain graph G^L_Θ. If L is invertible, then*

$$\mu_{q_p} \leq 2^c.$$

Proof. Note that if L is invertible then $|S| = 0$ and as a consequence FBAD does not occur. Hence we can take $n = 0$. □

3.4 Related Work

In [85] Mennink analyzed the Key-prediction security of Keyed Sponge using a special type of data structure that is close to but different from our chain graph structure. Here we give a brief overview of Mennink's work in our notations and describe how our structure is different from the structure considered by him.

Let $\mathcal{L} = ((u_1, v_1), \ldots, (u_t, v_t))$ be a list of pairs of b-bit elements such that $u_1, \ldots u_t$ are distinct and $v_1, \ldots, v_t$ are distinct. Let $c < b$ be any positive integer. For any such list of pairs, we write $\mathsf{domain}(\mathcal{L}) = \{u_1, \ldots, u_t\}$ and $\mathsf{range}(\mathcal{L}) = \{v_1, \ldots, v_t\}$. Given such a list we define a labeled directed graph $\mathcal{G}_\mathcal{L}$ over the set of vertices $\mathsf{range}(\mathcal{L}) \subseteq \{0,1\}^b$ as follows: A directed edge $v_i \to v_j$ with label x (also denoted as $v_i \xrightarrow{x} v_j$) is in the graph if $v_i \oplus x \| 0^c = u_j$. We can similarly extend this to a label walk $\mathcal{W}$ from a node w_0 to w_k as

$$\mathcal{W} : w_0 \xrightarrow{x_1} w_1 \xrightarrow{x_2} w_2 \cdots \xrightarrow{x_k} w_k.$$

We simply denote it as $w_0 \xrightarrow{x} w_k$ where $x = (x_1, \ldots, x_k)$. Here k is the length of the walk. The set $yield_{c,k}(\mathcal{L})$ consists of all possible labels x such that there exists a

k-length walk of the form $0^b \xrightarrow{x} w_k$ in the graph $\mathcal{G}_{\mathcal{L}}$.

Consider the graph, $\mathcal{G}_{\mathcal{L}}$. The configuration of a walk from w_0 to w_k is defined as a tuple $C = (C_1, \ldots, C_k) \in \{0,1\}^k$ where $C_i = 0$ if $w_{i-1} \xrightarrow{x_i} w_i$ comes from a forward primitive query and $C_i = 1$ if it corresponds to an inverse primitive query.

Mennink provided an upper bound of $yield_{c,k}(\mathcal{L})$ by bounding the maximum number of possible labeled walks from 0^b to any given $w_k \in \{0,1\}^b$ with a given configuration C.

The use of tools like multi-collision and the similarity in the data structure of [85] with our multi-chain structure can be misleading. Here we try to discuss the difference between them and show that the underlying motivation behind both the problems is philosophically as different as possible.

Note that using a multi-chain structure, we try to bound the number of different walks with the same label and distinct starting points whereas $yield_{c,k}(\mathcal{L})$ is the number of different walks with the same starting point namely 0^b and distinct labels. Hence the multi-chain structure deals with a different problem than $yield_{c,k}(\mathcal{L})$. A notable change in our work is to deal with multicollision of the sum of two permutation calls (we call it meet in the middle multicollision, see definition of $\mathsf{W}^{\mathsf{mitm}}$). This computation is not straightforward like usual computation of expectation of multi-collision (see Subsection 2.3.2).

Chapter 4

Transform-then-Permute: Design and Analysis

4.1 Introduction

In this chapter, we study a Sponge-type AEAD construction called Transform-then-Permute (or TtP) with a generalization of the feedback function used in the duplexing interface. We give tight security bound for the special case when the feedback function is invertible. In Section 4.2 we define the Transform-then-Permute construction in details. In Section 4.3, using the multi-chain security game from Section 3.1 we give a complete security proof of the AEAD security bound given in Theorem 5. In Section 4.4, we show that the TtP generalization encompasses the feedback functions used in Sponge AE, Beetle, SpoC etc. Particularly, Beetle and SpoC modes fall under the class where the feedback functions are invertible, and hence for those modes, optimal AEAD security is achieved. In Section 4.5 we extend our result for the general Sponge duplex construction. Finally, in Section 4.6, we give some attack strategies to justify the tightness of our bounds.

4.2 Transform-then-Permute Construction

In this section, we describe the Transform-then-Permute (or TtP in short) construction in detail.

4.2.1 Parameters and Components

We first describe some parameters of our wide family of AEAD algorithms.

1. State-size: The underlying primitive of the construction is a b-bit public permutation. We call b the state size of the permutation.
2. Key-size: Let κ denote the key-size. Here we assume $\kappa < b$.
3. Nonce-size: In this thesis, we consider fixed size nonce. Let ν denote the size of nonce.
4. Rate: Let $r, r' \leq b$ denote the rate of processing message and associate data respectively. The capacity is defined as $c := b - r$.

Let $\mathbb{N}_0$ be the set of all non-negative integers and $\theta := b - \kappa - \nu$. For $x \in \mathbb{N}_0$, we define

$$a(x) := \begin{cases} 0 & \textit{if } x \leq \theta \\ \lceil \frac{x-\theta}{r'} \rceil & \textit{otherwise} \end{cases}$$

Parsing Function: Let $D = N \| A$ where $N \in \{0,1\}^\nu$ and $A \in \{0,1\}^*$ with $a := a(|A|)$.

– **Case** $|A| \leq \theta$: $\mathsf{parse}(N, A) = D \parallel 0^{\theta - |A|} \in \{0,1\}^{b-\kappa}$.

– **Case** $|A| > \theta$: $\mathsf{parse}(N, A) := (IV, A_1, \ldots, A_a)$ where $D = IV \| D'$, $IV \in \{0,1\}^{b-\kappa}$ and $(A_1, \ldots, A_a) \xleftarrow{r'} D'$. Note that $|D'| = |A| - \theta$ and so when we parse D' to blocks of size r', we get $a(|A|) = \lceil \frac{|A|-\theta}{r'} \rceil$ many blocks.

In addition to parsing $N \| A$, we also parse a message or ciphertext Z as $(Z_1, \ldots, Z_m) \xleftarrow{r} Z$ into m blocks of size r where $m = \lceil |Z|/r \rceil$.

We define $t := a + m$ to be the total number of blocks corresponding to an input query of the form (N, A, Z).

Domain Separation: To every pair of non-negative integers $(|A|, |Z|)$ with $a = a(|A|)$, $m = \lceil |Z|/r \rceil$, and for every $0 \leq i \leq a + m$, we associate a small integer δ_i where

$$\delta_i = \begin{cases} 0 & \textit{if } i \notin \{a\} \cup \{t\} \\ 1 & \textit{if } (i = a \wedge r' \mid |A| - \theta) \vee (i = t \wedge r \mid |M|) \\ 2 & \textit{otherwise.} \end{cases}$$

We collect all these δ values through the following function $\mathsf{DS}(|A|, |Z|) = (\delta_0, \delta_1, \ldots, \delta_{a+m})$.

Encoding Function: Let $\mathcal{D}_{DS} := \{0,1\}^2 \times \{0,1,2\}$ and $r_{\max} = \max\{r, r'\}$. Let

$$\mathsf{encode} : \{0,1\}^{\leq r_{\max}} \times \mathcal{D}_{DS} \to \{0,1\}^b$$

be an injective function such that for any $D, D' \in \{0,1\}^x$, $1 \leq x \leq r_{\max}$ and for all $\Delta \in \mathcal{D}_{DS}$, we have $\mathsf{encode}(D, \Delta) \oplus \mathsf{encode}(D', \Delta) = 0^{b-x} \| (D \oplus D')$. Actual description

of this encode function is determined by the construction.

FORMAT FUNCTION: We define a formatting function Fmt which maps a triple (N, A, M) to $(D_0, \ldots, D_{a+m}) \in (\{0,1\}^b)^{a+m+1}$ where $a := a(|A|)$ and $m = \lceil |Z|/r \rceil$. The exact description of format function is described in Algorithm 1.

Algorithm 1 Description of the format function (Fmt)

function FMT(N, A, Z)
  $a \leftarrow a(|A|)$, $m \leftarrow \lceil |Z|/r \rceil$
  $(A_0, A_1, \ldots, A_a) \leftarrow \mathsf{Parse}(N, A)$
  $(Z_1, \ldots, Z_m) \xleftarrow{r} Z$
  $(\delta_0, \ldots, \delta_t) \leftarrow \mathsf{DS}(|A|, |Z|)$
  for $i = 0$ to a **do**
    if $i = a$ and $m = 0$ **then**
      $D_i \leftarrow \mathsf{encode}(A_i, (0, 1, \delta_i))$
    else
      $D_i \leftarrow \mathsf{encode}(A_i, (0, 0, \delta_i))$
  for $i = 1$ to m **do**
    $D_{a+i} \leftarrow \mathsf{encode}(Z_i, (1, 0, \delta_{i+m}))$
  return $(D_0, \ldots, D_t)$

Lemma 3. *Given any two tuples* $(N, A, Z) \neq (N', A', Z')$ *and* $\mathsf{Fmt}(N, A, Z) = (D_0, \ldots, D_t)$ *and* $\mathsf{Fmt}(N', A', Z') = (D'_0, \ldots, D'_{a'+m'})$*, we have*

1. $(D'_0, \ldots, D'_a) \neq (D_0, \ldots, D_a)$ *whenever* $(N, A) \neq (N', A')$ *and* $a \leq a'$.
2. $(D'_a, \ldots, D'_t) \neq (D_a, \ldots, D_t)$ *whenever* $(N, A) = (N', A')$ *and* $m \leq m'$.

Proof. We write $\mathsf{parse}(N, A) = (A_0, A_1, \ldots, A_a)$ and $\mathsf{parse}(N', A') = (A'_0, A'_1, \ldots, A'_{a'})$.

1. Let $(N, A) \neq (N', A')$. Then we have $(A_0, A_1, \ldots, A_a) \neq (A'_0, A'_1, \ldots, A'_{a'})$. Now if, $a < a'$ then we have $D_a = \mathsf{encode}(A_a, 0, \delta)$ where $\delta \in \{1, 2\}$ and $D'_a =$

$\mathsf{encode}(A'_a, 0, 0)$. Hence by injectivity of encode we have $D_a \neq D'_a$. If $a = a'$ then there exists non-negative $i \leq a$ such that $A_i \neq A'_i$ and hence $D_i \neq D'_i$.

2. Let $(N, A) = (N, A')$. Then we have $(A_0, A_i, \ldots, A_a) = (A'_0, A'_i, \ldots, A'_a)$. Note that m, m' both cannot be 0. So if $m = 0$, then $m' > 0 \implies D_a = \mathsf{encode}(A_a, 0, \delta)$ for some $\delta \in \{1, 2\}$ and $D'_a = \mathsf{encode}(A_a, 0, 0)$. Hence $D_a \neq D'_a$. Let $m, m' > 0$ then if, $m < m'$ then we have $D_t = \mathsf{encode}(M_m, 1, \delta)$ where $\delta \in \{1, 2\}$ and $D'_a = \mathsf{encode}(M'_m, 1, 0)$. Else if $m = m'$, then there exists positive $i \leq m$ such that $M_i \neq M'_i$. Hence $D_{a+i} \neq D'_{a+i}$.

□

FEEDBACK FUNCTIONS: We also need some linear functions $L_{ad}, L_e : \{0,1\}^b \to \{0,1\}^b$ which are used to process associate data and message respectively in an encryption algorithm.

Now, given a linear function $L : \{0,1\}^b \to \{0,1\}^b$, $1 \leq x \leq r$, the following function $L' : \{0,1\}^b \times \{0,1\}^x \times \mathcal{D}_{DS} \to \{0,1\}^b \times \{0,1\}^x$, is used to process the j-th block Z (either a plaintext or a ciphertext) using the output Y of the previous invocation of the random permutation:

$$L'(Y, Z, \Delta) = (X := L(Y) \ \oplus \mathsf{encode}(Z, \Delta),\ Z' := \lceil Y \rceil_{|Z|} \ \oplus Z)$$

For $1 \leq i \leq r$, let $L_{d,i}(x) := L_e(x) \oplus \left(0^{b-i} \| \lceil x \rceil_i\right)$. Then, it is easy to see from the property of encoding function that $L'_{d,|C|}(Y, C, \Delta) = (X, C)$ if and only if $L'_e(Y, M, \Delta) = (X, C)$. Figure 4-1 provides an illustration how a message block is processed.

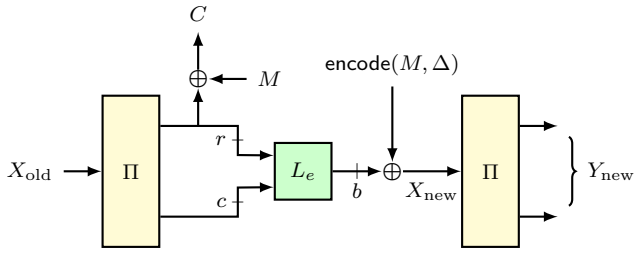


Figure 4-1: Illustration of the feedback process for a message block M of $|M|$ bits. Here $\mathsf{encode}(M, \Delta)$ represents some encoding of $|M|$ bits string to a b-bit string as described above and L_e is a linear transformation applied on b-bit strings.

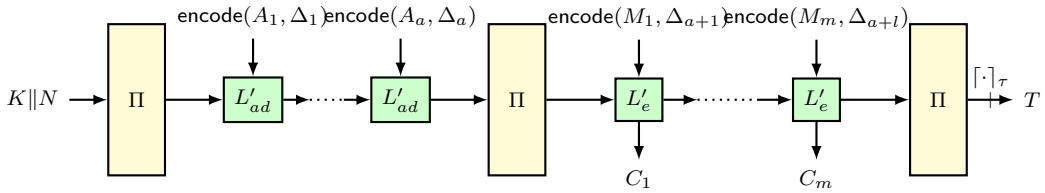


Figure 4-2: Schematic of the Transform-then-Permute AEAD mode. Here we assume $|N| = b - \kappa$, $L'_{ad}(Y, A) = L_{ad}(Y) \oplus A$. L_{ad}, L'_e, encode functions and Δ values are as described before.

4.2.2 Description of the Transform-then-Permute AEAD

We describe the Transform-then-Permute construction in Algorithm 2 which generalizes duplexing method used in sponge-type AEADs. Figure 4-2 illustrates a simple case when $|N| = b - \kappa$.

4.2.3 Security Analysis of TtP

We prove the following result on the AE security of Transform-then-Permute when the linear functions $L_{d,i}$ and L_e are invertible for all $1 \leq i \leq r$. Let q_p, q_e and q_d define the number of primitive, encryption and decryption queries respectively by an adversary and let σ_e and σ_d define all the data blocks processed, including nonce, associated data and message, in those encryption and decryption queries, respectively.

Algorithm 2 Description of Encryption/Decryption algorithms of the Transform-then-Permute mode with Associated data. $X = (x =_? y : p, q)$ means $X = p$ if $x = y$ and $X = q$ otherwise.

1: **function** $\mathsf{Enc}(K, N, A, M)$
2: $a \leftarrow a(|A|)$, $m \leftarrow \lceil |M|/r \rceil$
3: $(D_0, D_1, \ldots, D_{a+m}) \leftarrow \mathsf{Fmt}(N, A, M)$
4: $(M_1, \ldots, M_m) \xleftarrow{r} M$
5: $X_0 \leftarrow K \| 0^{b-\kappa} \oplus D_0$
6: $Y_0 \leftarrow \Pi(X_0)$
7: **for** $i = 1$ to a **do**
8: $X_i \leftarrow L_{ad}(Y_{i-1}) \oplus D_i$
9: $Y_i \leftarrow \Pi(X_i)$
10: **for** $j = 1$ to m **do**
11: $i = a + j$
12: $X_i \leftarrow L_e(Y_{i-1}) \oplus D_i$
13: $C_j \leftarrow M_j \oplus \lceil Y_{i-1} \rceil_{|M_j|}$
14: $Y_i \leftarrow \Pi(X_i)$
15: $T \leftarrow \lceil Y_{a+m} \rceil_\tau$
16: **return** $(C_1 \| \ldots \| C_m, T)$

1: **function** $\mathsf{Dec}(K, N, A, C, T)$
2: $a \leftarrow a(|A|)$, $m \leftarrow \lceil |C|/r \rceil$
3: $(D_0, D_1, \ldots, D_{a+m}) \leftarrow \mathsf{Fmt}(N, A, C)$
4: $(C_1, \ldots, C_m) \xleftarrow{r} C$
5: $X_0 \leftarrow K \| 0^{b-\kappa} \oplus D_0$
6: $Y_0 \leftarrow \Pi(X_0)$
7: **for** $i = 1$ to a **do**
8: $X_i \leftarrow L_{ad}(Y_{i-1}) \oplus D_i$
9: $Y_i \leftarrow \Pi(X_i)$
10: **for** $j = 1$ to m **do**
11: $i = a + j$
12: $X_i \leftarrow L_{d,|C_i|}(Y_{i-1}) \oplus D_i$
13: $M_j \leftarrow C_j \oplus \lceil Y_{i-1} \rceil_{|C_j|}$
14: $Y_i \leftarrow \Pi(X_i)$
15: $T \leftarrow \lceil Y_{a+m} \rceil_\tau$
16: **return** $T' =_? T : M_1 \| \ldots \| M_m, \perp$

Theorem 5. *Let TtP be a construction where $L_{d,i}$ for all $i \in [r]$ and L_e are invertible. For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\text{inv-TtP}}(\mathscr{A}) \leq\ & \frac{\sigma_d \mathsf{mcoll}(q_p, 2^\tau)}{2^c} + \frac{\sigma_d \mathsf{mcoll}(q_p, 2^r)}{2^c} + \frac{\sigma_d \mathsf{mcoll}'(q_p^2, 2^b)}{2^c} \\ & + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{2q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} \\ & + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.\end{aligned}$$

4.3 Proof of Theorem 5

The proof employs the coefficient H-technique of Theorem 2. To apply this method we need to first describe the ideal world which basically tries to simulate the construction. The real world behaves the same as the construction and would be described later. For the sake of notational simplicity, we assume the size of the nonce is at most $b-\kappa$. Later we mention how one can extend the proof when the nonce size is more than $b-\kappa$. We also assume that the adversary makes exactly q_p, q_e, and q_d many primitive, encryption, and decryption queries respectively.

4.3.1 Ideal World and Real World

Online Phase of Ideal World. The ideal world responds to three oracles, namely encryption queries, decryption queries, and primitive queries in the online phase.

(1) On Primitive Query $(\mathsf{W}_i, \mathsf{dir}_i)$:

The ideal world simulates $\Pi^{\pm}$ query honestly.[1] In particular, if $\mathsf{dir}_i = 1$, it sets $\mathsf{U}_i \leftarrow \mathsf{W}_i$ and returns $\mathsf{V}_i = \Pi(\mathsf{U}_i)$. Similarly, when $\mathsf{dir}_i = -1$, it sets $\mathsf{V}_i \leftarrow \mathsf{W}_i$ and returns $\mathsf{U}_i = \Pi^{-1}(\mathsf{V}_i)$.

(2) On Encryption Query $Q_i := (\mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i)$:

It samples $\mathsf{Y}_{i,0}, \ldots, \mathsf{Y}_{i,t_i} \xleftarrow{\$} \{0,1\}^b$ where $t_i = a_i + m_i$, $a_i = a(|\mathsf{A}_i|)$ and $m_i = \lceil \frac{|\mathsf{M}_i|}{r} \rceil$. Then, it returns $(\mathsf{C}_{i,1} \| \cdots \| \mathsf{C}_{i,m_i}, \mathsf{T}_i)$ where $(\mathsf{M}_{i,1}, \ldots, \mathsf{M}_{i,m_i}) \xleftarrow{r} \mathsf{M}_i$, $\mathsf{C}_{i,j} = \lceil \mathsf{Y}_{i,a_i+j-1} \rceil_{|\mathsf{M}_{i,j}|} \oplus \mathsf{M}_{i,j}$ for all $j \in [m_i]$ and $\mathsf{T}_i \leftarrow \lceil \mathsf{Y}_{i,t_i} \rceil_\tau$.

[1] For example, one can use lazy sampling to simulate random permutation.

(3) On Decryption Query $Q_i := (\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*, \mathsf{T}_i^*)$:

According to our convention we assume that the decryption query is always non-trivial. So the ideal world returns abort symbol $\mathsf{M}_i^* := \perp$.

Offline Phase of Ideal World. After completion of oracle interaction (the above three types of queries possibly in an interleaved manner), the ideal oracle sets $\mathcal{E}, , \mathcal{D}, \mathcal{P}$ to denote the set of all query indices corresponding to encryption, decryption and primitive queries respectively. So $\mathcal{E} \sqcup \mathcal{D} \sqcup \mathcal{P} = [q_e + q_d + q_p]$ and $|\mathcal{E}| = q_e$, $|\mathcal{D}| = q_d$, $|\mathcal{P}| = q_p$. Let the primitive transcript $\omega_p = (\mathsf{U}_i, \mathsf{V}_i, \mathsf{dir}_i)_{i \in \mathcal{P}}$ and let $\omega_p' := (\mathsf{U}_i, \mathsf{V}_i)_{i \in \mathcal{P}}$. The decryption transcript $\omega_d := (\mathsf{M}_i^*)_{i \in \mathcal{D}}$ where M_i^* is always $\perp$.

Now we describe some extended transcript (releasing additional information) for encryption queries. It samples $K \leftarrow\!\!\$\, \{0,1\}^\kappa$. For all i, let $\mathsf{Fmt}(\mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i) = (D_{i,0}, \ldots, D_{i,t_i})$ and for every $0 \le j \le t_i$, the intermediate input (X-value) is defined as

$$\mathsf{X}_{i,j} = \begin{cases} D_{i,0} \oplus K \| 0^{b-\kappa} & \text{if } j = 0 \\ L_e(\mathsf{Y}_{i,j-1}) \oplus D_{i,j} & \text{if } 1 \le j \le t_i \end{cases}$$

The encryption transcript $\omega_e = (\mathsf{X}_{i,j}\mathsf{Y}_{i,j})_{i \in \mathcal{E}, j \in [0..t_i]}$. So, the transcript of the adversary consists of $\omega := (Q, \omega_p, \omega_e, \omega_d)$ where $Q := (Q_i)_{i \in \mathcal{E} \cup \mathcal{D}}$.

Real World. In the online phase, the AE encryption and decryption queries and direct primitive queries are faithfully responded to based on $\Pi^{\pm}$. Like the ideal world, after the completion of interaction, the real world returns all X-values and Y-values corresponding to the encryption queries only. Note that a decryption query may return M_i which is not $\perp$.

4.3.2 Bad Transcripts

We define the bad transcripts into two main parts. We first define bad events due to encryption and primitive transcript. The following bad events say that (i) there is a collision among inputs/outputs of ω_p and ω_e (ii) there is a collision among input/outputs of ω_e. So, given that there is no such collision, all inputs and outputs are distinct and hence $\omega_e \cup \omega_p$ is permutation compatible (can be realized by random permutation). More formally, we define the following bad events:

- B1: For some $(\mathsf{U}, \mathsf{V}) \in \omega_p$, $\mathsf{K} = \lfloor \mathsf{U} \rfloor_\kappa$.
- B2: For some $i \in \mathcal{E}$, $j \in [t_i]$, $\mathsf{Y}_{i,j} \in \mathsf{range}(\omega_p)$, (in other words, $\mathsf{range}(\omega_e) \cap \mathsf{range}(\omega_p) \neq \emptyset$)
- B3: For some $i \in \mathcal{E}$, $j \in [t_i]$, $\mathsf{X}_{i,j} \in \mathsf{domain}(\omega_p)$, (in other words, $\mathsf{domain}(\omega_e) \cap \mathsf{domain}(\omega_p) \neq \emptyset$)
- B4: For some $(i \in \mathcal{E}, j \in [t_i]) \neq (i' \in \mathcal{E}, j' \in [t_{i'}])$, $\mathsf{Y}_{i,j} = \mathsf{Y}_{i',j'}$,
- B5: For some $(i \in \mathcal{E}, j \in [t_i]) \neq (i' \in \mathcal{E}, j' \in [t_{i'}])$, $\mathsf{X}_{i,j} = \mathsf{X}_{i',j'}$,

Now we describe the bad event due to decryption queries. Suppose the bad events $(\mathtt{B1} \vee \cdots \vee \mathtt{B5})$ as defined above due to encryption queries and primitives don't occur i.e. we have $\omega_p \cup \omega_e$ is permutation compatible. Suppose Π' is the partially defined permutation defined over the domain of $\omega_p \cup \omega_e$ and mapping the corresponding range elements. For each decryption query $Q_i = (\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*, \mathsf{T}_i^*)$, we compute $a_i = a(|\mathsf{A}_i^*|)$, $m_i = \lceil |\mathsf{C}_i^*|/r \rceil$ and $\mathsf{Fmt}(\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*) = (D_{i,0}^*, \ldots, D_{i,t_i}^*)$. We define p_i' is the largest index j for which the input X_j is in the domain of $\omega_e \cup \omega_p$ while we run the decryption

algorithm using Π' for Q_i. Consider the case, $p'_i = t_i$ i.e. the complete decryption algorithm computation for the query is determined by the $\omega_e \cup \omega_p$ transcript. In such a case we define bad (called $\texttt{mBAD}$) if the corresponding tag also matches. Note that for this bad transcript the real world should not abort the decryption query. Now we define all bad events in a more formal way.

DEFINITION OF p_i. Before we define p'_i, we first define p_i which is the input index we can compute for the decryption query only using encryption queries transcript. Formally, p_i is defined as -1 if for all $i' \in \mathcal{E}$, $\mathsf{N}_{i'} \neq \mathsf{N}^*_i$. Otherwise, there exists a unique $i' \in \mathcal{E}$ such that $\mathsf{N}_{i'} = \mathsf{N}^*_i$ (as we consider nonce-respecting adversary only). Let $p_i + 1$ denote the length of the longest common prefix of $(D_{i',0}, \cdots, D_{i',t_{i'}})$ and $(D^*_{i,0}, \cdots, D^*_{i,t_i})$. Note that $p_i = -1$ in case there is no common prefix.

We now define $\mathsf{Y}^*_{i,0..p_i} = \mathsf{Y}_{i',0..p_i}$, $\mathsf{X}^*_{i,0..p_i} = \mathsf{X}_{i',0..p_i}$ when $p_i \geq 0$ and

$$\mathsf{X}^*_{i,p_i+1} = \begin{cases} L_e(\mathsf{Y}_{i',p_i}) \ \oplus \ D^*_{i,p_i+1} & \text{if } p_i \geq 0. \\ \mathsf{K} \| \mathsf{N}^*_i & \text{if } p_i = -1. \end{cases}$$

By Lemma 3, $p_i < t_i$, $p_i < t_{i'}$. By definition of longest common-prefix, we have $\mathsf{X}^*_{i,p_i+1} \neq \mathsf{X}_{i',p_i+1}$.

DEFINITION OF p'_i. If $p_i < a_i$ or if $\mathsf{X}^*_{i,p_i+1} \notin \mathsf{domain}(\omega_p)$ define $p'_i = p_i$. Else, we further extend X^*-values and Y^*-values based on the primitive transcript ω_p. Let $x_{i,j} := D^*_{i,j}$ for all $i \in \mathcal{D}, 1 \leq j \leq t_i$. If there is a labeled walk (in the labeled directed graph induced by ω_p as described in Section 3.1 from Y^*_{i,p_i+1} with label $(x_{i,p_i+2}, \ldots, x_{i,j})$ then

we denote the end node as $\mathsf{Y}^*_{i,j}$. In notation we have

$$\mathsf{Y}^*_{i,p_i+1} \xrightarrow{(x_{i,p_i+2},\ldots,x_{i,j})} \mathsf{Y}^*_{i,j}.$$

Let p'_i denote the maximum of all such possible j's. For all those i and j in which $\mathsf{Y}^*_{i,j}$ has been defined as described above, we define $\mathsf{X}^*_{i,j+1} := L_d(\mathsf{Y}^*_{i,j}) \oplus x_{i,j+1}$.

Bad events due to the decryption queries in the transcript:

$\mathtt{mBAD}$: For some $i \in \mathcal{D}$ with $p'_i = t_i$ and $\lceil \mathsf{Y}^*_{i,t_i} \rceil_\tau = \mathsf{T}^*_i$.

$\mathtt{B6}$: For some $i \in \mathcal{D}$, $p'_i < t_i$ and, $\mathsf{X}^*_{i,p'_i+1} \in \mathsf{domain}(\omega_e) \cup \mathsf{domain}(\omega_p)$.

We write $\mathtt{BAD}$ to denote the event that the ideal world transcript Θ_0 is bad. Then, with a slight abuse of notations and union bound, we have

$$\mathtt{BAD} = \mathtt{mBAD} \cup \left(\bigcup_{i=1}^{6} \mathtt{Bi} \right). \tag{4.1}$$

Lemma 4 upper bounds the probability of $\mathtt{mBAD}$ and Lemma 5 upper bounds the probability of $\bigcup_{i=1}^{6} \mathtt{Bi}$. The proofs of Lemma 4 and 5 are postponed to Subsections 4.3.4 and 4.3.5, respectively.

Lemma 4. *Let μ_{q_p} be the maximum multi-chain advantage (see Subsection 3.2.1) over q_p primitive queries. Then, we have*

$$\Pr\left[\mathtt{mBAD}\right] \leq \frac{\sigma_d \cdot \mu_{q_p}}{2^c}.$$

Lemma 5. *For $q_p < 2^{b-1}$, we have*

$$\Pr\left[\bigcup_{i=1}^{6} \mathtt{Bi}\right] \leq \frac{q_p}{2^\kappa} + \frac{6\sigma_e q_p}{2^b} + \frac{2\sigma_e^2}{2^b} + \frac{\sigma_e + q_p}{2^b} + \frac{2q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^{b-\tau}} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

4.3.3 Good Transcript Analysis

The motivation for all the bad events would be clear from the understanding of a good transcript (i.e., not a bad transcript). Let $\omega = (Q, \omega_p, \omega_e, \omega_d)$ be a good transcript. For the sake of notation simply we ignore the query transcript Q as it is not required to compute the probability of a transcript.

1. The tuples ω_e is permutation compatible and disjoint from ω_p. So union of tuples $\omega_e \cup \omega_p$ is also permutation compatible.

2. Let $\mathcal{D}_1$ (type-1 decryption query) be the set of all $i \in \mathcal{D}$, if $p'_i = t_i$ with $\lceil \mathsf{Y}^*_{i,t_i} \rceil_\tau \neq \mathsf{T}^*_i$. In this case, decryption algorithm should abort with probability one. Set of all other indices is denoted as $\mathcal{D}_2$ (type-2 decryption query). In this case, $p'_i < t_i$ but $\mathsf{X}^*_{i,p'_i+1} \notin \mathsf{domain}(\omega_e \cup \omega_p)$. So, Y^*_{i,p'_i+1} value and subsequent Y-values will have almost b-bit entropy. Thus, with a negligible probability we may not abort the query.

Ideal World Interpolation Probability. Let Θ_0 and Θ_1 denote the transcript random variable obtained in the ideal world and real world respectively. As noted before, all the input-output pairs for the underlying permutation are compatible. In

the ideal world, all the Y values are sampled uniform at random; the list ω_p is just the partial representation of Π; and all the decryption queries are degenerately aborted; whence we get

$$\Pr[\Theta_0 = \omega] = \frac{1}{2^{b\sigma_e}(2^b)_{q_p}}.$$

Here σ_e denotes the total number of blocks present in all encryption queries including nonce. In notation $\sigma_e = q_e + \sum_i m_i$.

Real World Interpolation Probability. In the real world, for ω we denote the encryption query, decryption query, and primitive query tuples by ω_e, ω_d and ω_p, respectively. Then, we have

$$\begin{aligned}
\Pr[\Theta_1 = \omega] &= \Pr[\Theta_1 = (\omega_e, \omega_p, \omega_d)] \\
&= \Pr[\omega_e, \omega_p] \cdot \Pr[\omega_d \mid \omega_e, \omega_p] \\
&= \Pr[\omega_e, \omega_p] \cdot (1 - \Pr[\neg\omega_d \mid \omega_e, \omega_p]) \\
&\leq \Pr[\omega_e, \omega_p] \cdot \left(1 - \sum_{i \in \mathcal{D}_2} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right) \qquad (4.2)
\end{aligned}$$

Here we have slightly abused the notation to use $\neg\omega_{d,i}$ to denote the event that the i-th decryption query successfully decrypts and $\neg\omega_d$ is the union $\cup_{i \in \mathcal{D}_2} \neg\omega_{d,i}$ (i.e. at least one decryption query successfully decrypts). The encryption and primitive queries are mutually permutation compatible, so we have

$$\Pr_{\Theta_1}[\omega_e, \omega_p] = 1/(2^b)_{\sigma_e + q_p} \geq \Pr_{\Theta_0}[\omega_e, \omega_p].$$

Now we show an upper bound $\Pr_{\Theta_1}[\neg\omega_{d,i} \mid \omega_e, \omega_p] \leq \frac{2(\sigma+q_p)}{2^b} + \frac{2}{2^\tau}$ for every type-2 decryption query. We quickly recall that $\mathsf{Fmt}(\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*) = (D_{i,0}^*, \ldots, D_{i,t_i}^*)$. So, $\neg\omega_{d,i}$

is same as $\lceil \Pi(\mathsf{X}^*_{i,t_i}) \rceil_\tau = \mathsf{T}^*_i$ where $\mathsf{X}^*_{i,j}$ values have been defined recursively as follows

$$\mathsf{X}^*_{i,j} = L_d\big(\Pi(\mathsf{X}^*_{i,j-1})\big) \oplus D^*_{i,j}, \quad p'_i + 1 < j \leq t_i.$$

Let $\mathcal{I}$ and $\mathcal{O}$ denote the set of inputs and outputs for Π which are present in the transcript (ω_e, ω_p). Recall that X^*_{i,p'_i+1} is fresh, i.e., $\mathsf{X}^*_{i,p'_i+1} \notin \mathcal{I}$.

Claim 1. $\Pr\left[\mathsf{X}^*_{i,j} \text{ is fresh }\right] \geq (1 - \frac{2(\sigma_e+q_p+t_i)}{2^b}) \;\; \forall\; p'_i + 1 < j \leq t_i$.

Proof. Since X^*_{i,p'_i+1} is not the last block, then the next input block may collide with some encryption or primitive input block with probability at most $\frac{\sigma_e+q_p}{2^b-\sigma_e-q_p}$. Applying this same argument for all the successive blocks till the last one, we get that if none of the previous block input collides then the probability that the last block input collides is at most $\frac{(\sigma_e+q_p+t_i-p'_i+2)}{2^b-\sigma_e-q_p-t_i+p'_i+2} \leq \frac{2(\sigma_e+q_p+t_i)}{2^b}$. □

Claim 2. $\Pr[\neg\omega_{d,i} \mid \mathsf{X}^*_{i,j} \text{ are fresh }] \leq \frac{2}{2^\tau}$.

Proof. Since the last input block X^*_{i,t_i} is fresh, hence $\Pi(\mathsf{X}^*_{i,t_i}) = \mathsf{T}^*_i$ with probability at most $2/2^\tau$ (provided $\sigma_e + q_p \leq 2^{b-1}$ which can be assumed, since otherwise our bound is trivially true). □

Let $\mathtt{E}_j$ denote the event that $\mathsf{X}^*_{i,j}$ is fresh and $\mathtt{E} := \wedge_{j=p'_i+1}^{t_i} \mathtt{E}_j$

Using the claims, we have

$$\begin{aligned}\Pr_{\Theta_1}[\neg\omega_{d,i} \mid \omega_e, \omega_p] &\leq \Pr_{\Theta_1}[\neg\omega_{d,i} \wedge E \mid \omega_e, \omega_p] + \Pr[E^c]. \\ &\leq \frac{2}{2^\tau} + \sum_{j=p'_i+1}^{t_i} \frac{\sigma_d + \sigma_e + q_p}{2^{b-1}}.\end{aligned}$$

The last inequality follows from the above claims. Now, we can proceed by using the union bound as follows.

$$\begin{aligned}\Pr[\neg\omega_d \mid \omega_e, \omega_p] &\leq \sum_{i\in\mathcal{D}} \frac{2t_i(\sigma_e + q_e + \sigma_d)}{2^b} + \frac{2}{2^\tau} \\ &\leq \frac{2\sigma_d(\sigma_e + \sigma_d + q_p)}{2^b} + \frac{2q_d}{2^\tau} \\ &= \frac{2\sigma_d(\sigma + q_p)}{2^b} + \frac{2q_d}{2^\tau}\end{aligned}$$

Finally, Theorem 5 follows from the H-technique (Theorem 2) combined with Theorem 3, Lemma 4, Lemma 5 and Eq. (4.2).

Remark 4. *As described in the algorithm, in the case where nonce size is greater than $b-\kappa$, we treat the excess length of the nonce as part of the associated data. For such a* TtP *construction the internal values of the encryption transcripts are chosen in a prefix respecting manner. Suppose the i, i'-th queries $(D_{i,0}, \ldots, D_{i,t_i})$ and $(D_{i',0}, \ldots, D_{i',t_j})$ have a maximum common prefix of length p_i and let without loss of generality $i < i'$. Then we set $Y_{i,j} = Y_{i',j}$ and $X_{i,j} = X_{i',j} \forall 0 \leq j \leq p_i$. The rest of the proof remains the same.*

4.3.4 Proof of Lemma 4 (Multi-chain Bad Transcript Analysis)

Suppose the event holds for the i-th decryption query and $\mathsf{N}_i^* = \mathsf{N}_{i'}$. So, $\left(\mathsf{X}_{i,p_i+1}^*, \mathsf{Y}_{i,p_i+1}^*\right)$ must be the one of the starting node of the multi-chain. Hence as in Definition 2, if (U, V) be any other starting node of the multi-chain, then we must have $\lceil U \rceil_r = \lceil \mathsf{X}_{i,p_i+1}^* \rceil_r$. Now as before , let $\mathsf{W}_{t_i - p_i}$ denote the maximum size of the set of multi-chain

of length $t_i - p_i$, induced by L_d and ω_p. As $\lfloor \mathsf{Y}_{i',p_i} \rfloor_c$ is chosen at random (and independent of ω_p), and C^*_{i,p_i+1} is fixed, the probability to hold $\mathtt{mBAD}$ for i-th decryption query is at most $\mathsf{W}_{m_i}/2^c$ given the transcript ω_p. So by union bound, the conditional probability $\Pr[\mathtt{mBAD} \mid \omega_p] \leq \sum_{i \in \mathcal{D}} \frac{\mathsf{W}_{m_i}}{2^c}$.

Since the decryption query data complexity of the adversary is bounded by σ_d blocks we have $\sum_{i \in \mathcal{D}} m_i \leq \sigma_d$. Now,

$$\sum_{i \in \mathcal{D}} \mathsf{W}_{m_i} \leq \sum_{i \in \mathcal{D}} \left(\max_{k \leq m_i} \frac{\mathsf{W}_k}{k} \times m_i \right) \leq \max_k \frac{\mathsf{W}_k}{k} \times \sigma_d.$$

Hence,

$$\Pr\left[\mathtt{mBAD}\right] \leq \sum_{i \in \mathcal{D}} \frac{\mathsf{Ex}\left[\mathsf{W}_{m_i}\right]}{2^c} \leq \max_k \mathsf{Ex}\left[\frac{\mathsf{W}_k}{k}\right] \times \frac{\sigma_d}{2^c} \leq \frac{\sigma_d \cdot \mu_{q_p}}{2^c}.$$

4.3.5 Proof of Lemma 5 (Bad Transcript Analysis)

From the union bound we have

$$\Pr\left[\bigcup_{i=1}^{6} \mathtt{Bi}\right] \leq \Pr\left[\mathtt{B1}\right] + \Pr\left[\mathtt{B2}\right] + \Pr\left[\mathtt{B3}|\neg\mathtt{B1}\right] + \Pr\left[\mathtt{B4}\right] + \Pr\left[\mathtt{B5}\right] + \Pr\left[\mathtt{B6}|\neg\mathtt{B1}\right].$$

It is sufficient to upper bound each of these individual probabilities. We bound the probabilities of these events in the following:

Bounding $\Pr[\mathtt{B1}]$: This is basically the key recovery event, i.e., the event that the ad-

versary recovers the master key K by direct queries to the internal random permutation (can be both forward or backward). For a fixed entry $(\mathsf{U}, \mathsf{V}) \in \omega_p$, the probability that $\mathsf{K} = \lfloor \mathsf{U} \rfloor_\kappa$ is bounded by at most $2^{-\kappa}$, as K is chosen uniform at random from $\{0,1\}^\kappa$. Thus, we have

$$\Pr[\mathtt{B1}] \leq \frac{q_p}{2^\kappa}.$$

Bounding $\Pr[\mathtt{B2}]$: This event can be analyzed in several cases as below:

Case 1: $\exists i, j, a$, $\mathsf{Y}_{i,j} = \mathsf{V}_a$, encryption after primitive: Since $Y_{i,j}$ are chosen uniformly at random, this case can be bounded for fixed i, j, a with probability at most $1/2^b$. We have at most σ_e many (i, j) pairs and q_p many a indices. Hence this case can be bounded by at most $\sigma_e q_p / 2^b$.

Case 2: $\exists i, j, a$, $\mathsf{Y}_{i,j} = \mathsf{V}_a$, $\mathsf{dir}_a = +$, encryption before primitive: This case can be bounded by probability at most $1/(2^b - q_p + 1)$. We have at most σ_e many (i, j) pairs and q_p many a indices. Thus this can be bounded by at most $\sigma_e q_p/(2^b - q_p + 1) \leq 2\sigma_e q_p / 2^b$ (as $q_p \leq 2^{b-1}$).

Case 3: $\exists i, j \neq t_i, a$, $\mathsf{Y}_{i,j} = \mathsf{V}_a$, $\mathsf{dir}_a = -$, encryption before primitive: Here the adversary has access to $\lceil \mathsf{Y}_{i,j} \rceil_r$, as this value has already been released. Let Φ_{out} denote the

number of multicollisions among all $\lceil \mathsf{Y}_{i',j'} \rceil_r$ values. Now, we have

$$\begin{aligned}
\Pr[\text{Case 3}] &= \sum_{\Phi_{out}} \Pr[\text{Case 3} \mid \Phi_{out}] \cdot \Pr[\Phi_{out}] \\
&\leq \sum_{\Phi_{out}} \frac{\Phi_{out} \times q_p}{2^c} \cdot \Pr[\Phi_{out}] \\
&\leq \frac{q_p}{2^c} \times \mathsf{Ex}\left[\Phi_{out}\right] \\
&\leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}.
\end{aligned}$$

Case 4: $\exists i, a, \mathsf{Y}_{i,t_i} = \mathsf{V}_a, \mathsf{dir}_a = -$, encryption before primitive: This case is same as case-3 plugging in r as τ and c as $b - \tau$. So, $\Pr[\text{Case 4}] \leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}}$.

By using the union bound, we have

$$\Pr[\mathtt{B2}] \leq \frac{3\sigma_e q_p}{2^b} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}}.$$

Bounding $\Pr[\mathtt{B3} | \neg \mathtt{B1}]$: This means $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$ where $j > 0$ (as $\mathtt{B1}$ does not hold). So, we can have the following cases with $j > 0$:

Case 1: $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$, encryption after primitive: This case can be bounded by probability at most $1/2^b$, as $\mathsf{Y}_{i,j-1}$ is chosen uniform at random and L_e in invertible. We have at most σ_e many (i, j) pairs and q_p many a indices. Thus this can be bounded by at most $\sigma_e q_p / 2^b$.

Case 2: $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$, $\mathsf{dir}_a = -$, encryption before primitive: This case can be bounded by probability at most $1/(2^b - q_p + 1)$. We have at most σ_e many (i, j) pairs and q_p many a indices. Thus this can be bounded by at most $2\sigma_e q_p / 2^b$.

Case 3: $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$, $\mathsf{dir}_a = +$, encryption before primitive: Since L_e is invertible, we can define $\mathsf{V}' = L_e^{-1}(U_a \oplus D_j)$. Then using the invertibility of L_e we have this event is same as the event $\exists i, 0 < j$, $\mathsf{Y}_{i,j-1} = \mathsf{V}'$ for some $\mathsf{V}' \in \omega_p$. Since $j \leq t_i$ we have this event is the same as Case 3 of B2. Hence,

$$\Pr[\text{Case 3}] \leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}.$$

$$\Pr[\mathtt{B3}|\neg\mathtt{B1}] \leq \frac{3\sigma_e q_p}{2^b} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}.$$

Bounding Pr[B4] and Pr[B5]: The probability of this event can be simply bounded by birthday paradox and so it is at most $\sigma_e(\sigma_e - 1)/2^b$.

Bounding Pr[B6|¬B1]: This event can be analyzed in several cases.

Case 1 $p'_i < a_i$: Since during associated data processing no information is leaked to the adversary and $Y^*_{i,j}$-s are sampled uniformly at random hence for $p'_i < a_i$, the distribution function of $X^*_{i,p'_i+1} = Y^*_{i,p'_i} \oplus D^*_{i,p'_i+1}$ is uniform. Hence

$$\Pr\left[\text{Case 1}\right] \leq \frac{\sigma_e + q_p}{2^b}.$$

Case 2 $a_i \leq p_i \leq p'_i$: This corresponds to the case when either the first non-trivial decryption query block doesn't match any primitive query or it matches a primitive query and follows a partial chain and then matches with some encryption query block. Doing similar analysis as in Case 3 of B3|¬B1, The probability that this happens for i-th decryption is at most $q_p/2^c \times m_i \Phi_{out}/2^c$. Summing over all $i \in \mathcal{D}$, the conditional

probability is at most $\frac{q_p \sigma_d \Phi_{out}}{2^{2c}}$. By taking expectation we obtain the following:

$$\Pr[\text{Case 3}] \leq \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

$$\Pr\left[\mathsf{B6} | \neg \mathsf{B1}\right] \leq \frac{\sigma_e + q_p}{2^b} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

By adding all these probabilities we prove our result.

4.4 Instantiating TtP and Application of Theorem 5

Now, we describe how Transform-then-Permute can capture a wide class of permutation-based sequential constructions such as duplex (or Sponge AE), Beetle, and SpoC, in which the only non-linear operation is the underlying permutation. We further show that Beetle and SpoC fall under a special class of TtP constructions where the feedback functions are invertible and hence we can apply Theorem 5 in those cases. Finally, we discuss the case of Sponge AE which doesn't belong to this special class.

4.4.1 How to Convert a Generalized Sponge-type Construction to TtP

Let $L : \{0,1\}^b \times \{0,1\}^r \rightarrow \{0,1\}^b \times \{0,1\}^r$ be any linear function defined by the transformation matrix $L = \begin{bmatrix} L_{1,1} & L_{1,2} \\ L_{2,1} & L_{2,2} \end{bmatrix}$ consisting of $b \times b$ matrix $L_{1,1}$, $b \times r$ matrix $L_{1,2}$, $r \times b$ matrix $L_{2,1}$, $r \times r$ matrix $L_{2,2}$. Consider the Sponge-type construction which takes state input X_i and data input M_i and generate the data output C_i and next state input X_{i+1} as follows:

$$Y_i = \Pi(X_i); \quad \begin{bmatrix} X_{i+1} \\ C_i \end{bmatrix} = L \cdot \begin{bmatrix} Y_i \\ M_i \end{bmatrix}$$

As $L_{2,1} \cdot Y + L_{2,2} \cdot M = C$, the rank of $L_{2,2}$ must be r, otherwise encryption is not a bijective function from message space to ciphertext space. For the sake of simplicity we can assume that $L_{2,2} = I_r$ (the identity matrix of size r). Otherwise, we can redefine message block as $M' = L_{2,2} \cdot M$.

Now, we observe that rank of $L_{2,1}$ is r. If not, then there exists a non-zero vector γ such that $\gamma \cdot L_{2,1} = 0$. Hence, $\gamma \cdot M = \gamma \cdot C$ holds with probability 1. In case of ideal permutation as γ is non-zero and C is chosen uniformly independent of M, this event occurs with probability $\frac{1}{2}$. Hence the privacy advantage of any adversary for such a construction will be $\geq \frac{1}{2}$. As rank of $L_{2,1}$ is r, there exists an invertible matrix $Z_{b \times b}$ such that $L_{2,1} \cdot Z = I_r \| 0_{r \times (b-r)}$. Let $L_e = L_{1,1} \cdot Z$. Then by simple matrix algebra we

have

$$\begin{bmatrix} X_{i+1} \\ C_i \end{bmatrix} = \begin{bmatrix} L_e & L_{1,2} \\ I_r \| 0_{r\times(b-r)} & I_r \end{bmatrix} \cdot \begin{bmatrix} Y_i' \\ M_i \end{bmatrix}$$

where $Y_i' = Z^{-1} \cdot Y_i$. Note that, multiplication by an invertible matrix is a permutation and composition of a random permutation with a public permutation is again a random permutation. Hence, we can redefine the random permutation output as $Z^{-1} \cdot \Pi(X_i)$. Let us denote $\mathsf{encode}(M) = L_{1,2} \cdot M$ and hence the general linear function based Sponge-type construction boils down to the construction TtP.

4.4.2 New Improved Security of Beetle

In Beetle [35], the linear function L_e is defined as $L_e(y\|x_1\|x_2) \mapsto (y\|x_2\|x_2 \oplus x_1)$, where $(y, x_1, x_2) \in \{0,1\}^c \times \{0,1\}^{r/2} \times \{0,1\}^{r/2}$. The linear function $L_{d,i}$ is defined by

$$L_{d,i}(y\|x_1\|x_2) = \begin{cases} (y\|x_2\|\lfloor x_2 \oplus x_1 \rfloor_{r/2-i}\|\lceil x_1 \rceil_i) \text{ for } 0 \le i \le r/2 \\ (y\|\lfloor x_2 \rfloor_{r-i}\|\lceil x_2 \oplus x_1 \rceil_{i-r/2}\|x_1) \text{ for } r/2 \le i \le r \end{cases},$$

where $(y, x_1, x_2) \in \{0,1\}^c \times \{0,1\}^{r/2} \times \{0,1\}^{r/2}$. Clearly the L_e and $L_{d,i}$ functions are invertible for all $0 \le i \le r$. Further, they have full rank.

Remark 5. *The PHOTON-Beetle [9] design which is a finalist in the NIST LwC standardization process uses a feedback function that is a linear transformation of the feedback function of Beetle [35]. By applying the conversion method as described in Subsection 4.4.1 the PHOTON-Beetle design can be viewed as a TtP design with the same linear function L_e as described above.*

PREVIOUS BOUND: In [35], the authors proved that for any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Beetle}}(\mathscr{A}) \leq \frac{2(\sigma_e + q_p)\sigma_d}{2^b} + \left(\frac{\sigma_e + q_p}{2^{r-1}} + \frac{q_p}{2^c}\right)^r + \frac{r\sigma_d}{2^c} + \frac{q_v}{2^r}. \tag{4.3}$$

The primary version of PHOTON-Beetle [9] has $r = \tau = c = 128$ and $b = 256$. Comparing with the σ and q_p values prescribed by NIST we have $2^r = 2^\tau \geq q_p \geq \sigma$ and $2^b \geq b^2 q_p^2$. The secondary version of PHOTON-Beetle [9] has $r = 32, c = 224, \tau = 128$ and $b = 256$. Comparing with the σ and q_p values prescribed by NIST we have $2^\tau \geq q_p \geq \sigma$, $\sigma \geq 2^r$ and $2^b \geq b^2 q_p^2$.

By Eq. 4.3 the advantage of Beetle is bounded by $\left(\frac{q_p}{2^{r-1}}\right)^r$. So, for Beetle to be secure, r has to be large. It can be noticed that the primary version of PHOTON-Beetle has $r = 128 > 112$. Hence by Eq. 4.3, it is secure within the NIST LwC requirements. For secondary version of PHOTON-Beetle, we have $r = 32 < 112$ and hence Eq. 4.3 does not guarantee the security for this version under NIST LwC requirements.

NEW IMPROVED BOUND: Since the feedback function of Beetle is invertible, we can apply Theorem 5. Specifically, we have

Corollary 4. *For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, its AEAD advantage against the primary version of PHOTON-Beetle is as follows*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \leq \frac{4\tau\sigma_d}{2^c} + \frac{4r\sigma_d}{2^c} + \frac{4b\sigma_d}{2^c} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} \\ + \frac{8rq_p}{2^c} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}.\end{aligned}$$

The AEAD advantage of $\mathscr{A}$ against the secondary version of PHOTON-Beetle is as

follows

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \leq \frac{4\tau\sigma_d}{2^c} + \frac{4\sigma_d \cdot q_p}{2^b} + \frac{4b\sigma_d}{2^c} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma + q_p)}{2^b} + \frac{16\sigma_e q_p}{2^b} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{5q_p\sigma_d\sigma_e}{2^{b+c}}.$$

Corollary 4 follows from Theorem 5, and Proposition 4 and 5. Further, using the relation that $\sigma \leq q_p$ (as per NIST LwC requirements) we can bound the advantage in case of primary version as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \frac{13rq_p}{2^c},$$

and the secondary version as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \frac{17q_p\sigma}{2^b}.$$

Clearly, by this new improved security bound, it is proved that both the primary and the secondary version of PHOTON-Beetle are secured under the NIST requirements.

The major difference between our analysis and the analysis of [35] is that we use the expected number of multi-chains to bound the security of Beetle, whereas in [36], it was only done using multicollision probability at the rate part. This is the reason why our new bound is much tighter than the existing one.

4.4.3 Security of SpoC

In SpoC [2], the linear function L_e is identity, and the linear function L_d is defined by the mapping $L(x, y) \mapsto (x, x\|0^{c-r} \oplus y)$, where $(x, y) \in \{0,1\}^r \times \{0,1\}^c$. Clearly, L_e and L_d functions are involutions, and hence invertible. Further, it is easy to check that they have full rank.

Corollary 5. *For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, the AEAD advantage of $\mathscr{A}$ against the primary version of SpoC is given by,*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC}}(\mathscr{A}) \leq \frac{5q_p\sigma_d}{2^{c+\tau}} + \frac{5q_p\sigma_d\cdot}{2^b} + \frac{4b^3q_p^2\sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma+q_p)}{2^b} \\ + \frac{6\sigma_e q_p}{2^b} + \frac{8rq_p}{2^c} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{\sigma_e+q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}\end{aligned}$$

Corollary 5 follows from Theorem 5, and Proposition 4 and 5. The primary version of SpoC mode of AEAD has $r = \tau = 64$, $b = 192$. Using the NIST prescribed values of σ and q_p we have $\sigma < 2^r$ but $2^r = 2^\tau \leq q_p$ and $2^b \leq b^2q_p^2$. Further, using the relation that $\sigma \leq q_p$ (as per NIST LwC requirements) we can bound the advantage as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \frac{13q_p\sigma}{2^b}.$$

4.4.4 Interpretation of Corollary 4 and 5 in Lieu of NIST LwC

Keeping in mind the NIST LwC requirement of time complexity $q_p = 2^{112}$ and data complexity $r\sigma = 2^{53}$ we try to find out the smallest possible permutation under which

the Beetle and SpoC modes can achieve security. We take $2^r \leq \sigma \leq q_p \leq 2^c$. We further assume that $\sigma \leq 2^\tau \leq q_p$ and $2^b \leq b^2 q_p^2$. Then, by applying Proposition 4 and 5 to simplify and improve the bounds in Corollary 4 or 5, we have

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC/Beetle}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \frac{17\sigma q_p}{2^b}.$$

It can be easily verified that Beetle and SpoC instantiated with a permutation of size at least 165-bit with a $r = 32$-bit rate can achieve security close to the NIST LwC requirements. For instance, Beetle and SpoC instantiated with the 176-bit permutation from the SPONGENT family [26] achieves NIST LwC requirements. Further, we note that there could be a possibility to further reduce the constants appearing in the above expression using a finer analysis. Specifically, if we ignore the constants, a 160-bit permutation with a $r = 32$-bit rate suffices for NIST LwC requirements.

4.5 Sponge as a Transform-then-Permute Mode

Note that the general Sponge construction can be viewed as an instantiation of TtP. In case of the original Sponge construction, the L_d function is defined by $L_d(x, y) \mapsto (0^r, y)$ where $(x, y) \in \{0,1\}^r \times \{0,1\}^c$. Note that the L_d function is not invertible. Hence the results of Theorem 3 can not be applied in case of original Sponge. However since L_e is invertible, with a similar analysis as in the case of TtP we get,

Corollary 6. *For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, its AEAD advantage against the general Sponge is given by*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Sponge}}(\mathscr{A}) \leq \frac{\sigma_d \cdot \mu_{q_p}}{2^c} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{2q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

Bounding μ_{q_p} in the case of Sponge is an interesting problem that we deal with next. We must mention that it seems very hard to have a tight estimate of μ_{q_p} for Sponge AEAD. A straightforward estimate of μ_{q_p} leads to the known security bound of $\sigma_d q_p / 2^c$. Hence to bound μ_{q_p} for the general Sponge AEAD we start by defining a new type of graph structure called "the query graph structure".

4.5.1 Query Graph Structure

Let $b > c$ be two integers and $\mathcal{L} = \{(U_1, V_1), (U_2, V_2), \ldots, (U_T, V_T)\} \subseteq \{0,1\}^b \times \{0,1\}^c$ be a set of tuples such that for all $i \neq j$, $U_i \neq U_j$. We consider a directed graph $G_{\mathcal{L}}$ (denoted simply as G whenever $\mathcal{L}$ is understood from the context) with set of vertices $V(G_{\mathcal{L}}) = \mathsf{range}(\mathcal{L}) \cup \{\lceil U \rceil_c \mid U \in \mathsf{domain}(\mathcal{L})\}$, label set $\{0,1\}^{b-c}$ and the set of all labeled edges of the form $u \xrightarrow{x} v$ for all $(u \| x, v) \in \mathcal{L}$. Given $u_0, u_l \in V(G)$ we define a directed walk of length l with label $(x_1, \ldots, x_l) \in \{0,1\}^{(b-c)l}$ from u_0 to u_l if and only if there exists vertices $u_1, \ldots, u_{l-1} \in V(G)$ such that for all $i \in \{1, \ldots, l\}$ we have $u_{i-1} \xrightarrow{x_i} u_i$. We denote this walk by

$$u_0 \xrightarrow{x_1} u_1 \xrightarrow{x_2} \cdots \xrightarrow{x_{l-1}} u_{l-1} \xrightarrow{x_l} u_l$$

or we simply write $u_0 \xrightarrow{x} u_l$ where $x := (x_1, \ldots, x_l)$. Given $u, v \in V(G)$ we define the *distance* between u and v as length of the shortest walk from u to v and denote it by $d(u, v)$. In notation,

$$d(u,v) := \min \left\{ l \in \mathbb{N} \mid \exists (x_1, \ldots, x_l);\ u \xrightarrow{x_1, \ldots, x_l} v \text{ in } G \right\}.$$

Definition 4. *Given a directed graph G, a vertex $v \in V(G)$ is called a collision point if there exists vertices $u, u' \in V(G)$ and labels $x, x' \in \{0,1\}^{b-c}$, $u\|x \neq u'\|x'$ such that $u \xrightarrow{x} v$ and $u' \xrightarrow{x'} v$ in G.*

Definition 5. *A vertex $u \in V(G)$ is called a collision vertex if for some $x_1, \ldots, x_l \in \{0,1\}^{b-c}$ and some collision point $v \in V(G)$ there exists a walk $u \xrightarrow{x_1, \ldots, x_l} v$. We say v is a collision point of u.*

Definition 6. *Let u be any collision vertex in G. Note that u may have multiple collision points. Define the degree of a collision vertex as its distance from its nearest collision point. In notation,*

$$degree(u) = \min\{d(u, v) \mid v \text{ is a collision point of } u\}.$$

Lemma 6. *Let λ_k denote the number of collision vertices of degree k in $V(G)$. Then,*

$$\lambda_{k+1} \leq \lambda_k \quad \forall k \in \mathbb{N}.$$

Proof. First, we show that for each collision vertex of degree $k+1$ there exists a collision vertex of degree k. Consider a collision vertex $u \in V(G)$ of degree $k+1 \in \mathbb{N}$ with nearest collision point $v \in V(G)$. Let $u \xrightarrow{x_1} u_1 \xrightarrow{x_2} \cdots \xrightarrow{x_{k+1}} v$ be the walk of length $k+1$ from u to v. Then by definition, none of $u_1, \ldots, u_k$ is a collision point. Hence u_1 is a collision vertex of degree $\leq k$. Now suppose u_1 is a collision vertex of degree $l < k$. Let v' be the nearest collision point of u_1. Then, there exists a walk $u_1 \xrightarrow{(y_1,\ldots,y_l)} v'$ in G for some labels $y_1, \ldots, y_l \in \{0,1\}^{b-c}$. But then $u \xrightarrow{(x_1,y_1,\ldots,y_l)} v'$ in G. Hence u has degree $l+1 < k+1$, a contradiction. Hence u_1 has degree k.

Further, if u' is another collision vertex of degree $k+1$ in G with nearest collision point v_1, then u_1 can't be a vertex in the u' to v_1 walk. This is because $d(u', u_1) \neq 1$ since $d(u, u_1) = 1$ and u_1 is not a collision point, and if $1 < d(u', u_1) < k+1$, then $d(u_1, v') < k$ and hence $u \xrightarrow{x_1} u_1 \xrightarrow{y} v'$ is a path from u to v' in G of length $< k+1$ implying u is a collision vertex of degree $< k+1$, a contradiction.

Hence we have an injective mapping from the set of collision vertices of degree $k+1$ to the set of collision vertices of degree k in G, and the lemma is proved.

□

4.5.2 Query Graph Security Games

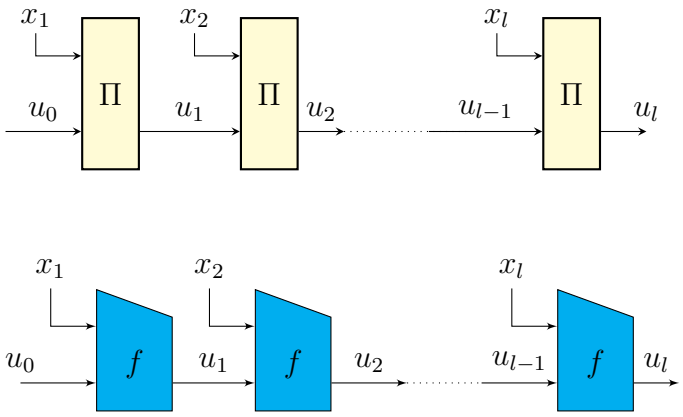


Figure 4-3: Labeled walks of length l in query graph G_Ω: $u_0 \xrightarrow{(x_1,\ldots,x_l)} u_l$. Here the adversary $\mathscr{A}$ is interacting with a random permutation Π or a random function f.

Consider an adversary $\mathscr{A}$ which makes T forward/backward queries to a random permutation $\Pi : \{0,1\}^b \to \{0,1\}^b$. Let the query transcript be of the form $\Omega' = \{(U_i, W_i, \mathsf{dir}_i)\}_{i=1}^T$, where $\mathsf{dir}_i = +$ if the i th query is a forward one and $\mathsf{dir}_i = -$ otherwise. Define $\Omega = \{(U_i, V_i, \mathsf{dir}_i)\}_{i=1}^T$ where $V_i = \lceil W_i \rceil_c$ for all $i \in [T]$. As above, we may sometimes omit the $+$ sign when we mean only forward queries and the notation Ω to also denote the set $\{(U_i, V_i)\}_{i=1}^T$.

Let S denote the set of all pairs of distinct indices such that there is a collision due to forward queries in $\mathsf{range}(\Omega)$, where Ω can be either of the above query transcripts. In notation,

$$S := \left\{ \{i, i'\} \;\middle|\; (U_i, V_i, +), (U_{i'}, V_{i'}, +) \in \Omega;\ i \neq i';\ V_i = V_{i'} \right\}.$$

Proposition 9. $\mathsf{Ex}\left[|S|\right] \leq \frac{T^2}{2^{c+1}}$.

Proof. For each $i \neq i' \in [T]$, define

$$I_{i,i'} = \begin{cases} 1 \text{ if } \{i, i'\} \in S \\ \\ 0 \text{ otherwise.} \end{cases}$$

Then,

$$\begin{aligned} \mathsf{Ex}\left[|S|\right] &= \mathsf{Ex}\left[\sum_{i,i' \in [T]} I_{\{i,i'\}}\right] \\ &= \sum_{i,i' \in [T]} \mathsf{Ex}\left[I_{\{i,i'\}}\right] \\ &\leq \sum_{i \neq i' \in [T]} \Pr\left[V_i = V_{i'}\right] \\ &\leq \frac{T^2}{2^{c+1}}. \end{aligned}$$

□

Now we consider the directed graph G_Ω for both the transcripts and have a look at their collision vertices. In this case, since the adversary can make adaptive backward queries, every vertex of $V(G_\Omega)$ can be a collision vertex with a probability of 1. Hence we are not interested in estimating the number of collision vertices in the whole of the query graph. Instead, we consider a particular family of subgraphs of the query graph with a fixed set of labels and try to estimate the expected number of collision vertices in each of these subgraphs.

For any set of labels $\chi := \{x_i \big| \, x_i \in \{0,1\}^{b-c} \ \forall i \in [l]\}$ define $G^\chi \subseteq G_\Omega$ to be the

subgraph generated by the labels in χ. Our objective is to bound the number of collision vertices in G^{χ} for any $\chi \subseteq \{0,1\}^{b-c}$. To do this, we start by defining a bad event with respect to the query transcript Ω.

BBAD : There exists $x \in \chi$ and $\{(U_{i_1}, V_{i_1}, -), \ldots, (U_{i_{n_1+1}}, V_{i_{n_1+1}}, -)\} \subseteq \Omega$ such that $\lfloor U_{i_k} \rfloor_{b-c} = x$ for all $k \in [1, n_1 + 1]$.

Proposition 10. *(bounding* BBAD*)*

$$\Pr[\mathtt{BBAD}] \leq \frac{\mathsf{mcoll}(T, 2^{b-c})}{n_1}.$$

Proof. This follows from the definition of $\mathsf{mcoll}(T, 2^{b-c})$ and Markov's inequality. □

Proposition 11. *Given any $\chi \subseteq \{0,1\}^{b-c}$, let ξ_k^{χ} be the number of collision vertices of degree k in the directed graph G^{χ} as defined above. If* BBAD *doesn't occur, then*

$$\mathsf{Ex}[\xi_k^{\chi}] \leq 2n_1|\chi| + \frac{T^2}{2^c}.$$

Proof. We start by bounding the number of collision vertices of degree 1.

Consider the subgraph G_f of G corresponding to only the forward queries. Observe that since the adversary has control over U_i for all the forward queries, it can restrict its forward queries in such a way that every edge in G_f has a label in the set χ i.e. G_f is a subgraph of G^{χ}.

Now a vertex $u \in V(G_f)$ is a collision vertex of degree 1 in G^{χ} if and only if one of the following events occurs.

- u is a collision vertex of degree 1 in G_f. A similar analysis as before bounds the

number of such u by $2|S|$.

- u is not a collision vertex of degree 1 in G_f but there exists an edge $u \xrightarrow{x} v$ in G_f and a backward query of the form $(U_i, v, -) \in \Omega$ with $\lfloor U_i \rfloor_{b-c} \in \chi$. Since `BBAD` doesn't occur number of such u is bounded above by $n_1 \cdot |\chi|$.

Since `BBAD` doesn't occur we have $|V(G^\chi) \setminus V(G_f)| \leq n_1 l$.

Hence we have $\xi_1^\chi \leq 2n_1 \cdot |\chi| + 2|S|$. Using Lemma 6, Proposition 9 and linearity of expectation we get Proposition 11.

□

4.5.3 Bounding μ_{q_p} for Sponge AEAD

Given any AEAD adversary $\mathscr{A}$ of Sponge, let $\Theta := \{(U_i, V_i, \mathsf{dir}_i)\}_{i\in[q_p]}$ denote the primitive transcript received by $\mathscr{A}$. By abuse of notation let Θ also denote the set $\{(U_i, V_i)\}_{i\in[q_p]}$. Let $L : \{0,1\}^b \to \{0,1\}^b$ be the linear function defined by $L(v) = \lceil v \rceil_c \| 0^{b-c}$. Consider the multichain graph G_Θ generated by the linear function L in Θ. Note that if a set of walks $\{\mathcal{W}_1, \ldots, \mathcal{W}_p\}$ is a multi-chain then so is any subset of it. Also there can be different multi-chains depending on the starting and ending vertices and different $x = (x_1, \ldots, x_k)$.

Note that in the security proof of TtP, the analysis of multi-chain graph appears only in the analysis of decryption query (see Section 4.3.4). In the last block processing the most significant c bits of the previous chain output is simply xor-ed with a constant (which depends on the Sponge construction and not on the ciphertext) before forwarding as the last chain input. Hence while bounding μ_{q_p} in Corollary 6, it is enough to consider only the multi-chains in G_Θ with labels $(x_1, \ldots, x_k)$, $k \le \sigma_d$ such that $\lceil x_j \rceil_c = \begin{cases} \delta_M & \text{if } j = k \\ 0^c & \text{otherwise.} \end{cases}$.

We redefine W_k to denote the maximum order of all such multi-chains of length $k \le \sigma_d$.

$$\mu_{q_p} := \max_{\mathscr{A}} \max_{k\le\sigma_d} \mathsf{Ex}\left[\frac{W_k}{k}\right].$$

Finally, we define an event we call TBAD.

TBAD : There exists $x \in \{0,1\}^{b-c}$ and

$$\{(U_{i_1}, V_{i_1}, -), \ldots, (U_{i_{n_2+1}}, V_{i_{n_2+1}}, -)\} \subseteq \Theta$$

such that $\lfloor U_{i_k} \rfloor_{b-c} = x$ for all $k \in [1, n_1 + 1]$.

Lemma 7.

$$\Pr[\texttt{TBAD}] \leq \frac{\mathsf{mcoll}(q_p, 2^{b-c})}{n_2}.$$

Proof. This follows from the definition of mcoll and Markov's inequality. □

Given a fixed label $(x_1, \ldots, x_k)$ $k \leq \sigma_d$ and $p \in \mathbb{N}$, let $\{V_0^j \overset{(x_1,\ldots,x_k)}{\longrightarrow} V_k^j\}_{j\in[p]}$ be a multi-chain of length k. Let $x_0 := \lfloor V_0^j \rfloor_{b-c}$, $v_k := \lfloor V_k^j \rfloor_\tau$.
Define $\chi = \{x_0, \lfloor x_1 \rfloor_{b-c}, \ldots, \lfloor x_{k-1} \rfloor_{b-c}\}$. Then $|\chi| \leq \sigma_d$.

Now consider the query graph G_Ω^χ generated by $\omega = \{(U, \lceil V \rceil_c) \mid (U, V) \in \Theta\}$.

Proposition 12. *If* TBAD *doesn't occur in* Θ *and* BBAD *doesn't happen in* Ω *then*

$$\mu_{q_p} \leq 2n_1\sigma_d + n_2 + \frac{q_p^2}{2^c} + \mathsf{mcoll}(q_p, 2^\tau).$$

Proof. Note that $V \xrightarrow{x} V'$ in G_Θ, such that $\lceil x \rceil_c = 0$ implies $\lceil V \rceil_c \overset{\lfloor x \rfloor_{b-c}}{\longrightarrow} \lceil V' \rceil_c$ in G_Ω^χ.

Hence for each $j \in [p]$ there exist a unique walk $\lceil V_0^j \rceil_c \overset{(x_0, \lfloor x_1 \rfloor_{b-c}, \ldots, \lfloor x_{k-1} \rfloor_{b-c})}{\longrightarrow} v^j$ in G_Ω^χ such that $(v^j \| 0^{b-c} \oplus x_k, V_k^j) \in \Theta$.

Let $v \in \{0,1\}^c$ be such that $(v \| 0^{b-c} \oplus x_k, * \| v_k) \in \Theta$. Then v must satisfy one of the conditions bellow:

- for all $j \in [p]$, $v^j \neq v$.

- There exists $j \in [p]$ such that $v^j = v$ is a collision point in G^{χ}_{Ω}.

- There exists $j \in [p]$ such that $v^j = v$ is not a collision point in G^{χ}_{Ω}.

Note that if $v^j = v$ for some $j \in [p]$ is a collision point in G^{χ}_{Ω} then $\lceil V_0^j \rceil_c$ is a collision vertex of degree $\leq k-1$ in G^{χ}_{Ω}. Again if $v^j = v$ for some $j \in [p]$ is not a collision point in G^{χ}_{Ω} then $\lceil V_0^j \rceil_c \neq \lceil V_0^{j'} \rceil_c$ for all $j' \neq j \in [p]$. Hence,

$$p \leq \left|\{v \in \{0,1\}^c \mid (v \| 0^{b-c} \oplus x_k, * \| v_k) \in \Theta\}\right| + \sum_{k' \leq k-1} \xi^{\chi}_{k'}.$$

Since $\{V_0^j \xrightarrow{(x_1,\ldots,x_k)} V_k^j\}_{j \in [p]}$ is any arbitrary multi-chain of length k in G_{Θ} we have, for all k,

$$W_k \leq \left|\{v \in \{0,1\}^c \mid (v \| 0^{b-c} \oplus x_k, * \| v_k) \in \Theta\}\right| + (k-1)\xi^{\chi}_1.$$

Hence,

$$\mu_{q_p} \leq \mathsf{Ex}\left[\left|\{v \in \{0,1\}^c \mid (v \| 0^{b-c} \oplus x_k, * \| v_k) \in \Theta\}\right|\right] + \mathsf{Ex}\left[\xi^{\chi}_1\right]. \tag{4.4}$$

Now it remains to bound $\mathsf{Ex}\left[\left|\{v \in \{0,1\}^c \mid (v \| 0^{b-c} \oplus x_k, * \| v_k) \in \Theta\}\right|\right]$.

Lemma 8. *If* `TBAD` *doesn't occur in* Θ *then,*

$$\mathsf{Ex}\left[\left|\{v \in \{0,1\}^c \mid (v \| 0^{b-c} \oplus x_k, * \| v_k) \in \Theta\}\right|\right] \leq n_2 + \mathsf{mcoll}(q_p, 2^{\tau}).$$

Proof. Since TBAD doesn't occur in Θ,

$$\left|\{v \in \{0,1\}^c \mid (v \| 0^{b-c} \oplus x_k, * \| v_k, -) \in \Theta\}\right| \leq n_2.$$

Now, by the definition of $\mathsf{mcoll}(q_p, 2^\tau)$,

$$\left|\{v \in \{0,1\}^c \mid (v \| 0^{b-c} \oplus x_k, * \| v_k, +) \in \Theta\}\right| \leq \mathsf{mcoll}(q_p, 2^\tau).$$

□

Proposition 12 follows from Eq. 4.4, Proposition 11 and Lemma 8.

□

4.5.4 Security Bound for Sponge AEAD

Consider any AEAD adversary making q_p permutation queries, q_e encryption queries with a total number of σ_e data blocks and q_d encryption queries with a total number of σ_d data blocks. Define $\sigma := \sigma_e + \sigma_d$. Then given any positive real number n_1, we can upper bound the AEAD advantage of the Sponge construction in the following way:

Corollary 7. *For any positive integers* n_1, n_2

$$\begin{aligned}\mathbf{Adv}^{\mathrm{AEAD}}_{\mathit{Sponge}}(\sigma_e, \sigma_d, q_p) \leq \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma(2\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{\sigma_e + q_p}{2^b} \\ + \frac{\mathsf{mcoll}(q_p, 2^{b-c})}{n_1} + \frac{\mathsf{mcoll}(q_p, 2^{b-c})}{n_2} + \frac{2q_p\mathsf{mcoll}(\sigma_e, 2^r)}{2^c} \\ + \frac{n_1\sigma_d^2}{2^c} + \frac{n_2\sigma_d}{2^c} + \frac{\sigma_d q_p^2}{2^{2c}} + \frac{\sigma_d\mathsf{mcoll}(q_p, 2^\tau)}{2^c} \\ + \frac{q_p\mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} + \frac{q_p\sigma_d\mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.\end{aligned}$$

4.5.5 Interpretation of Corollary 7

For practical purposes assume that $\sigma \ll q_p \ll \min\{2^\tau, 2^c\} \leq \max\{2^\tau, 2^c\} \ll 2^b$. Then the dominating terms in Corollary 7 is $\frac{n_1\sigma^2}{2^c} + \frac{\sigma q_p^2}{2^{2c}} + \frac{\mathsf{mcoll}(q_p, 2^{b-c})}{n_2} + \frac{\mathsf{mcoll}(q_p, 2^{b-c})}{n_1}$. If $q_p \ll 2^{b-c}$ take $n_1 = n_2 = b - c$, else take $n_1 = n_2 = \frac{(b-c)q_p}{2^{b-c}}$. Ascon, which is a finalist in the NIST LwC competition, has two variants both of which uses a tag size of 128 bits. Assume $q_p \leq 2^{112}$ and $(b - c)\sigma \leq 2^{53}$ as prescribed by NIST [93]. Hence, plugging in the Ascon parameters [48] in Corollary 7 we conclude that both the variants can achieve security by using a capacity of size $c \geq 140$. In other words, Ascon128 (resp. Ascon128a) is secured within NIST parameters with an underlying permutation of state size $b \geq 204$ (resp. $b \geq 205$).

4.6 Matching Attack on **Transform-then-Permute**

Now we see some matching attacks for the bound. We explain the attacks for the simplified version (by considering empty associated data).

1. Suppose μ_{q_p} maximizes for some adversary $\mathcal{B}$ interacting with Π. Now, the AE algorithm $\mathcal{A}$ will run the algorithm $\mathcal{B}$ to get the primitive transcript ω_p. We first make q_d many encryption queries with single block messages with distinct nonces $N_1, \ldots, N_{q_d}$ and hence for all $1 \leq i \leq q_d$, $\lceil Y_{i,0} \rceil_r$, $\lceil X_{i,1} \rceil_r$ and $\lceil Y_{i,1} \rceil_\tau$ values are known. Suppose for length m_i, the multi-chain for the graph induced by ω_p start from the nodes (whose r most significant bits of the domain is u_i) to the nodes (whose τ most significant bits of the range is T_i) and with label x_i. Now we choose the appropriate ciphertext C_1^* such that $\lceil X_{i,1}^* \rceil_r = u_i$. Moreover, we choose $C_{i,j}^*$ such that $\overline{C_{i,j}^*}$ is same as $x_{i,j}$ (here we assume that $\mathcal{B}$ makes queries so that the labels are compatible with encoding function).

   Now, we make decryption queries (N_i, C_i^*, T_i). With probability $\mathsf{W}_{m_i}/2^c$, the ith forgery attempt would be successful. Then maximizing $\frac{\mathsf{W}_{m_i}}{m_i}$ and by taking expectation, we achieve the desired success probability.

2. Guessing the key K through primitive query would lead to a key-recovery and hence all other attacks. The correct guess of the key can be easily detected by making some more queries for each guess to compute an encryption query. This attack requires $q_p = \mathcal{O}(2^\kappa)$. Similarly, random forging gives a success probability of forging about $\mathcal{O}(q_d/2^\tau)$.

3. Another attack strategy can be adapted to achieve $\sigma_e q_p/2^b$ bound. We look for a

collision among X-values and primitive-query inputs. This can be again detected by adding one or two queries to each guess. The same attack works with success probability $q_p\mathsf{mcoll}(\sigma_e, 2^r)/2^c$ if we make primitive queries after making all encryption queries.

4. A similar attack strategy can be adapted to achieve $q_p\mathsf{mcoll}(\sigma_e, 2^r)/2^{b-\tau}$ bound. We look for a collision among T-values and primitive-query inputs where primitive queries are done after the encryption queries to predict the unknown $b-\tau$ bits of the final output value.

These attacks show that the bounds in Theorem 5 and Corollary 6 are tight.

4.7 Conclusion

In this chapter, we defined a general Sponge-type construction called Transform-then-Permute. We showed that the Sponge-duplex AEAD construction and many other popular Sponge-type constructions such as PHOTON-Beetle, SpoC can be viewed as instantiations of Transform-then-Permute. We also analyzed the security of Transform-then-Permute and, consequently, derived tight security bounds for PHOTON-Beetle and SpoC, which shows that both of these constructions are well secure within the NIST prescribed bounds. We also derive an upper bound for general Sponge AEAD.

Note that all the security bounds in this chapter necessitate a high value of $c = b - r$; hence, the security of all constructions mentioned in this chapter dictates a low rate of message absorption per permutation call. This is a limitation of such constructions as it dictates many permutation calls per encryption/decryption query. In the next chapter, we will overcome this limitation and try to construct Sponge-type constructions that can achieve the maximum possible message absorption rate without compromising security.

Chapter 5

frTtP AEAD: Design and Analysis

5.1 Introduction

In this chapter we concentrate our focus towards increasing the rate part in Sponge-type constructions. An increase in the rate part of a Sponge-type construction leads to less number of permutation calls per encryption/decryption query, which directly impacts the runtime of such constructions. For example, if the rate of message absorption is doubled, the number of permutation calls decreases by a factor of 2, and consequently (ignoring the other factors), the runtime of the encryption/decryption also decreases by a factor of 2. The runtime attains its minimum when the rate of message absorption reaches its maximum. But, this increase in rate doesn't come without consequences (security degradation, for example). Consider, for example, the Transform-then-Permute construction with encryption feedback function L_e as defined in Chapter 4. We say that a Transform-then-Permute AEAD has Full-rate if $r = b$. We start by showing that a Full-rate Transform-then-Permute construction is not secure.

Proposition 13. *A full-rate Transform-then-Permute AEAD is not secure.*

Proof. Consider a TtP construction with full-rate i.e. $r = b$. Then the encryption feedback function $\mathcal{E}$ can be written as

$$\mathcal{E} = \begin{bmatrix} \mathcal{E}_1 & \mathcal{E}_3 \\ I_b & I_b \end{bmatrix}$$

The weakness of the construction comes from the fact that at each internal state during the encryption query the previous permutation output can be completely recovered as $Y = M \oplus C$. More formally an adversary $\mathscr{A}$ can forge as follows.

- $\mathscr{A}$ makes encryption queries of the form (N, M) where $|N| = b - \kappa, |M| = b$ to recieve response (C, T).
- $\mathscr{A}$ computes $D_1 = \mathsf{Fmt}(\phi, M)$,$Y_0 = C \oplus D_1$.
- $\mathscr{A}$ choses $C' \in \{0,1\}^b$ such that $C' \neq C$ and makes a forging query of the form (N, C', T') where $D'_1 = \mathsf{Fmt}(\phi, C')$, and $T' = \lceil \Pi\,(Y_0 \oplus D'_1) \rceil_\tau$.

□

Hence to secure a full-rate Sponge-type construction it is necessary to introduce an extra secrete state. In Section 5.1 we define a Full-rate Transform-then-Permute (frTtP in short) construction using extra state in details. We also talk about the security of such a construction and how it should be interpreted in the context of the NIST LwC competition. Then in Section 5.5 we give an instantiation of an frTtP construction called ORANGE-Zest which was submitted to the NIST LwC competition. We then discuss the weakness of the original ORANGE-Zest proposal and make necessary modifications to achieve a fully secured variant of it. Finally, in Section 5.6 we consider some other popular full-rate feedback functions available in the literature and try to construct secure frTtP constructions using these feedback functions.

5.2 Full-rate Transform-then-Permute AEAD with Extra State

We now define a Full-rate Transform-then-Permute (frTtP in short) AEAD mode which uses an s-bit extra secret state. The necessity of this extra-state is evident from Proposition 13. As before consider a frTtP encryption protocol with a permutation Π of state size b bits, key size κ, nonce size $b - \kappa$ and tag size τ.

Initialization: Given any encryption query of the form (N, A, M) the encryption algorithm generates $(D_1, \ldots, D_a, \ldots, D_{a+m}) := \mathsf{Fmt}(A, M)$ and defines $X_0 = K \| N; Y_0 = \Pi(\mathsf{X_0})$. Additionally the algorithm uses an extra-state initialization protocol to generate the initial extra-state S_0.

Data Processing and Ciphertext Generation: For $i \in [1, l]$ and a linear feedback function $\mathcal{E} : \{0,1\}^{2b+s} \rightarrow \{0,1\}^{2b+s}$, the algorithm recursively calculates Y_i, S_i, C_i as follows:

$$(X_i, S_i, C_i) = \mathcal{E}(Y_{i-1}, S_{i-1}, D_i);\ Y_i = \Pi(X_i).$$

Ciphertext and Tag Generation: Finally the protocol outputs $\lfloor C_l \| \cdots \| C_{a+1} \rfloor_{|M|}$ as the ciphertext and $\lfloor Y_l \rfloor_\tau$ as the tag.

5.2.1 Formal Representation of the Feedback Function

Let b, s, τ denote the permutation state size and the extra state size of a Full-rate Transform-then-Permute mode that uses an extra state.

Let $\mathcal{E} : \{0,1\}^{2b+s} \to \{0,1\}^{2b+s}$ be any feedback function defined by $\mathcal{E}(Y, S, M) = (X, S', C)$. By abuse of notation, let $\mathcal{E} = \begin{bmatrix} \mathcal{E}_1 & \mathcal{E}_2 & \mathcal{E}_3 \\ \mathcal{E}_4 & \mathcal{E}_5 & \mathcal{E}_6 \\ \mathcal{E}_7 & \mathcal{E}_8 & \mathcal{E}_9 \end{bmatrix}$ be the transformation matrix such that

$$\begin{bmatrix} X \\ S' \\ C \end{bmatrix} = \mathcal{E} \begin{bmatrix} Y \\ S \\ M \end{bmatrix}.$$

Further, define $\mathcal{D} = \begin{bmatrix} \mathcal{D}_1 & \mathcal{D}_2 & \mathcal{D}_3 \\ \mathcal{D}_4 & \mathcal{D}_5 & \mathcal{D}_6 \\ \mathcal{D}_7 & \mathcal{D}_8 & \mathcal{D}_9 \end{bmatrix}$ such that,

$$\begin{bmatrix} X \\ S' \\ M \end{bmatrix} = \mathcal{D} \begin{bmatrix} Y \\ S \\ C \end{bmatrix} \Leftrightarrow \begin{bmatrix} X \\ S' \\ C \end{bmatrix} = \mathcal{E} \begin{bmatrix} Y \\ S \\ M \end{bmatrix}.$$

Necessary Properties of the Encryption and Decryption Submatrices

Proposition 14. *If $\mathcal{E}$ and $\mathcal{D}$ are the encryption and decryption feedback function of a secured frTtP construction, then $\mathcal{E}, \mathcal{D}$ must satisfy the following conditions.*

(C1) $\mathsf{rank}(\mathcal{E}_9) = \mathsf{rank}(\mathcal{D}_9) = b$,

(C2) $\mathsf{rank}(\mathcal{E}_8) = \mathsf{rank}(\mathcal{D}_8) \neq 0$.

(C3) $\mathsf{rank}(\begin{bmatrix}\mathcal{E}_7 & \mathcal{E}_8\end{bmatrix}), \mathsf{rank}(\begin{bmatrix}\mathcal{D}_7 & \mathcal{D}_8\end{bmatrix}) = b$.

Proof. (C1) follows from the observation that if $\mathsf{rank}(\mathcal{E}_9) \neq b$, then there exists $M \neq M'$ such that $\mathcal{E}_9 \cdot M = \mathcal{E}_9 \cdot M'$, and hence the decryption function will not be deterministic. $\mathsf{rank}(\mathcal{D}_9) = b$ follows from a similar argument.

(C2) follows from the fact that if $\mathsf{rank}(\mathcal{E}_8) = 0$ or $\mathsf{rank}(\mathcal{D}_8) = 0$ then the internal Y state values are completely determined and hence the adversary can forge the construction in the same way as the frTtP construction with no extra-state.

For (C3), suppose $\mathsf{rank}(\begin{bmatrix}\mathcal{E}_7 & \mathcal{E}_8\end{bmatrix}) \neq b$. Then, there exists a non zero vector γ such that $\gamma \cdot (\begin{bmatrix}\mathcal{E}_7 & \mathcal{E}_8\end{bmatrix}) = 0$. Hence, $\gamma \cdot C = \gamma \cdot M$ with probability 1. □

Our Simplified Assumptions on the Encryption and Decryption Submatrices

Now we make some further assumptions on the feedback function $\mathcal{E}$ and try to justify why these assumptions can be made.

(P1) $\mathcal{E}_9 = \mathcal{D}_9 = I_b$,

(P2) $\mathcal{E}_6 = \mathcal{D}_6 = 0$.

(P3) $\mathcal{E}_7 = \mathcal{D}_7 = I_b$.

To justify (P1), since $\mathsf{rank}(\mathcal{E}_9) = b$ one can simply define $M' = \mathcal{E}_9 \cdot M$, and proceed with that.

For (P2), observe that since M is known, it doesn't contribute to the randomness of the extra state, and hence taking $\mathcal{E}_6 = 0$ doesn't affect the security of the AEAD scheme. $\mathcal{D}_6 = 0$ follows from $\mathcal{E}_6 = 0$ and assumption (P1).

Finally, we admit that (P3) is a much stronger assumption than the necessary condition (C3) and is made just for the simplicity of calculations in a very special class of general frTtP feedback functions. We do not have any matching attack on frTtP to justify (P3). Nonetheless, as we will see in Section 5.6, many full-rate feedback functions used in popular constructions such as COFB, HyENA satisfy this condition. Moreover, in the feedback functions used in Transform-then-Permute constructions without the extra state, (P3) is a necessary condition.

Assuming (P3), define $\Pi' = \mathcal{E}_7 \cdot \Pi$. Then, through a proper modification of $\mathcal{E}$ we may assume $\mathcal{E}_7 = I_b$.

From assumptions (P1), (P2), (P3) and simple linear algebraic calculations, we can make the following observations :

$$\mathcal{D}_1 = \mathcal{E}_1 \oplus \mathcal{E}_3;\ \mathcal{D}_2 = \mathcal{E}_2 \oplus \mathcal{E}_3 \cdot \mathcal{E}_8;\ \mathcal{D}_i = \mathcal{E}_i\ \forall 3 \leq i \leq 9. \tag{5.1}$$

With these simplifications, we can represent our frTtP feedback function $\mathcal{E}$ by Figure 5-1.

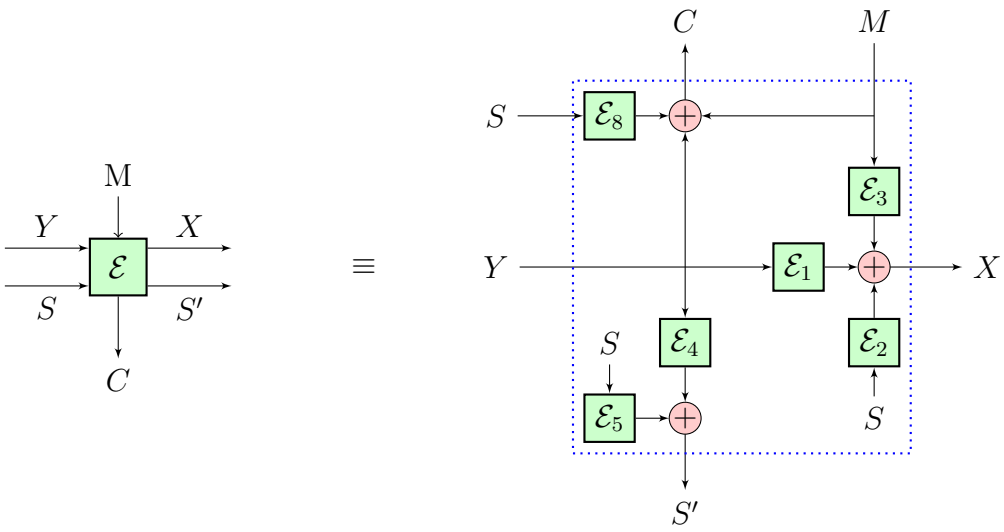


Figure 5-1: Simplified Representation of an frTtP feedback function.

5.2.2 Extra state Generation Protocol

Consider a frTtP construction with extra state size s and linear feedback function $\mathcal{E}$ as defined above. Note that, during associated data processing no information is leaked. The weakness of the construction comes from the information leaked due to the ciphertext output during message processing. More formally, suppose an adversary makes an encryption query of the form (N, A, M) such that $\mathsf{Fmt}(A, M) = (D_1, \cdots, D_a, D_{a+1}, D_{a+m})$ to recieve response $(C_{a+m} \| \cdots \| C_{a+1}, T)$ then even if $S_i = 0$ for all $0 \leq i \leq a-1$, the adversary cannot compute Y_i.

Proposition 15. *For any encryption query (N, A, M) define $a := \lceil |A|/b \rceil$. If S_a is independent or linearly dependent on N then there exist a forging adversary against the frTtP construction.*

Proof. For $A \in \{0,1\}^\star, M \in \{0,1\}^b$, suppose an adversary makes two encryption queries $(N^1, A, M) \neq (N^2, A, M)$ such that for $i = 1, 2$, $\mathsf{Fmt}(A, M) = (D_1, \ldots, D_a, D_{a+1})$. Now

suppose S_a^i is independent of N^i. Then clearly $S_a^1 = S_a^2$. Let $(C^1, T^1), (C^2, T^2)$ be the respective query responses.

Then we have

$$Y_a^1 = \mathcal{E}_8 \cdot S_a^1 \oplus C^1 \oplus D_{a+1};\ Y_a^2 = \mathcal{E}_8 \cdot S_a^1 \oplus C^2 \oplus D_{a+1}.$$

which implies

$$Y_a^1 = Y_a^2 \oplus C^1 \oplus C^2.$$

i.e.,

$$X_{a+1}^1 = \mathcal{D}_1 \cdot Y_a^1 \oplus \mathcal{D}_2 \cdot S_a^1 \oplus \mathcal{D}_3 \cdot C^1 = \mathcal{D}_1 Y_a^2 \oplus \mathcal{D}_2 \cdot S_a^1 \oplus \mathcal{D}_1(C^1 \oplus C^2) \oplus \mathcal{D}_3 \cdot C^1.$$

Hence if an adversary chooses $C^* \in \{0,1\}^b$ in such a way that $\mathcal{D}_3 \cdot (C^* \oplus C^1) = \mathcal{D}_1 \cdot (C^1 \oplus C^2)$ then (N^2, A^2, C^*, T^1) is a valid forgery.

The non-linearity follows through a similar analysis with $S_a^1 \oplus S_a^2 = \mathcal{F} \cdot (N^1 \oplus N^2)$ where $\mathcal{F}$ is some $s \times \nu$ linear matrix. Here (N^2, A^2, C^*, T^1) is a valid forgery where $\mathcal{D}_3 \cdot (C^* \oplus C^1) = \mathcal{D}_1 \cdot (C^1 \oplus C^2 \oplus \mathcal{E}_8 \cdot \mathcal{F} \cdot (N^1 \oplus N^2)) \oplus \mathcal{D}_2 \cdot \mathcal{F} \cdot (N^1 \oplus N^2)$.

□

Proposition 16. *For any encryption query (N, A, M) define $a = \lceil |A|/b \rceil$. If S_a is linear function of Y_a, then there exist a distinguishing adversary against the* frTtP *construction.*

Proof. Suppose $S_a = \rho(Y_a)$. Then $(I_b \oplus \mathcal{E}_8 \cdot \rho)Y_a = C_{a+1} \oplus D_{a+1}$. If $\mathsf{rank}(I_b \oplus \mathcal{E}_8 \cdot \rho) = b$ then Y_a can be calculated as $(I_b \oplus \mathcal{E}_8 \cdot \rho)^{-1} \cdot (C_{a+1} \oplus D_{a+1})$. If $\mathsf{rank}((I_b \oplus \mathcal{E}_8 \cdot \rho)) < b$

then there exists vector γ such that $\gamma \cdot (I_b \oplus \mathcal{E}_8 \cdot \rho) = 0$ which implies $\gamma \cdot C_{a+1} = \gamma \cdot D_{a+1}$ with probability 1. □

Note that for an encryption query (N, A, M) if $a = \lceil |A|/b \rceil$ then it is not necessary to generate the extra-state values $S_0, \ldots, S_{a-1}$. However, since an adversary can make encryption/decryption query without associated data, for uniformity of the construction we assume that the extra state is initialized in the beginning for every encryption/decryption query. If we assume that the underlying primitive Π is the only nonlinear component in our frTtP construction then given any linear functions $\rho : \{0,1\}^b \to \{0,1\}^s$, $\rho' : \{0,1\}^\kappa \times \{0,1\}^\nu \to \{0,1\}^b$, a natural choice for the initial extra-state would be $\rho \circ \Pi \circ \rho'(N, K)$. By Proposition 16, $\rho'(N, K) \neq K \| N$ which is the case in many AEAD protocols such as CoFB [36]. However, we can assume $\rho'(N, K) \neq K \| N$ if $|A| > 0$ for all (N, A, M). One way to ensure $|A| > 0$ for all (N, A, M) is to redefine $(D_1, \ldots, D_l) := \mathsf{Fmt}(\mathsf{pad}(A), Z)$ where $\mathsf{pad}(A) = 0^{b-x-1}1 \| A, x = |A| \bmod b$ for all encryption queries of the form (N, A, Z) and decryption queries of the form (N, A, Z, T). From now on by abuse of notation we write $\mathsf{Fmt}(A, M)$ to mean $\mathsf{Fmt}(\mathsf{pad}(A), M)$ and take $\rho'(N, K) = K \| N$.

The frTtP structure is depicted in Figure 5-2 where the initial extra state is calculated as $S_0 = \rho(Y_0)$, for some linear function $\rho : \{0,1\}^b \to \{0,1\}^s$.

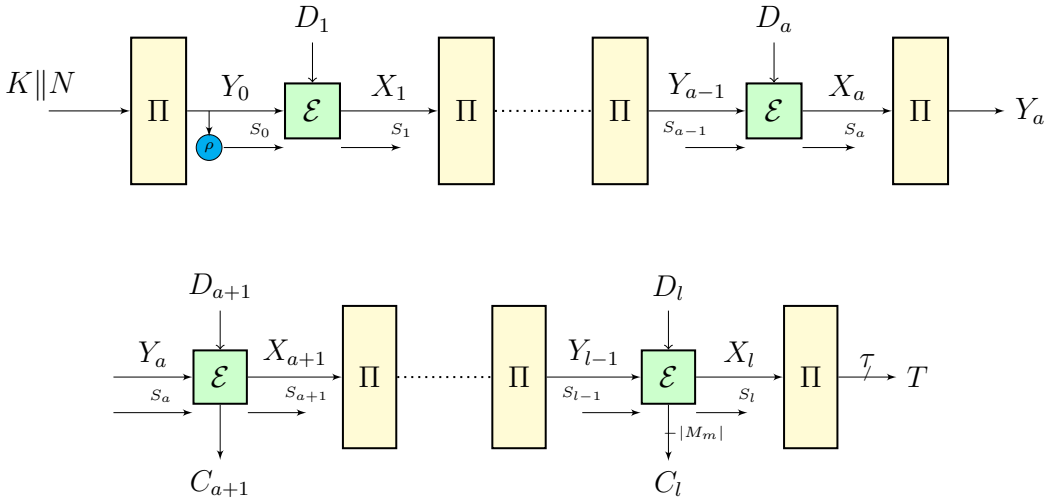


Figure 5-2: A frTtP AEAD with extra state. Here $(D_1, \ldots, D_l) = \mathsf{Fmt}(A, M)$. $\rho : \{0,1\}^b \to \{0,1\}^s$ is a linear function of rank s.

5.3 Security of frTtP AEAD with Extra State

In this section, we try to bound the advantage of any AEAD adversary making q_p many primitive queries, q_e many encryption queries with a total of σ_e many blocks and q_d many decryption queries with a total of σ_d many blocks against the frTtP construction defined in Section 5.1.

Consider an frTtP construction with the encryption and decryption feedback functions $\mathcal{E}$ and $\mathcal{D}$ respectively. We consider a linear function $\rho : \{0,1\}^b \to \{0,1\}^s$ of rank s for processing the initial extra-state with transformation matrix ρ (by abuse of notation). Define

$$r_3 := \mathsf{rank}(\mathcal{E}_3^s); r_8 = \mathsf{rank}(\mathcal{E}_8^s);$$

$$r_{12} := \mathsf{rank}\left((\mathcal{E}_1 \cdot \mathcal{E}_8 \oplus \mathcal{E}_2)^s\right); r_{45} := \mathsf{rank}\left((\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^s\right);$$

$$r'_{45} := \mathsf{rank}\left((I_s \oplus \mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^s\right).$$

Further define,

$$r_e := \mathsf{rank}\left(\begin{bmatrix}\mathcal{E}_1 & \mathcal{E}_2\\ \mathcal{E}_4 & \mathcal{E}_5\end{bmatrix}\right); r_d := \mathsf{rank}\left(\begin{bmatrix}\mathcal{D}_1 & \mathcal{D}_2\\ \mathcal{D}_4 & \mathcal{D}_5\end{bmatrix}\right).$$

Theorem 6. *The AEAD advantage of all adversaries making q_p many primitive queries, a total of σ_e blocks in encryption queries, and a total of σ_d blocks in decryption queries against an frTtP construction with s bit extra-state as defined above, can be bounded as follows*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{AEAD}}_{\mathit{frTtP}}(q_p,\sigma_e,\sigma_d) \le{}& \frac{q_p}{2^\kappa} + \frac{10\sigma_e q_p}{2^{r_e-r_{45}}} + \frac{2\sigma_e^2}{2^{r_e-r_{45}}} + \frac{q_p\mu_{q_p}\sigma_d}{2^{r_d}} + \frac{2\sigma_d}{2^\tau} + \frac{3\sigma_d(\sigma+q_p)}{2^{r_d-r_{45}}}\\ &+ \frac{q_p\mathsf{mcoll}(\sigma_e,2^\tau)}{2^{b-\tau}} + \frac{2q_p\mathsf{mcoll}(\sigma_e,2^{b+s-r_{45}-r_8})}{2^{r_e+r_8-s-b}}\\ &+ \frac{\sigma_d\sigma_e}{2^{r_{12}+r_3+r_{45}+r'_{45}-b-2s}} + \frac{\sigma_d q_p\mathsf{mcoll}(\sigma_e,2^{b+s-r_{45}-r_8})}{2^{r_e+r_d+r_8-r_{45}-b-s}}.\end{aligned}$$

Where, $\sigma := \sigma_e + \sigma_d$ and μ_{q_p} is defined as the adversarial advantage against the s-extended multi-chain graph formed by adversarial primitive queries and linear function $L_d := \begin{bmatrix}\mathcal{D}_1 & \mathcal{D}_2\\ \mathcal{D}_4 & \mathcal{D}_5\end{bmatrix}$.

5.3.1 Interpretation of Theorem 6

Consider any conventional Sponge type constructions such as Beetle. For Beetle [35], where r bits of message is absorbed per primitive call with state size b, Chakraborty et. al. [37] showed that

$$\mathbf{Adv}^{\mathrm{AEAD}}_{\mathsf{Beetle}}(q_p, \sigma) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \mathcal{O}\left(\frac{\sigma + q_p}{2^{b-r}}\right) + \mathcal{O}\left(\frac{\sigma q_p}{2^{2b-2r}}\right) + \mathcal{O}\left(\frac{\sigma q_p^2}{2^{2b-r}}\right).$$

Assuming $q_p\sigma \ll 2^b$, Beetle construction is secure with a data absorption rate $r \ll b - \log(\sigma + q_p)$.

If the feedback function of an frTtP construction is chosen in such a way that $r_8 = r_{12} = r_{45} = r'_{45} = s$, $r_3 = b$ and $r_e = r_d = b + s$. Then Theorem 6 can be re-written as

Corollary 8. *If $r_5 = r_8 = r_{12} = r_{45} = r'_{45} = s$, $r_3 = b$ and $r_e = r_d = b + s$, then*

$$\mathbf{Adv}^{\mathrm{AEAD}}_{\mathit{frTtP}}(q_p, \sigma) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \mathcal{O}\left(\frac{\sigma q_p}{2^b}\right) + \mathcal{O}\left(\frac{\sigma^2 + q_p}{2^s}\right).$$

From Corollary 8 it is evident that, If we assume $q_p\sigma \ll 2^b$ then with use of an extra state of size $s \gg \log(\sigma^2 + q_p)$, we can construct a secured full-rate Sponge type AEAD schemes where maximum possible message absorption per primitive call is achieved. Hence we observe that in such an frTtP construction the $b - r$ bit increase in the data absorption rate as compared to Beetle is compensated by using a $b - r$ bit extra-state (assuming $\sigma^2 < q_p$).

5.4 Proof of Theorem 6

5.4.1 Ideal World and Defining the Bad Transcripts

The ideal world responds to three oracles, namely encryption queries, decryption queries, and primitive queries in the online phase.

(1) On Primitive Query $(\mathsf{W}_i, \mathsf{dir}_i)$:

The ideal world simulates $\Pi^{\pm}$ query honestly. In particular, if $\mathsf{dir}_i = 1$, it sets $\mathsf{U}_i \leftarrow \mathsf{W}_i$ and returns $\mathsf{V}_i = \Pi(\mathsf{U}_i)$. Similarly, when $\mathsf{dir}_i = -1$, it sets $\mathsf{V}_i \leftarrow \mathsf{W}_i$ and returns $\mathsf{U}_i = \Pi^{-1}(\mathsf{V}_i)$.

(2) On Encryption Query $Q_i := (\mathsf{N}_i, \mathsf{A_i}, \mathsf{M}_i)$:

It samples $Y_{i,0}, \ldots, Y_{i,l_i} \xleftarrow{\$} \{0,1\}^b$ where $a_i = \lceil |\mathsf{pad}(\mathsf{A}_i)|/b \rceil$, $m_i = \lceil |\mathsf{M}_i|/b \rceil$ and $l_i = a_i + m_i$. For all $1 \leq j \leq l_i$ it then calculates

$$S_{i,j} = \begin{cases} \rho \cdot Y_{i,0} & \text{if } j = 0 \\ \mathcal{E}_5^{j-1} \cdot (\mathcal{E}_4 \oplus \mathcal{E}_5 \cdot \rho) \cdot Y_{i,0} \oplus \bigoplus_{k=1}^{j-1} \mathcal{E}_5^{j-1-k} \cdot \mathcal{E}_4 \cdot Y_{i,k} & \text{otherwise.} \end{cases}$$

Finally it returns $(\mathsf{C}_i, \mathsf{T}_i)$ where $(D_{i,1} \ldots, D_{i,l_i}) := \mathsf{Fmt}(\mathsf{pad}(\mathsf{A}_i), \mathsf{M_i})$ and $\forall\ a_i + 1 \leq j \leq l_i$

$$C_{i,j} = Y_{i,j-1} \oplus \mathcal{E}_8 \cdot S_{i,j-1} \oplus D_{i,j};$$

$$\mathsf{C}_i = \lfloor C_{i,l_i} \| \cdots \| C_{i,a_i+1} \rfloor_{|\mathsf{M}_i|}.$$

$$\mathsf{T}_i = \lfloor Y_{i,l_i} \rfloor_\tau.$$

(3) On Decryption Query $Q_i := (\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*, \mathsf{T}_i^*)$:

According to our convention we assume that the decryption query is always non-trivial. So the ideal world returns abort symbol $\mathsf{M}_i^* := \perp$.

Offline Phase of Ideal World. After completion of oracle interaction (the above three types of queries possibly in an interleaved manner), the ideal oracle sets $\mathbb{E}, \mathbb{D}, \mathbb{P}$ to denote the sets of all the query indices corresponding to the encryption, decryption and primitive queries respectively. So $\mathbb{E} \sqcup \mathbb{D} \sqcup \mathbb{P} = [q_e + q_d + q_p]$ where $|\mathbb{E}| = q_e, |\mathbb{D}| = q_d, |\mathbb{P}| = q_p$. Let the primitve transcript be $\omega_p = (U_i, V_i, \mathsf{dir}_i)_{i \in \mathbb{P}}$. Denote $\omega_p' := (U_i, V_i)_{i \in \mathbb{P}}$. Let the decryption transcript $\omega_d = (\mathsf{M}_i^*)_{i \in \mathbb{D}}$ where M_i^* is always $\perp$.

Now we describe the extended transcript for the encryption queries. It samples $K \leftarrow\!\!\$\, \{0,1\}^\kappa$. For all $i \in \mathbb{E}$ and $j \in [0, l_i]$, we define

$$X_{i,j} = \begin{cases} K \| \mathsf{N}_i & \text{if } j = 0 \\ \mathcal{E}_1 \cdot Y_{i,j-1} \oplus \mathcal{E}_2 \cdot S_{i,j-1} \oplus \mathcal{E}_3 \cdot D_{i,j} & \text{otherwise.} \end{cases}$$

where, $(D_{i,1}, \ldots, D_{i,l_i}) := \mathsf{Fmt}(\mathsf{pad}(\mathsf{A}_i), \mathsf{M}_i)$.

The encryption transcript $\omega_e = (X_{i,j}, Y_{i,j}, S_{i,j})_{i \in \mathbb{E}, j \in [0,l_i]}$. So the transcript of the adversary consists of $\omega = (Q, \omega_e, \omega_d, \omega_p)$ where $Q := (Q_i)_{i \in \mathbb{P} \sqcup \mathbb{E} \sqcup \mathbb{D}}$.

Bad Transcripts due to Encryption Queries

We now consider some events that may occur due to the primitive and encryption query transcript.

BAD1: $\exists (U, V) \in \omega_p :\ K = \lceil U \rceil_\kappa$.

BAD2: $\exists i \in \mathbb{E}, j \in [l_i]$ such that $Y_{i,j} \in \mathsf{range}(\omega'_p)$.

BAD3: $\exists\ i \in \mathbb{E},\ j \in [l_i]$ such that $X_{i,j} \in \mathsf{domain}(\omega'_p)$.

BAD4: $\exists\ (i, j) \neq (i', j')$ such that $Y_{i,j} = Y_{i',j'}$, where $i \in \mathbb{E},\ j \in [l_i],\ i' \in \mathbb{E},\ j' \in [l_{i'}]$.

BAD5: $\exists\ (i, j) \neq (i', j')$ such that $X_{i,j} = X_{i',j'}$, where $i \in \mathbb{E},\ j \in [l_i],\ i' \in \mathbb{E},\ j' \in [l_{i'}]$.

We point out that these events broadly represent some collisions in the internal states. More formally, if these events do not occur then the partial permutation function Π' generated by $\omega_p \cup \omega'_e$ is permutation compatible. Hence we call them bad events.

Bad Transcripts due to Decryption Queries

Suppose $\mathtt{BAD}_{enc}$ doesn't occur i.e. Π' as defined above is permutation compatible. Given any decryption query $(N_i^*, A_i^*, C_i^*, T_i^*), i \in \mathbb{D}$ we define $a_i^* = \lceil |\mathsf{pad}(A_i^*)|/b \rceil$, $m_i^* = \lceil |C_i^*|/b \rceil$ and $l_i^* := a_i^* + m_i^*$. Let $(D_{i,1}^*, \ldots, D_{i,l_i^*}^*) := \mathsf{Fmt}(\mathsf{pad}(A_i^*), C_i^*)$. Let p_i denotes the length of the longest common prefix for the ith decryption query with the encryption query. Formally, it is define as follows:

$$p_i = \begin{cases} -1, & \text{if } \forall i' \in \mathbb{E},\ N_i^* \neq N_{i'} \\ k, & \text{if } \exists i' \in \mathbb{E} :\ N_i^* = N_{i'},\ \forall j \in [k < l_i] : D_{i,j}^* = D_{i',j},\ D_{i,k+1}^* \neq D_{i',k+1} \\ l_i - 1, & \text{otherwise.} \end{cases}$$

Further, for all $i \in \mathcal{D}$ and $0 \leq j \leq p_i$, we define the internal states of the ith decryption query as follows:

$$\begin{aligned} X^*_{i,j} &= X_{i',j}, \\ Y^*_{i,j} &= Y_{i',j}, \\ S^*_{i,j} &= S_{i,j}, \\ X^*_{i,p_i+1} &= \mathcal{D}_1 \cdot Y^*_{i,p_i} \oplus \mathcal{D}_2 \cdot S^*_{i,p_i} \oplus \mathcal{D}_3 \cdot D^*_{i,p_i+1}, \\ S^*_{i,p_i+1} &= S_{i',p_i+1}. \end{aligned}$$

Note that by property of Fmt function, $X^*_{i,p_i+1} \neq X^*_{i',p_i+1}$.

If $a_{i'} \leq p_i$ using $Y^*_{p_i+1}$, we consider all possible labeled walks $(Y^*_{p_i+1}, S^*_{p_i+1}) \xrightarrow{(D^*_{i,p_i+2},\ldots,D^*_{i,j})} (Y^*_{i,p'_i}, S^*_{i,j})$ in the *s-extended* multi-chain graph $G^{L_d}_{\omega_p}$ where $L_d = \begin{bmatrix} \mathcal{D}_1 & \mathcal{D}_2 \\ \mathcal{D}_4 & \mathcal{D}_5 \end{bmatrix}$. Let $j_{\max}$ denote the maximum of all such js.

Next, we define a new variable p'_i in the following way.

$$p'_i = \begin{cases} p_i & \text{if } p_i < a_{i'} \text{ or } X^*_{i,p_i+1} \notin \mathsf{domain}(\omega'_p) \\ j_{\max} & \text{Otherwise.} \end{cases}$$

Finally, we define

$$\begin{aligned} X^*_{i,p'_i+1} &= \mathcal{D}_1 \cdot Y^*_{i,p'_i} \oplus \mathcal{D}_2 \cdot S^*_{i,p'_i} \oplus \mathcal{D}_3 \cdot D^*_{i,p'_i+1}, \\ S^*_{p'_i+1} &= \mathcal{D}_4 \cdot Y^*_{i,p'_i} \oplus \mathcal{D}_5 \cdot S^*_{i,p'_i}. \end{aligned}$$

With these definitions, we consider the following two events that occur due to the

decryption queries.

BAD6: $\exists i \in \mathbb{D}$, $p'_i = l_i$ and $\lfloor Y^*_{i,l_i} \rfloor_\tau = T^*_i$.

BAD7: $\exists i \in \mathcal{D}$, $p'_i \leq l_i - 1$ and $(X^*_{i,p'_i+1}) \in \mathsf{domain}(\omega_e)$.

.

5.4.2 Bounding the Bad Events

Bounding the Bad Events due to Encryption

Define,

$$\mathtt{BAD}_{enc} := \bigcup_{i=1}^{5} \mathtt{BADi}.$$

Lemma 9.

$$\Pr\left[\mathtt{BAD}_{enc}\right] \leq \frac{q_p}{2^\kappa} + \frac{10\sigma_e q_p}{2^{r_e - r_{45}}} + \frac{2\sigma_e^2}{2^{r_e - r_{45}}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_8-s}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_e+r_8-b-s}}.$$

Proof. We start by bounding the individual bad events. Lemma 9 then follows from the union bound.

Bounding BAD1: For some $(U, V) \in \omega_p$, $\quad K = \lceil U \rceil_\kappa$.

This is basically the key recovery event, i.e., the event that the adversary recovers the master key K by direct queries to the internal random permutation (can be both forward or backward). For a fixed entry $(U, V) \in \omega'_p$, the probability that $K = \lceil U \rceil_\kappa$

is bounded by at most $2^{-\kappa}$, as K is chosen uniform at random from $\{0,1\}^{\kappa}$. Thus, we have

$$\Pr[\texttt{BAD1}] \leq \frac{q_p}{2^{\kappa}}.$$

Bounding BAD2: This event can be analyzed in several cases as below

CASE 1. $\exists i, j, a$ such that $Y_{i,j} = V_a$, encryption after primitive Since $Y_{i,j}$ are chosen uniformly at random, this case can be bounded for fixed i, j, a with probability at most $1/2^b$. We have at most σ_e many (i, j) pairs and q_p many a indices. Hence this case can be bounded by at most $\sigma_e q_p/2^b$.

Case 2: $\exists i, j, a$, $Y_{i,j} = V_a$, $\mathsf{dir}_a = +$, encryption before primitive: This case can be bounded by probability at most $1/(2^b - q_p + 1)$. We have at most σ_e many (i, j) pairs and q_p many a indices. Thus this can be bounded by at most $\sigma_e q_p/(2^b - q_p + 1) \leq 2\sigma_e q_p/2^b$ (assuming $q_p \leq 2^{b-1}$).

Case 3: $\exists i$,$j < l_i$, a, $Y_{i,j} = V_a$, $\mathsf{dir}_a = -$, encryption before primitive:

For $0 \leq j < a_i$ $Y_{i,j}$ is chosen uniformly at random and no output is generated hence probability of this case is bounded by at most $2\sigma_e q_p/2^b$.

Proposition 17. *For $a_i + 1 \leq j \leq l_i$ the adversary can know at most $b - s(r_{45} - s) - r_8$ bits of $Y_{i,j+1}$.*

Proof. Note that $Y_{i,0}$ is unknown and random. For $j \geq a + 1$, with simple matrix algebra we can rewrite $Y_{i,j}$ as

$$Y_{i,j} = \mathcal{E}_8 \cdot (\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^j \cdot \rho(Y_{i,0}) \oplus \sum_{k=1}^{j-1} (\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^{k-1} \cdot \mathcal{E}_4 \cdot (C_k \oplus D_k).$$

Lemma 10. $\mathsf{rank}(\mathcal{E}_8 \cdot (\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^j \cdot \rho) \geq r_{45} + r_8 - s$.

Proof.

$$\begin{aligned} \mathsf{rank}(\mathcal{E}_8 \cdot (\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^j \cdot \rho) &\geq \mathsf{rank}(\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^s + r_8 - s \qquad \text{[by Theorem 1]} \\ &= r_{45} + r_8 - s. \end{aligned}$$

□

Proposition 17 follows from Lemma 10.

□

Let Φ_{out} denote the number of multicollisions among all $Y_{i',j'}$ values. Then,

$$\begin{aligned} \Pr[Y_{i,j} = V_a] &= \sum_{\Phi_{out}} \Pr[\text{Case 3} \mid \Phi_{out}] \cdot \Pr[\Phi_{out}] \\ &\leq \sum_{\Phi_{out}} \frac{\Phi_{out} \times q_p}{2^{r_{45}+r_8-s}} \cdot \Pr[\Phi_{out}] \\ &\leq \frac{q_p}{2^{r_{45}+r_8-s}} \times \mathsf{Ex}\left[\Phi_{out}\right] \\ &\leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_8-s}}. \end{aligned}$$

Hence

$$\Pr\left[\text{Case 3}\right] \leq 2\sigma_e q_p / 2^b + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_8-s}}.$$

Case 4: $\exists i, a$, $Y_{i,l_i} = V_a$, $\mathsf{dir}_a = -$, encryption before primitive:

This case is similar to Case 3. The only difference is that the adversary has access to $\lfloor Y_{i,l_i} \rfloor_\tau$. Hence doing a similar analysis as in the previous case we have

$$\Pr\left[\text{Case 4}\right] \leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}}.$$

Since all the cases are exhaustive we have

$$\Pr\left[\mathtt{BAD2}\right] \leq \frac{5\sigma_e q_p}{2^b} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_8-s}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}}.$$

Bounding $\mathtt{BAD3} \setminus \mathtt{BAD1}$:

Note that this event occurs if and only if there exists $i \in \mathbb{E}, j \in [1, l_i], k \in \mathbb{P}$ such that $\begin{bmatrix} \mathcal{E}_1 & \mathcal{E}_2 \end{bmatrix} \begin{bmatrix} Y_{i,j-1} \\ S_{i,j-1} \end{bmatrix} = U_k$. Since $Y_{i,j-1}$ is chosen uniformly at random and $S_{i,j-1} = \bigoplus_{k=1}^{j-2} \mathcal{E}_5^{j-2-k} \mathcal{E}_4 Y_{i,k} \oplus \mathcal{E}_5^{j-2}(\mathcal{E}_4 \oplus \mathcal{E}_5 \rho) Y_{i,0}$ implies $S_{i,j-1}$ is calculated independently of $Y_{i,j-1}$, probability of this event is bounded by $2^{b+r_{45}-r_e} \times \Pr\left[\mathtt{BAD2} \setminus \text{Case 4}\right]$. Hence,

$$\Pr\left[\mathtt{BAD3} \setminus \mathtt{BAD1}\right] \leq \frac{5\sigma_e q_p}{2^{r_e-r_{45}}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_e+r_8-s-b}}.$$

Bounding $\mathtt{BAD4}$: Since all the $Y_{i,j}$'s are chosen uniformly at random,

$$\Pr\left[\mathtt{BAD4}\right] \leq \frac{\sigma_e(\sigma_e - 1)}{2^b}.$$

Bounding BAD5: Since all $Y_{i,j}$'s are chosen uniformly at random and $S_{i,j}$ is independent of $Y_{i,j}$,

$$\Pr\left[\texttt{BAD5}\right] \leq \frac{\sigma_e(\sigma_e - 1)}{2^{r_e - r_{45}}}.$$

Summing over all the above bounds and using the union bound we have Lemma 9. □

Bounding the Bad Event Due to Decryption

Lemma 11. *Let* $L_d : \{0,1\}^{b+c} \to \{0,1\}^{b+c}$ *be the linear function with the transformation matrix* $\begin{bmatrix} \mathcal{D}_1 & \mathcal{D}_2 \\ \mathcal{D}_4 & \mathcal{D}_5 \end{bmatrix}$. *Consider the s-extended multi-chain graph* $G^{L_d}_{\omega_p}$ *and* μ_{q_p} *as defined before.*

$$\Pr\left[\texttt{BAD6}\right] \leq \frac{q_p \mu_{q_p} \sigma_d}{2^{r_d}}. \tag{5.2}$$

Proof. Suppose the event holds for the i-th decryption query and $N^*_i = N_{i'}$ for some $i' \in \mathbb{E}$. We use the *s-extended* multi-chain structure to bound the probability of BAD6. Specifically, mBAD implies that the decryption query is completed via a walk in G_{ω_p} with starting node $(Y^*_{i,p_i+1}, S^*_{i,p_i+1})$ and ending node $(V, S) \in V(G_{\omega_p})$ such that $\lfloor V \rfloor_\tau = T^*_i$. This is equivalent to the condition that $(Y^*_{i,p_i+1}, S^*_{i,p_i+1}) \xrightarrow{(D^*_{i,p_i+2}, \ldots, D^*_{i,l_i})}$ is an element of an $l_i - p_i$ length *s-extended* multi-chain with label $(D^*_{i,p_i+2}, \ldots, D^*_{i,l_i})$ terminating at some $V \in \mathsf{range}(\omega_p)$ such that $\lfloor V \rfloor_\tau = T^*_i$. Now since the adversary can make both forward/backward primitive queries, number of such V is bounded by q_p.

Hence the probability that BAD6 holds for the i-th decryption query is bounded by

$$\begin{aligned}
\Pr[\mathtt{BAD6} \mid \omega_p] &\leq q_p \times \sum_{(Y',S')} \sum_{\stackrel{(D^*_{i,p_i+2},\ldots,D^*_{i,l_i})}{\longrightarrow}_{(V,S)}} \Pr\left[Y^*_{i,p_i+1} = Y' \cap S^*_{i,p_i+1} = S'\right] \\
&\leq q_p \times \mu_{q_p}(l_i - p_i) \\
&\quad \times \Pr\left[Ł_d \cdot \begin{bmatrix} Y^*_{i,p_i} \\ S^*_{i,p_i} \end{bmatrix} \oplus \mathcal{E}_3 \cdot \begin{bmatrix} D^*_{i,p_i+1} \\ 0_{s\times 1} \end{bmatrix} = \begin{bmatrix} X' \\ S' \end{bmatrix} \middle| (X',Y') \in \Theta, S' \in \{0,1\}^s \right] \\
&\leq \frac{q_p \times \mu_{q_p}(l_i - p_i)}{2^{r_d}}.
\end{aligned}$$

Hence varying over all $i \in \mathbb{D}$, given a transcript ω_p

$$\begin{aligned}
\Pr[\mathtt{BAD6} \mid \omega_p] &\leq \sum_{i\in\mathbb{D}} \frac{q_p \times \mu_{q_p}(l_i - p_i)}{2^{r_d}} \\
&\leq \frac{q_p \mu_{q_p}}{2^{r_d}} \sum_{i\in\mathcal{D}} (l_i - p_i) \\
&\leq \frac{q_p \mu_{q_p} \sigma_d}{2^{r_d}}
\end{aligned}$$

where the last inequality follows from the fact that $\sum_{i\in\mathcal{D}}(l_i - p_i) < \sigma_d$. □

Lemma 12.

$$\Pr\left[\mathtt{BAD7} \mid \neg\mathtt{BAD}_{\mathsf{enc}}\right] \leq \frac{\sigma_e + q_p}{2^{r_d - r_{45}}} + \frac{\sigma_d \sigma_e}{2^{r_{12}+r_3+r_{45}+r'_{45}-b-2s}} + \frac{\sigma_d q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_e+r_d+r_8-r_{45}-b-s}}.$$

Proof. To bound BAD7 notice that this event can be subdivided into three cases.

Case 1: $p'_i < a_{i'}$ Since no information is leaked during the associated data processing of the encryption queries $Y_{i',0}, \ldots, Y_{i',p'_i}$ are sampled uniformly at random. Since S_{i',p'_i} is completely determined by $Y'_{i',0}, \cdots, Y_{i',p'_i-1}$ hence it is independent of Y_{i',p'_i} . Further,

$$X^*_{i,p'_i+1} = \mathcal{D}_1 \cdot Y_{i',p'_i} \oplus \mathcal{D}_2 \cdot S_{i',p'_i} \oplus \mathcal{D}_3 \cdot D^*_{i,p'_i}.$$

Hence at least $r_d - r_{45}$ bits of X^*_{i,p'_i+1} is random.

Hence $\Pr[\text{Case 1}] \leq \frac{\sigma_e+q_p}{2^{r_d-r_{45}}}$.

Case 2: $\exists i' \in \mathbb{E}, j \in [m_{i'}]$ s.t. $a_{i'} \leq p'_i = p_i$ and $X^*_{i,p_i+1} = X_{i',j}$

Note that, by definition of p'_i there exists an $i'' \in \mathbb{E}$ such that $Y_{i'',p_i} = Y^*_{i,p_i}$ and

$$S_{i'',p_i} = S^*_{i,p_i}$$

.

Hence, this event occurs if and only if

$$\mathcal{E}_3 \cdot (D^*_{i,p_i+1} \oplus \cdot D_{i',j}) = \mathcal{D}_1 \cdot (Y_{i',j-1} \oplus Y_{i'',p_i}) \oplus \mathcal{D}_2 \cdot (S_{i',j-1} \oplus S_{i'',p_i}).$$

By simple matrix algebra, this reduces to the condition

$$\mathcal{E}_3 \cdot D^*_{i,p_i+1} = (\mathcal{E}_1 \cdot \mathcal{E}_8 \oplus \mathcal{E}_2) \cdot (\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^{p_i} \cdot \big(\rho(Y_{i',0}) \oplus (\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^{j-p_i} \cdot \rho(Y_{i'',0})\big) \oplus \chi$$

for some known χ.

Now note that by definition of p_i either $i' \neq i''$ or $i' = i''$ but $j > p_i + 1$. For the first case, we have $Y_{i',0}, Y_{i'',0}$ are independent and random.

When $i' = i'', j > p_i + 1$ we have,

$$\mathcal{E}_3 \cdot D^*_{i,p_i+1} = (\mathcal{E}_1 \cdot \mathcal{E}_8 \oplus \mathcal{E}_2) \cdot (\mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^{p_i} \cdot (I \oplus \mathcal{E}_4 \cdot \mathcal{E}_8 \oplus \mathcal{E}_5)^{j-p_i} \cdot \rho(Y_{i,0}) \oplus \chi.$$

Hence, probability that any of these happens in the i th query is bounded by at most $\frac{\sigma_e}{2^{r_{12}+r_{45}+r'_{45}-2s}}$.

Further given everything else fixed there are at most 2^{b-r_3} many possible choices of D^*_{i,p_i+1}. Hence given any i this event can be bounded by at most $\frac{\sigma_e}{2^{r_{12}+r_{45}+r'_{45}-2s+r_3-b}}$. Varying over all $i \in \mathcal{D}$ we have

$$\Pr\left[\text{Case 2}\right] \leq \frac{\sigma_d \sigma_e}{2^{r_{12}+r_3+r_{45}+r'_{45}-2s-b}}.$$

Case 3: $p'_i > p_i$ and $\exists i' \in \mathbb{E}, j \in [m_{i'}]$ s.t. $X^*_{i,p'_i+1} = X_{i',j}$

This corresponds to the case when the first nontrivial decryption query block matches a primitive query and follows a partial chain before and then matches an encryption query block. Hence, doing a similar analysis as in event BAD3 the probability of this case occuring in the i th decryption query is bounded by $\frac{q_p}{2^{r_d-r_{45}}} \times \frac{m_i \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_e+r_8-s-b}}$

Summing over all $i \in \mathcal{D}$ we have

$$\Pr\left[\text{Case 3}\right] \leq \frac{\sigma_d q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_e+r_d+r_8-r_{45}-b-s}}.$$

Since all 3 cases are exhaustive we have the lemma.

□

5.4.3 Real World and Good Transcript Analysis

In the online phase, the AE encryption and decryption queries and direct primitive queries are faithfully responded to based on $\Pi^{\pm}$. Like the ideal world, after the completion of interaction, the real world returns all X-values Y -values and S-values corresponding to the encryption queries only. Note that a decryption query may return M_i which is not $\perp$.

Consider a good transcript $\omega = (\omega_p, \omega_e, \omega_d)$. Suppose for all $0 \leq j \leq p'_i$, $Y^*_{i,j}, S^*_{i,j}$ and $X^*_{i,j+1}$ are defined as before. We observe the following:

1. The tuples ω_e is permutation compatible and disjoint from ω_p. So union of tuples $\omega_e \cup \omega_p$ is also permutation compatible.

2. For all $i \in \mathcal{D}$ we have either $p'_i = l_i - 1$ and $(X^*_{i,m_i}, \star \| T^*_i) \in \omega_p \cup \omega_e$ (call it a Type-1 decryption query) or $p'_i < l_i - 1$ but $X^*_{i,p'_i+1} \notin \omega_p \cup \omega_e$(call it a Type-2 decryption query). Type-1 decryption queries would be rejected due to `BAD6` . For Type-2 decryption query, observe that X^*_{i,p'_i+1} is fresh i.e. it has never been queried before by the adversary. So $\Pi(X^*_{i,p'_i+1})$ is random over a large set. This would ensure with high probability that such decryption queries will also be rejected.

With these observations, we next analyze the good transcripts.

Good Transcript Analysis: Fix a good transcript ω. Let Θ_0 and Θ_1 denote the transcript random variable obtained in the ideal world and real-world respectively. As observed

above, all the input-output pairs for the underlying permutation are compatible. In the ideal world, all the Y values are sampled uniformly at random; the list ω_p is just the partial representation of Π and all the decryption queries are degenerately aborted. Hence we get

$$\Pr\left[\Theta_0 = w\right] \leq \frac{1}{2^{b\sigma_e}(2^b)_{q_p}}$$

Here σ_e denotes the total number of blocks present in all encryption queries including nonce. In notation $\sigma_e = \sum_{i \in \mathbb{E}} l_i$.

In the real world, for ω we denote the encryption query, decryption query, and primitive query tuples by ω_e, ω_d and ω_p, respectively. Then, we have

$$\begin{aligned}
\Pr[\Theta_1 = \omega] &= \Pr[\Theta_1 = (\omega_e, \omega_p, \omega_d)] \\
&= \Pr[\omega_e, \omega_p] \cdot \Pr[\omega_d \mid \omega_e, \omega_p] \\
&= \Pr[\omega_e, \omega_p] \cdot (1 - \Pr[\neg\omega_d \mid \omega_e, \omega_p]) \\
&\leq \Pr[\omega_e, \omega_p] \cdot \left(1 - \sum_{i \in \mathcal{D}} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right) \qquad (5.3)
\end{aligned}$$

Here we have slightly abused the notation to use $\neg\omega_{d,i}$ to denote the event that the i-th decryption query successfully decrypts and $\neg\omega_d$ is the union $\cup_{i \in \mathcal{D}_2} \neg\omega_{d,i}$ (i.e. at least one decryption query successfully decrypts). The encryption and primitive queries are mutually permutation compatible, so we have

$$\Pr_{\Theta_1}[\omega_e, \omega_p] = \frac{1}{(2^b)_{\sigma_e + q_p}} \geq \Pr_{\Theta_0}[\omega_e, \omega_p].$$

Proposition 18. $\Pr_{\Theta_1}[\neg\omega_{d,i} \mid \omega_e, \omega_p] \leq \frac{2(\sigma + q_p)}{2^{r_d - r_{45}}} + \frac{2}{2^\tau}$ *for every Type-2 decryption query.*

Proof. We recall that $\neg\omega_{d,i}$ occurs if and only if $\lfloor \Pi(X^*_{i,m_i}) \rfloor_\tau = T^*_i$ where X^*_{i,p'_i+1} is fresh. Further, for all $p'_i + 1 < j \leq l_i$, $X^*_{i,j}$ values have been defined recursively as follows

$$X^*_{i,j} = \mathcal{D}_1 \cdot \big(\Pi(X^*_{i,j-1})\big) \oplus \mathcal{D}_2 \cdot S^*_{i,j-1} \oplus \mathcal{D}_3 \cdot D^*_{i,j}.$$

Claim 1. $\Pr[X^*_{i,j} \textit{ is fresh }] \geq (1 - \frac{2(\sigma_e+q_p+m_i)}{2^{r_d-r_{45}}}) \;\; \forall\; p'_i + 1 < j \leq l_i.$

Proof. Since X^*_{i,p'_i+1} is not the last block, then the next input block may collide with some encryption or primitive input block with probability at most $\frac{\sigma_e+q_p}{2^{r_d-r_{45}}-\sigma_e-q_p}$. Applying similar argument for all the successive blocks till the last one, we get that if none of the previous block input collides then the probability that the last block input collides is at most $\frac{(\sigma_e+q_p+l_i-p'_i+2)}{2^{r_d-r_{45}}-\sigma_e-q_p-l_i+p'_i+2} \leq \frac{2(\sigma_e+q_p+m_i)}{2^{r_d-r_{45}}}$. □

Claim 2. $\Pr[\neg\omega_{d,i} \mid X^*_{i,j} \textit{ are fresh }] \leq \frac{2}{2^\tau}$.

Proof. Since the last input block X^*_{i,l_i} is fresh, hence $\lfloor \Pi(X^*_{i,l_i}) \rfloor_\tau = T^*_i$ with probability at most $2/2^\tau$ (provided $\sigma_e + q_p \leq 2^{b-1}$ which can be assumed, since otherwise our bound is trivially true). □

Let $\mathtt{E}_j$ denote the event that $X^*_{i,j}$ is fresh and $\mathtt{E} := \wedge_{j=p'_i+1}^{m_i} \mathtt{E}_j$

Using the claims, we have

$$\begin{aligned}\Pr_{\Theta_1}[\neg\omega_{d,i} \mid \omega_e, \omega_p] &\leq \Pr_{\Theta_1}[\neg\omega_{d,i} \wedge \mathtt{E} \mid \omega_e, \omega_p] + \Pr[\neg\mathtt{E}].\\ &\leq \frac{2}{2^\tau} + \sum_{j=p'_i+1}^{l_i} \frac{\sigma_d + \sigma_e + q_p}{2^{r_d-r_{45}-1}}.\end{aligned}$$

The last inequality follows from the above claims. Now, we can proceed by using the union bound as follows.

$$\begin{aligned}\Pr[\neg\omega_d \mid \omega_e, \omega_p] &\leq \sum_{i\in\mathcal{D}} \left(\frac{2m_i(\sigma_e + q_p + \sigma_d)}{2^{r_d - r_{45}}} + \frac{2}{2^\tau}\right) \\ &\leq \frac{2\sigma_d(\sigma_e + \sigma_d + q_p)}{2^{r_d - r_{45}}} + \frac{2q_d}{2^\tau} \\ &= \frac{2\sigma_d(\sigma + q_p)}{2^{r_d - r_{45}}} + \frac{2q_d}{2^\tau}. \end{aligned} \tag{5.4}$$

□

Theorem 6 follows from Theorem 2, Lemma 9,11,12 and Proposition 18.

5.5 ORANGE-Zest as a Full-Rate Transform-then-Permute AEAD

5.5.1 ORANGE-Zest

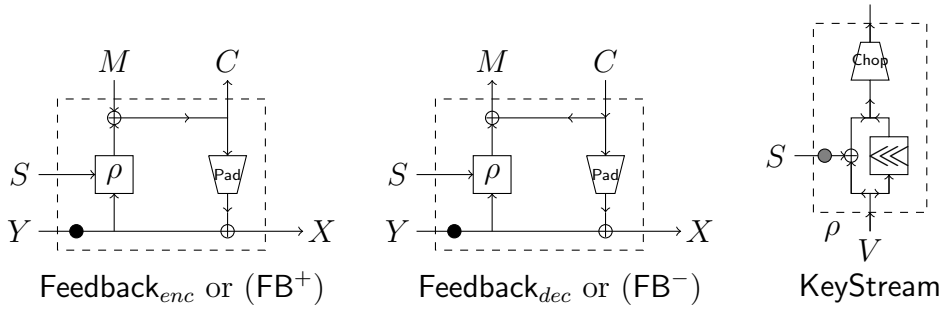


Figure 5-3: Feedback process for ORANGE-Zest: KeyStream module or the function ρ describes how the key-stream is defined. Feedback functions describe to define the next input X for the block cipher and the ciphertext (for encryption feedback) and message (for decryption feedback). The black circular dot represents the mult operation which is nothing but the α^{δ_M}-multiplication to the most significant half of Y (the previous block cipher output). Note that $\delta_M = 0, 1, 2$ for imtermediate block, complete last block, partial last block respectively .The gray circular dot represents the mult operation which is nothing but the α-multiplication to S. Here, Pad and Chop, pads and chops appropriate amounts of bits from MSB or LSB sides. The exact definitions of these process can be found in Algorithm 3

Algorithm 3 The ORANGE-Zest algorithm. Permutation state size $b = 2n$, extra-state size $s = n$, tag size $\tau = n$.

function ORANGE-ZEST$_{[\mathsf{P}]}$.enc(K, N, A, M)
  $(A_{a-1}, \ldots, A_0) \xleftarrow{2n} A$
  $(M_{m-1}, \ldots, M_0) \xleftarrow{2n} M$
  if $a = 0, m = 0$ **then**
    $T \leftarrow \lfloor \mathsf{P}((K \oplus 2) \| N) \rfloor_n$
    return (λ, T)
  if $a = 0, m \neq 0$ **then**
    $(C, U) \leftarrow \mathsf{proc\_txt}(K, (K \oplus 1) \| N, M, +)$
    return $(C, \mathsf{proc\_tg}(U))$
  $C \leftarrow \lambda$
  if $a \neq 0$ **then**
    $(U, S) \leftarrow \mathsf{proc\_hash}(K \| N, A, 1, 2)$
    if $m \neq 0$ **then**
      $(C, U) \leftarrow \mathsf{proc\_txt}(K, U, M, +)$
  return $(C, \mathsf{proc\_tg}(U))$

function ORANGE-ZEST$_{[\mathsf{P}]}$.dec(K, N, A, C, T)
  $(A_{a-1}, \ldots, A_0) \xleftarrow{2n} A$
  $(C_{m-1}, \ldots, C_0) \xleftarrow{2n} C$, $M \leftarrow \lambda$
  if $a = 0, m = 0$ **then**
    $T' \leftarrow \lfloor \mathsf{P}((K \oplus 2) \| N) \rfloor_n$
  if $a = 0, m \neq 0$ **then**
    $(M, U) \leftarrow \mathsf{proc\_txt}(K, (K \oplus 1) \| N, C, -)$
    $T' \leftarrow \mathsf{proc\_tg}(U)$
  if $a \neq 0$ **then**
    $(U, S) \leftarrow \mathsf{proc\_hash}(K \| N, A, 1, 2)$
    if $m \neq 0$ **then**
      $(M, U) \leftarrow \mathsf{proc\_txt}(K, U, C, -)$
    $T' \leftarrow \mathsf{proc\_tg}(U)$
  if $T \neq T'$ **then**
    return $\perp$
  else
    return $(M, \top)$

function proc_hash(X, D, c_0, c_1)
  $(D_{d-1}, \ldots, D_0) \xleftarrow{2n} D$
  $X_0 \leftarrow X$
  for $i = 0$ **to** $d - 2$ **do**
    $Y_i \leftarrow \mathsf{E}(X_i)$
    $X_{i+1} \leftarrow Y_i \oplus D_i$
  $c \leftarrow (2n \mid |D_{d-1}|)?c_0 : c_1$
  $Y_{d-1} \leftarrow \mathsf{E}(X_{d-1})$
  $S \leftarrow \lceil Y_{d-1} \rceil_n$
  $Y_{d-1} \leftarrow \mathsf{mult}(c, Y_{d-1})$
  $X_d \leftarrow Y_{d-1} \oplus \mathsf{pad}(D_{d-1})$
  return (X_d, S)

function proc_txt$(S_0, U_0, D, \mathsf{dir})$
  $(D_{d-1}, \ldots, D_0) \xleftarrow{2n} D$
  for $i = 0$ **to** $d - 1$ **do**
    $V_i \leftarrow \mathsf{E}(U_i)$
    if $i = d - 1$ **then**
      $c \leftarrow (2n \mid |D_{d-1}|)?1 : 2$
      $V_i \leftarrow \mathsf{mult}(c, V_i)$
    $KS_i \leftarrow \mathsf{Feed}(S_i, V_i)$
    $D'_i \leftarrow D_i \oplus \lfloor KS_i \rfloor_{|D_i|}$
    if $\mathsf{dir} =$ " + " **then** $D_i \leftarrow D'_i$
    $S_{i+1} \leftarrow \lceil V_i \rceil_n$
    $U_{i+1} \leftarrow V_i \oplus \mathsf{pad}(D_i)$
  return (D', U_d)

function Feed(S, Y)
  $(Y^b, Y^t) \xleftarrow{n} Y$
  $Z \leftarrow (Y^b \oplus \alpha S) \| (Y^t \lll 1)$
  return Z

function mult(c, V)
  $(V^b, V^t) \xleftarrow{n} V$
  return $\alpha^c \cdot V^b \parallel V^t$

function proc_tg(U)
  return $\lfloor \mathsf{P}(U) \rfloor_n$

5.5.2 Security Analysis of ORANGE-Zest

In the first round of official comments, Dobraunig et. al. [49] showed that the construction is insecure by providing a successful forgery attack. Later Kairallah et. el. [70] showed that the modified version of ORANGE-Zest suffers from the attack due to full-state tag generation. In this section, we first discuss the attacks by Dobraunig et. al. in our notations, and then we show that a modified ORANGE-Zest construction can be viewed as an frTtP construction, and thus with a limited-sized tag generation it is fully secure.

Proposition 19. *[49]* ORANGE-Zest *is not secure.*

We see that the attack [49] arises from the construction of ORANGE-Zest where if no associated data is used then the secret key is taken as the initial extra-state value and hence doesn't depend on the nonce which is a necessary condition as discussed in Proposition 16. Consequently an adversary as defined in the proof of proposition 16 can successfully forge the ORANGE-Zest construction. Next, we show that this weakness is only due to the initial extra-state generation protocol and is not a weakness of the ORANGE-Zest feedback function.

The ORANGE-Zest feedback function can be represented as follows:

$$\mathcal{E}_{\text{ORANGE-ZEST}} = \begin{bmatrix} \begin{bmatrix} I_{b-c} \oplus A^{-1} & 0_{(b-c)\times c} \\ 0_{c\times(b-c)} & 0_{c\times c} \end{bmatrix} & \begin{bmatrix} 0_{(b-c)\times c} \\ I_c \end{bmatrix} & I_b \\ \begin{bmatrix} 0_{c\times(b-c)} & \alpha \cdot I_c \end{bmatrix} & 0_c & 0_{c\times b} \\ I_b & \begin{bmatrix} 0_{(b-c)\times c} \\ I_c \end{bmatrix} & I_b \end{bmatrix}$$

where $A_{b-c} = \begin{bmatrix} 0_{(b-c-1)\times 1} & I_{b-c} \\ 1 & 0_{1\times(b-c)} \end{bmatrix}$.

Let $\rho : \{0,1\}^b \to \{0,1\}^c$ be defined as $\rho(X) = \lfloor X \rfloor_c \ \forall X \in \{0,1\}^b$. Then $r_\rho = c$. Further re-define $\mathsf{Fmt}(N, A, M) = \mathsf{Fmt}(N, \mathsf{pad}(A), M)$ where $\mathsf{pad}(A) = 0^{b-x-1}1 \| A, x = |A| \bmod b$ for all $A \in \{0,1\}^*$. We call the frTtP construction, which uses Fmt as the formatting function, $\rho(\Pi(Key \| Nonce))$ as the initial extra secret state and $\mathcal{E}_{\text{ORANGE-ZEST}}$ as the underlying feedback function, the ORANGE-ZEST$_{mod}$ construction.

We can easily note that $r_3 = b; r_{12} = r_{45} = r'_{45} = r_8 = c$. Further, $r_e = r_d = b + c$.

Corollary 9.

$$\mathbf{Adv}^{\text{AEAD}}_{\text{ORANGE-ZEST}_{mod}}(\sigma, q_p) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \mathcal{O}\left(\frac{\sigma q_p}{2^b}\right) + \mathcal{O}\left(\frac{\sigma^2 + q_p}{2^c}\right).$$

5.6 frTtP: More Instantiations

In this section, we try to discuss some concrete instantiations of the general Full-rate Transform-then-Permute construction. More specifically we use some full-rate feedback functions used in popular block cipher based full-rate AEAD modes such as COFB

HyENA and show that with slight modifications they can also be used securely in the frTtP constructions.

5.6.1 frTtP with CoFB Feedback

The CoFB [36] feedback function can be represented as bellow.

$$\mathcal{E}_{\text{CoFB}} = \begin{bmatrix} G & \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix} & I_b \\ 0_{\frac{b}{2}\times b} & \alpha \cdot I_{\frac{b}{2}} & 0_{\frac{b}{2}\times b} \\ I_b & 0_{b\times\frac{b}{2}} & I_b \end{bmatrix},$$

where G is a square matrix of size b, such that both G and $G \oplus I_b$ are non-singular.

Now consider an frTtP which uses $\mathcal{E}_{\text{CoFB}}$ as the underlying feedback function and an extra state initialisation protocol as discussed in Section 5.2.2 and call it COFBSponge construction.

Proposition 20. COFB*Sponge* *AEAD is insecure.*

Proof. Note that in COFB feedback function $r_8 = 0$ and hence COFBSponge does not satisfy condition (C3) in Proposition 14. □

We modify the COFB feedback function such that $E_8 \neq 0_{\frac{b}{2}}$. In particular we take $E_8 = \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix}$, then $r_8 = b/2$.

$$\mathcal{E}_{\text{CoFB}_{\text{mod}}} = \begin{bmatrix} G & \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix} & I_b \\ 0_{\frac{b}{2}\times b} & \alpha \cdot I_{\frac{b}{2}} & 0_{\frac{b}{2}\times b} \\ I_b & \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix} & I_b \end{bmatrix}$$

Since G and $G + I_b$ are invertible, $r_e = r_d = \frac{3b}{2}$. Note that $r_{12} = b/2, r_3 = b$. Further since, $E_4 = 0_{\frac{b}{2}\times b}$ and $E_5 = \alpha \cdot I_{b/2}$ we have $r_{45} = r'_{45} = \frac{b}{2}$.

Corollary 10.

$$\mathbf{Adv}^{\text{AEAD}}_{\text{COFB}\mathit{Sponge}_{\text{mod}}}(\sigma, q_p) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \mathcal{O}\left(\frac{\sigma q_p}{2^b}\right) + \mathcal{O}\left(\frac{\sigma^2 + q_p}{2^{\frac{b}{2}}}\right).$$

5.6.2 frTtP with HyENA Feedback

The HyENA [34] feedback function can be represented as bellow.

$$\mathcal{E}_{\text{HyENA}} = \begin{bmatrix} \begin{bmatrix} I_{\frac{b}{2}} & 0_{\frac{b}{2}} \\ 0_{\frac{b}{2}} & 0_{\frac{b}{2}} \end{bmatrix} & \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix} & I_b \\ 0_{\frac{b}{2}\times b} & \alpha \cdot I_{\frac{b}{2}} & 0_{\frac{b}{2}\times b} \\ I_b & 0_{b\times\frac{b}{2}} & I_b \end{bmatrix},$$

Now consider an frTtP which uses $\mathcal{E}_{\text{HyENA}}$ as the underlying feedback function and an extra state initialisation protocol as discussed in Section 5.2.2 and call it HyENASponge construction.

Proposition 21. HyENA*Sponge* *AEAD is insecure.*

Proof. Note that in HyENA feedback function $r_8 = 0$ and hence HyENASponge does not satisfy condition (C3) in Proposition 14. □

We modify the HyENA feedback function so that $E_8 \neq 0_{\frac{b}{2}}$. In particular we take $E_8 = \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix}$. Then $r_8 = \frac{b}{2}$. We further modify $\mathcal{E}_1 = \begin{bmatrix} \alpha \cdot I_{\frac{b}{2}} & 0_{\frac{b}{2}} \\ 0_{\frac{b}{2}} & 0_{\frac{b}{2}} \end{bmatrix}$ so that $r_d = r_e = \frac{3b}{2}$.

$$\mathcal{E}_{\text{HyENA}_{\text{mod}}} = \begin{bmatrix} \begin{bmatrix} \alpha \cdot I_{\frac{b}{2}} & 0_{\frac{b}{2}} \\ 0_{\frac{b}{2}} & 0_{\frac{b}{2}} \end{bmatrix} & \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix} & I_b \\ 0_{\frac{b}{2}} & \alpha \cdot I_{\frac{b}{2}} & 0_{\frac{b}{2}} \\ I_b & \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix} & I_b \end{bmatrix}$$

Note that $r_{12} = b/2, r_3 = b$. Further since, $E_4 = 0_{b/2 \times b}$ and $E_5 = \alpha \cdot I_{b/2}$ we have $r_{45} = r'_{45} = \frac{b}{2}$.

Corollary 11.

$$\mathbf{Adv}^{\text{AEAD}}_{\text{HyENA}\mathit{Sponge}_{\text{mod}}}(\sigma, q_p) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \mathcal{O}\left(\frac{\sigma q_p}{2^b}\right) + \mathcal{O}\left(\frac{\sigma^2 + q_p}{2^{\frac{b}{2}}}\right).$$

5.7 Conclusion

In this chapter, we introduced a general full-rate Sponge type construction called the frTtP, which uses an extra-state as compensation for increasing the size in the rate part. We further showed that an frTtP construction could achieve security up to $D \ll 2^{c/2}, T \ll 2^c$ when this extra-state is initialized properly, and the underlying feedback function satisfies some suitable conditions. As a consequence, we showed that the weakness in the full-rate construction ORANGE-Zest [39] was due to an improper initialization protocol, and with a proper initialization function, one can get a secure full-rate Sponge-type AEAD scheme using the ORANGE-Zest feedback function. We also considered some full-rate feedback functions used in popular constructions like CoFB and HyENA and showed that with some proper modifications, these feedback functions could also be used to construct secure full-rate Sponge-type modes.

As noted, an increase in the rate of message absorption in encryption/decryption protocol leads to the higher efficiency of an AEAD construction in terms of runtime. Hence attaining the maximum possible rate of message processing per primitive call is a desired property in any AEAD. In this chapter, we tried to construct full-rate lightweight AEAD constructions using permutation as the underlying primitives. Since block cipher is another popular choice as primitive in AEADs, constructing a lightweight block-cipher-based AEAD, which attains the maximum possible message absorption rate per block cipher call, is an important cryptographic problem. In the next chapter, we will try to address this.

Chapter 6

Designing a TBC-based Full-rate AEAD

6.1 Introduction

In this chapter, our primary goal is to design a lightweight block (tweakable) cipher-based AEAD scheme, that should be efficient, provide high performance, and be able to perform well in low-end devices. In addition, we also demand robustness in security. While designing such a scheme we keep in mind the following.

Maximal Rate

The design must guarantee the maximum possible rate of message absorption per block (tweakable) cipher calls so as to maximize efficiency in terms of run-time.

Minimal State The design must guarantee appropriate implementation characteristics on both lightweight and high-performance systems and must have a state size that is equal to the block size of the underlying cipher.

Inverse-Free An authenticated algorithm with no inverse should be used in the design. No decryption call to the underlying block cipher is necessary for the algorithm's validated encryption or decryption. This greatly reduces the overall hardware footprint, particularly for solutions that incorporate authorized encryption and verified decryption.

Minimally Xored Mixture Feedback We use a minimum number of xors to process each block. This makes the design simpler and has a very low footprint in software. The rationale behind having a mixture of plaintext and ciphertext feedback is to achieve NIST-aimed security. During encryption, we ensure 192-bit entropy for each block process. We have a 128-bit dynamic secret key and 64-bits LSB of the inputs that influence the 64-bits LSB of the previous block cipher call.

While decrypting, we have 64-bit MSB of the previous outputs going to the corresponding position of the next input. This would provide about 64-bit security for forgery attempts.

In Section 6.2, we define some newly introduced security definitions associated with AEAD modes and tweakable block ciphers. In Section 6.3, we introduce the TBC-based AEAD scheme called mF. In Section 6.4, we reduce the security of the mF mode to the security of the underlying TBC against the newly introduced TBC security games defined in Section 6.2. In Section 6.5, we define a new TBC construction using a block cipher and a key updation function (KUF) and upper bound the advantages of any adversary playing those new TBC security games against it. In Section 6.6, we consider the mF mode under this new TBC and derive an upper bound on the security of such a mode. Finally, in Section 6.7, we make a theoretical comparison between an instantiation of mF mode called the $\mathsf{mF}_{\mathsf{prim}}$ mode of AEAD and some other existing TBC-based lightweight AEAD schemes.

6.2 Security Definitions for μ-respecting Adversaries

6.2.1 μ-respecting TPRP-security

Let μ be a positive integer. We define μ-TPRP advantage of $\tilde{E}$ to be $\mathbf{Adv}_{\tilde{E}}^{\mu\text{-tprp}}(q,t) = \max_{\mathscr{A}} \mathbf{Adv}_{\tilde{E}}^{\text{tprp}}(\mathscr{A})$ where the maximum is taken over all μ-respecting adversaries $\mathscr{A}$ (i.e. the number of queries (tw, X) by $\mathscr{A}$ with a same plaintext input X is at most μ) running in time t. When the tweakable block cipher is instantiated in the ideal cipher model, the time parameter t denotes the number of ideal cipher calls.

6.2.2 Multi-Commitment Prediction

Let n be a positive integer. Let $\mathscr{A}$ be an adversary which has oracle access to an n-bit tweakable block cipher $\tilde{E}$ with a tweak space $\mathcal{T}$ in the first phase.

PHASE 2:

1. After all the queries of the first phase are done, it makes at most λ commitments of the form (tw_i, x_i, y_i) where $x_i, y_i \in \{0,1\}^{\frac{n}{2}}$, $tw_i \in \mathcal{T}$.

2. $\mathscr{A}$ makes prediction queries of the form (tw'_j, X_j) such that (tw'_j, X_j) are fresh i.e., $\forall j, (tw'_j, X_j)$ has never been queried before.

We say that any adversary $\mathscr{A}$ wins the λ-*multi-commitment-prediction* game if for some prediction query tuple (tw'_j, X_j) there exist a commitment tuple (tw_i, x_i, y_i) such that

$$tw_i = tw'_j;\ x_i = \lceil X_j \rceil_{\frac{n}{2}};\ \lfloor \tilde{E}_K(tw_i, X_j) \rfloor_{\frac{n}{2}} = y_i.$$

The λ-*multi-commitment-predicting* advantage of an adversary $\mathscr{A}$ is defined as

$$\mathbf{Adv}^{\lambda\text{-mcp}}_{\tilde{E}}(\mathscr{A}) = \Pr\left[\mathscr{A}^{\tilde{E}} \text{wins the } \lambda\text{-}\textit{multi-commitment-prediction} \text{ game}\right]$$

and we write,

$$\mathbf{Adv}^{\lambda\text{-mcp}}_{\tilde{E}}(q,t) = \max_{\mathscr{A}} \mathbf{Adv}^{\lambda\text{-mcp}}_{\tilde{E}}(\mathscr{A})$$

where maximum is taken over all adversaries $\mathscr{A}$ running in time t making at most q queries.

We define (μ,λ)-mcp advantage of $\mathscr{A}$ to be

$$\mathbf{Adv}^{(\mu,\lambda)\text{-mcp}}_{\tilde{E}}(q,t) = \max_{\mathscr{A}} \mathbf{Adv}^{\lambda\text{-mcp}}_{\tilde{E}}(\mathscr{A})$$

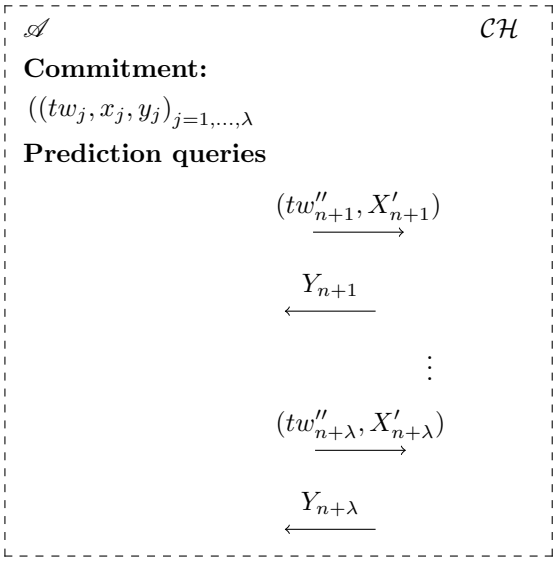
where the maximum is taken over all adversaries as defined above with the additional restriction that they make μ-respecting queries in the first phase of the game.

We would like to note that in the ideal cipher model the (μ,λ)-multi-commitment prediction security is defined in the same way as above with an additional restriction that the adversary doesn't make any primitive calls to E in the phase-2.

6.2.3 Multicollision Security Game

We say that an adversary $\mathscr{A}$ with oracle access to $\mathcal{O}$ produces a μ-multicollision if it makes μ distinct queries $x_1,\ldots,x_\mu$ to $\mathcal{O}$, for which all responses are identical. The

Figure 6-1: PHASE 2 of the (λ, μ)-mcp game between $\mathscr{A}$ and $\mathcal{CH}$. The Phase 1 $\mathcal{CH}$ queries are responded similarly to as in the case of the μ-TPRP game. For Phase 2 queries, the μ-restriction is lifted. Note that the Phase 2 queries and predictions can be done in any order. The only condition is that a prediction (tw'_j, X_j) must be fresh i.e., it has not been queried before.



μ-*multicollision*-advantage of the adversary $\mathscr{A}$ is defined as

$$\mathbf{Adv}_{\mathcal{O}}^{\mu\text{-mcoll}}(\mathscr{A}) = \Pr\left[\mathscr{A}^{\mathcal{O}} \text{ produces } \mu\text{-multicollision}\right]$$

and $\mathbf{Adv}_{\mathcal{O}}^{\mu\text{-mcoll}}(q) = \max_{\mathscr{A}} \mathbf{Adv}_{\mathcal{O}}^{\mu\text{-mcoll}}(\mathscr{A})$, where maximum is taken over all adversaries $\mathscr{A}$ making at most q queries.

6.3 The mF Mode of AEAD

We start by defining the positive feedback function FB^+ which takes a chain input Y of size n-bits and a data input M of size less than or equal to n-bits to generate a data output C of size $|M|$-bits and a chain output of size n. The negative feedback function can be described in a similar way. The general description of the feedback functions can be understood from Figures 6-2 and 6-3. For the case when the data input M has length n, the feedback functions can be described in a much simpler way. If $|M| = n$, then $FB^+(Y, M) = (X, C)$, where $C = Y \oplus M$ and $X = \lceil C \rceil_{n/2} \| \lfloor M \rfloor_{n/2}$. Similarly, if $|C| = n$, then $FB^-(Y, C) = (X, M)$, where $M = Y \oplus C$ and $X = \lceil C \rceil_{n/2} \| \lfloor M \rfloor_{n/2}$.

Let $\tilde{E}$ be a tweakable block cipher with state size n and tweak-size $t > n - 8$. We define the mF mode of AEAD using this TBC as follows. Given any data D, we define $d := \left\lceil \frac{|D|}{n} \right\rceil$. We parse the data D into d parts of n bit data blocks. In notation $(D_d, \ldots, D_1) \xleftarrow{n} D$, where $|D_d| = \begin{cases} n \text{ if } n \mid |D| \\ r \text{ if } \exists r > 0 \text{ s.t } |D| \equiv r \bmod n. \end{cases}$

Given any data $(N, A, M) \in \{0,1\}^{n-8} \times \{0,1\}^* \times \{0,1\}^*$ and distinct but predefined $\{a_1, \ldots, a_6\}$, we define (a, δ_A) and (m, δ_M) depending on $|A|$ and $|M|$ using the Fmt function as described in Algorithm 4. We restrict the values of a, m such that $a+m+2 \leq 2^{t-n+8}$. With this setup, the mF mode encryption with the secret key K, simply outputs $\tilde{E}_K(0^t, N \| 0^6 10)$ as the tag if both $a, m = 0$. Else it sets $N' = N \| 0^7 1$ if $a = 0$ and $N' = N \| 0^8$ if $a \neq 0$. It takes $((N', 0), N \| 0^8)$ as the first tweakable block cipher input to generate Y_0.

Associated data Processing: It parses $(A_a, \ldots, A_1) \xleftarrow{n} A$. For each $i \in [0, a-1]$, it

evaluates $FB^+(Y_i, A_{i+1})$ to generate $(X_{i+1}, \star)$ and use $((N', i), X_{i+1})$ as the next TBC input to generate Y_{i+i}. Finally, it makes another TBC call with input $((N', a+1), Y_a \oplus \delta_A)$ to generate Y_{a+1}. It outputs Y_{a+1} as tag, if $m = 0$.

Message Processing: It parses $(M_m, \ldots, M_1) \xleftarrow{n} M$. For each $i \in [a+1, l]$ it evaluates $FB^+(Y_i, M_{i-a})$ to generate (X_{i+1}, C_{i-a}) and use $((N', i), X_{i+1})$ as the next TBC input to generate Y_{i+i}. Finally it makes another TBC call with input $((N', l+2), Y_{l+1} \oplus \delta_M)$ to generate the tag. It outputs $(C_m, \ldots, C_1)$ as the ciphertext.

Let pad be 0^*1 padding function. $\perp$ represents invalid. $P_1 \; ? \; a_1 \; : \; a_2$ evaluates to a_1 if P_1 is true and a_2 otherwise. $P_1 \; \& \; P_2 \; ? \; a_1 \; : \; a_2 \; : \; a_3 \; : \; a_4$ evaluates to a_1 if both P_1 and P_2 are true, to a_2 if only P_1 is true, to a_3 if only P_2 is true and to a_4 if none of P_1 , P_2 are true. $\mathsf{Feed}(\star, \star, \text{dir}) := \begin{cases} FB^+ \text{ if dir} = + \\ FB^- \text{ if dir} = -. \end{cases}$. Then, with these notations, mF mode of AEAD can be best understood from Figure 6-4 and Algorithm 4.

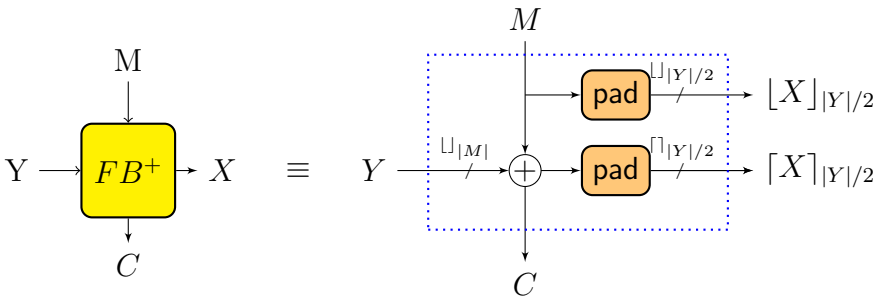


Figure 6-2: The FB^+ Function in mF. pad is 0^*1 padding.

Algorithm 4 mF Mode

function $\text{MF}_{[\mathsf{E}]}.\mathsf{enc}(K, N, A, M)$
  $((a, \delta_A), (m, \delta_M)) \leftarrow \mathsf{Fmt}(A, M)$
  if $a = 0, m = 0$ **then**
    $T \leftarrow \tilde{\mathsf{E}}_K(0^t, N \| 0^6 10)$
    return (λ, T)
  else if $a = 0$ **then** $N' \leftarrow N \| 0^7 1$
  else $N' \leftarrow N \| 0^8$
  $T \leftarrow \tilde{\mathsf{E}}_K((N', 0), N \| 0^8)$
  $C \leftarrow \lambda$
  if $a \neq 0$ **then**
    $(*, T) \leftarrow \mathsf{proc\_txt}(N', 0, K, T, A, \delta_A, +)$
  if $m \neq 0$ **then**
    $(C, T) \leftarrow \mathsf{proc\_txt}(N', a + 1, K, T, M, \delta_M, +)$
  return (C, T)

function $\text{MF}_{[\mathsf{E}]}.\mathsf{dec}(K, N, A, C, T)$
  $((a, \delta_A), (m, \delta_C)) \leftarrow \mathsf{Fmt}(A, C)$
  if $a = 0, m = 0$ **then**
    $(T', *) \leftarrow \mathsf{E}_K(N \| 0^6 10)$
    return $(T = T')? \top : \bot$
  else if $a = 0$ **then** $N' \leftarrow N \| 0^7 1$
  else $N' \leftarrow N \| 0^8$
  $T \leftarrow \tilde{\mathsf{E}}_K((N', 0), N \| 0^8)$
  $C \leftarrow \lambda$
  if $a \neq 0$ **then**
    $(*, T') \leftarrow \mathsf{proc\_txt}(N', 0, K, T', A, \delta_A, +)$
  if $m \neq 0$ **then**
    $(M, T') \leftarrow \mathsf{proc\_txt}(N', a + 1, K, T', C, \delta_C, -)$
  if $T \neq T'$ **then**
    return $\bot$
  else
    return $(M, \top)$

function $\mathsf{Fmt}(A, M)$
  $(A_{a-1}, \ldots, A_0) \xleftarrow{n} A$
  $(M_{m-1}, \ldots, M_0) \xleftarrow{n} M$
  $\delta_A \leftarrow (n \mid |A_{a-1}|)\ \&\ (m = 0)?\ a_1 : a_2 : a_3 : a_4$
  $\delta_M \leftarrow (n \mid |M_{m-1}|)?\ a_6 : a_5$
  return $((a, \delta_A), (m, \delta_M))$

function $\mathsf{proc\_txt}(N, l, K, Y_0, D, \delta_D, dir)$
  $(D_d, \ldots, D_1) \xleftarrow{n} D$
  for $i = 1$ **to** d **do**
    $(X_i, D'_i) \leftarrow \mathsf{Feed}(Y_i - 1, D_i, dir)$
    $Y_i \leftarrow \tilde{\mathsf{E}}_K((N, l + i), X_i)$
  $X_{d+1} \leftarrow Y_d \oplus 0^{n-4} \| \delta_D$
  $Y_{d+1} \leftarrow \tilde{\mathsf{E}}_K((N, l + d + 1), X_{d+1})$
  return (D', Y_{d+1})

function $\mathsf{Feed}(Y, D, \text{dir})$
  $D' \leftarrow D \oplus \lfloor Y \rfloor_{|D|}$
  if dir $=$ " $+$ " **then**
    $B \leftarrow \lceil \mathsf{pad}(D') \rceil_{n/2} \| \lfloor \mathsf{pad}(D) \rfloor_{n/2}$
  if dir $=$ " $-$ " **then**
    $B \leftarrow \lceil \mathsf{pad}(D) \rceil_{n/2} \| \lfloor \mathsf{pad}(D') \rfloor_{n/2}$
  $X \leftarrow B \oplus Y$
  return (X, D')

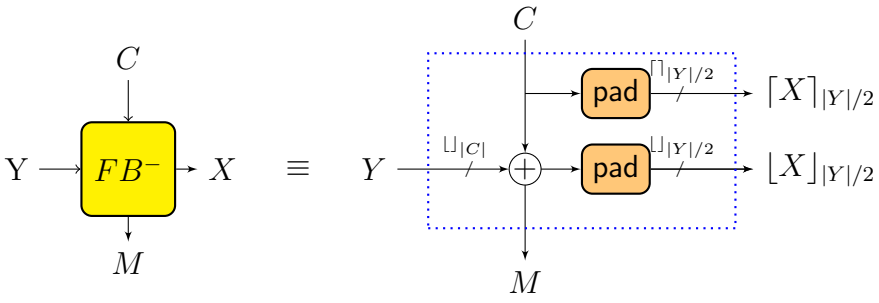


Figure 6-3: The FB^{-} Function in mF. pad is 0^*1 padding.

.

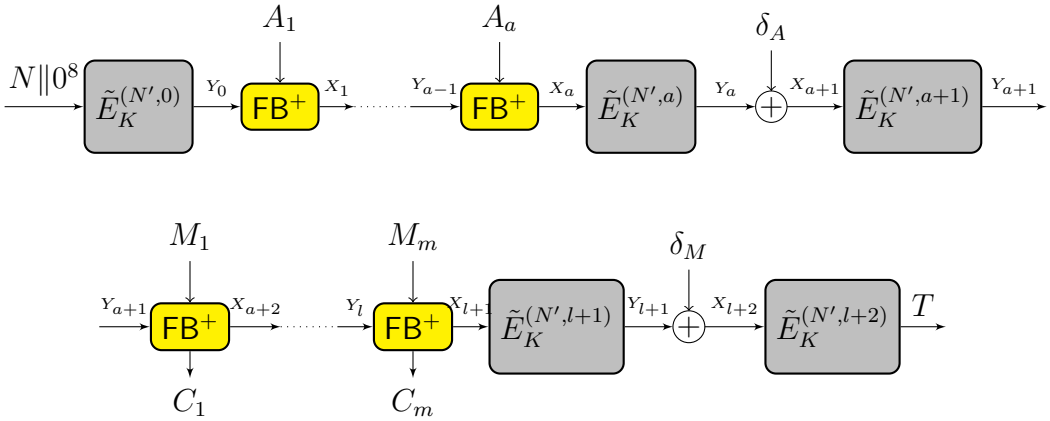


Figure 6-4: Block diagram for mF encryption. Here, $N' = N\|x$, where $x = 0^8/0^71$ depending on the condition that $(a \neq 0)$ or $(a = 0\ \&\ m \neq 0)$ respectively. We define $l = a + m$. δ_A,δ_M are as defined in Figure 4

.

6.4 Security Reductions of mF

Here, we give upper bounds on the *privacy advantage* and *forging advantage* of mF against any adversary $\mathscr{B}$. For notational reference, see Figure 6-4.

6.4.1 Privacy

Theorem 7. *For any privacy adversary $\mathscr{B}$ of mF, there is an μ-TPRP adversary $\mathscr{A}$ of $\tilde{E}$, such that*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{mF}}(\mathscr{B}) \leq \mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}(\mathscr{A}) + \sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^{\mu},$$

where σ is the total number of queries by $\mathscr{A}$.

Proof. Note that mF is the mode based on a tweakable block cipher $\tilde{E}$. If we replace $\tilde{E}$ by an n-bit tweakable random permutation P with same tweak space, we denote the construction as mF_P. From the construction it is easy to see that all tweaks used in the tweakable random permutation while we execute nonce-respecting queries are distinct. Hence, all output bits of mF_P are random, so it is equivalent to the oracle \$. So, $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{mF}}(\mathscr{B}) = \Pr\left[\mathscr{B}^{\mathsf{mF}_P} = 1\right] - \Pr\left[\mathscr{B}^{\mathsf{mF}_{\tilde{E}}} = 1\right]$.

By using straightforward reduction, one can construct an adversary $\mathscr{A}'$ which mainly simulates the mode $\mathsf{mF}_{\mathcal{O}}$, where $\mathcal{O}$ (which is either P or $\tilde{\mathsf{E}}_K$) is the challenge oracle of $\mathscr{A}'$. Clearly, $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{mF}}(\mathscr{B}) \leq \mathbf{Adv}^{\mathrm{tprp}}_{\tilde{\mathsf{E}}}(\mathscr{A}')$. However, $\mathscr{A}'$ does not necessarily follow μ-input-respecting for small μ. So we follow a slightly different strategy to define $\mathscr{A}$. It is basically same as $\mathscr{A}'$, except that it aborts and returns 0 whenever it is going to

violate μ-input-restriction. More precisely, it maintains a list of all queries (tw_i, X_i) to its challenger. If for some i, there exists $\mu+1$ number of $j < i$ with $X_j = X_i$, it aborts (instead of making the query) and returns zero. When it does not abort, it returns whatever $\mathscr{B}$ returns. Now,

$$\begin{aligned}
\Pr\left[\mathscr{B}^{\mathsf{mF}_P} = 1\right] - \Pr\left[\mathscr{B}^{\mathsf{mF}_{\tilde{E}}} = 1\right] \leq{}& \Pr\left[\mathscr{B}^{\mathsf{mF}_P} = 1 \cap \mathscr{A} \text{ doesn't Abort}\right] \\
&+ \Pr\left[\mathscr{A}^P \text{Aborts}\right] \\
&- \Pr\left[\mathscr{B}^{\mathsf{mF}_{\tilde{E}}} = 1 \cap \mathscr{A} \text{ doesn't Abort}\right] \\
\leq_{(1)}{}& \left(\Pr\left[\mathscr{A}^P = 1\right] - \Pr\left[\mathscr{A}^{\mathsf{mF}_{\tilde{E}}} = 1\right]\right) \\
&+ \Pr\left[\mathscr{A}^P \text{Aborts}\right] \\
\leq_{(2)}{}& \mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}(\mathscr{A}) + \sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^{\mu}.
\end{aligned}$$

The inequality (1) follows from the definition that $\mathscr{A}$ returns 1 if and only if it does not abort and $\mathscr{B}$ returns 1. We now bound $\Pr\left[\mathscr{A}^P \text{ Aborts}\right] \leq \sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^{\mu}$ which justifies inequality (2) above.

Consider the event that $\mathscr{A}^P$ Aborts i.e., $\exists\{i_1, \ldots, i_{\mu+1}\} \in [1, \sigma]$ such that $\mathscr{A}$ needs to make $\mu+1$ queries of the form (tw_{i_j}, X_{i_j}) to P such that $X_{i_j} = X_{i_{j'}}$ $\forall j, j' \in [1, \mu+1]$. Now, note that for all such queries, if the previous query by $\mathscr{A}$ is of the form (tw_{i_j-1}, X_{i_j-1}) and it received the output Y_{i_j-1}, then

$$\lfloor X_{i_j} \rfloor_{\frac{n}{2}} = \lfloor Y_{i_j-1} \oplus C_{i_j} \rfloor_{\frac{n}{2}}$$

for some known $C_{i_1}, \ldots, C_{i_{\mu+1}}$. Now consider an oracle $\mathcal{O}_P$ which takes input of the form (tw_i, X_i, C_i) and outputs $z_i = \lfloor Y_i \oplus C_i \rfloor_{\frac{n}{2}}$ as response, where $Y_i = P(tw_i, X_i)$.

Then, clearly,

$$\Pr\left[\mathscr{A}^P \text{ Aborts}\right] \leq \mathbf{Adv}^{(\mu+1)\text{-mcoll}}_{\mathcal{O}_P}(\sigma)$$

.

Finally, to bound $\mathbf{Adv}^{(\mu)\text{-mcoll}}_{\mathcal{O}_P}(\sigma)$, let $\omega_d = ((N_i, j_i), X_i, z_i)_{i\in\mathcal{E}}$ be the online transcript of any adversary playing μ-mcoll game with $\mathcal{O}_P$.

The μ-multi collision occurs if $\exists i_1, \ldots, i_\mu \in [1, \sigma]$ such that $z_{i_k} = z_{i_l}$ for all $k, l \in [1, \mu]$.

Note that the probability of μ-multi collision is highest when the tweak is the same for all the queries.

In that case, for a given $x \in \{0,1\}^{\frac{n}{2}}$ and fixed $i_k \in [1, \sigma]$ number of possible tuples of (Y_{i_k}, C_{i_K}) such that $z_{i_k} = \lfloor Y_{i_k} \oplus C_{i_k} \rfloor_{\frac{n}{2}} = x$ is bounded by $2^{\frac{n}{2}}$. Hence, varying over all $i_1, \ldots, i_\mu \in [1, \sigma]$, we have number of possible tuples $(Y_{i_1}, C_{i_1}), \ldots, (Y_{i_\mu}, C_{i_\mu})$ such that $z_{i_k} = \lfloor Y_{i_k} \oplus C_{i_k} \rfloor_{\frac{n}{2}} = x \ \forall k \in [1, \mu]$ is bounded by $2^{\frac{\mu n}{2}}$.

Varying over all $x \in \{0,1\}^{\frac{n}{2}}$ and for all combination of $i_1, \ldots, i_\mu \in [1, \sigma]$, we have number of ways in which μ-multi collision occurs is at most $\binom{\sigma}{\mu} 2^{\frac{(\mu+1)n}{2}}$.

Hence, we have

Lemma 13.

$$\begin{aligned}\mathbf{Adv}^{\mu\text{-mcoll}}_{\mathcal{O}_p}(\sigma) = \Pr\left[\mu\text{-mcoll}\right] &\leq \frac{\binom{\sigma}{\mu} 2^{\frac{(\mu+1)n}{2}}}{(2^n)_\mu} \\ &\leq \sigma\left(1 + \frac{\mu^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^{\mu-1}.\end{aligned}$$

□

6.4.2 Forgery

Define an oracle $\mathcal{O}_{\tilde{E}_K}$, which takes a query input of the form (tw, X, C) and returns

$$X' = C \oplus \left(0^{\frac{n}{2}} \| \lfloor \tilde{E}_K(tw, X) \rfloor_{\frac{n}{2}}\right).$$

We can similarly define $\mathcal{O}_P$, where the the tweakable block cipher above is replaced by tweakable random permutation P. For any $(\mu+1)$-multicollision $(\mu+1)$-input-restricting adversary $\mathscr{C}$ with oracle access to $\mathcal{O}_{\tilde{E}_K}$, there is an $(\mu+1)$-*multicollision* adversary $\mathscr{C}'$ with oracle access to $\mathcal{O}_P$, such that

$$\mathbf{Adv}_{\tilde{E}}^{(\mu+1)\text{-mcoll}}(\mathscr{C}) \leq \mathbf{Adv}_{\tilde{E}}^{\mu\text{-tprp}}(\mathscr{C}') + \sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^{\mu}.$$

This follows from the standard reduction and Lemma 13 .

Theorem 8. *For any nonce-respecting forging adversary $\mathscr{B}$ of mF making q_e encryption queries with σ_e encryption query blocks, q_d decryption queries with σ_d decryption query blocks, there is (i) (μ, σ_d)-mcp adversary $\mathscr{A}$ of $\tilde{E}$, and (ii) $(\mu+1)$-multicollision adversary $\mathscr{C}$ with oracle access to $\mathcal{O}_{\tilde{E}_K}$ (as defined above), such that*

$$\begin{aligned}\mathbf{Adv}_{mF}^{\text{forge}}(\mathscr{B}) \leq\ & \mathbf{Adv}_{\tilde{E}}^{(\mu,\sigma_d)\text{-mcp}}(\mathscr{A}) + 2 \cdot \mathbf{Adv}_{\tilde{E}}^{\mu\text{-tprp}}(\mathscr{C}') \\ & + \sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^{\mu} + \frac{2\sigma_e}{2^{\frac{n}{2}}}.\end{aligned}$$

Let $\mathscr{B}$ be any forging adversary of mF. Suppose $\mathscr{B}$ makes q_e encryption queries with σ_e encryption query blocks and q_d forging attempts with effectively σ_d encryption blocks. We construct a (μ, σ_d)-mcp adversary $\mathscr{A}$ which uses $\mathscr{B}$ to win the (μ, σ_d)-multi-

commitment-prediction game of $\tilde{E}$.

The Reduction Game

Let $\mathcal{CH}$ be a (μ, λ)-mcp challenger. $\mathscr{A}$ acts as a forgery challenger for $\mathscr{B}$, as follows:
PHASE 1 :

I. Whenever $\mathscr{B}$ sends an encryption query of the form $(N^i, A^i, M^i)_{i\in[1,q_e]}$,

   A. $\mathscr{A}$ responds to the query by computing (C^i, T^i) by making the required $\tilde{E}_K$ queries to $\mathcal{CH}$.

   B. In the previous step, $\mathscr{A}$ always follows the restriction that no more than μ queries to $\tilde{E}$ have the same input. Otherwise, it aborts.

II. For every decryption query of the form $(N^{*j}, A^{*j}, C^{*j}, T^{*j})_{j\in[1,q_d]}$, $\mathscr{A}$ simply responds it with $\perp$.

III. When all the encryption and decryption queries by $\mathscr{B}$ have been responded, $\mathscr{A}$ revisits all the decryption queries made by $\mathscr{B}$. For each $j \in [1, q_d]$, $\mathscr{A}$ proceeds as follows:

   A. $\mathscr{A}$ checks if $\mathscr{B}$ has previously made any encryption query (N^i, A^i, M^i) and received output of the form (C^i, T^i) such that $N^i = N^{*j}$ and defines an in-

teger p_j as follows:

(a) if there doesn't exist any encryption query (N^i, A^i, M^i) from $\mathscr{B}$ such that $N^i = N^{*j}$, then $\mathscr{A}$ sets $p_j = -1$.

(b) Else if $\exists (N^i, A^i, M^i)$ such that $N^i = N^{*j}$ but $T^i \neq T^{*j}$ or $l_i < l_j^*$, then $\mathscr{A}$ sets $p_j = -1$.

(c) Else if $\exists (N^i, A^i, M^i)$ such that $N^i = N^{*j}$ but $l_i > l_j^*$ and $T^{*j} \neq Y^i_{l_j^*+2}$, then $\mathscr{A}$ sets $p_j = -1$.

(d) Else if $\exists (N^i, A^i, M^i)$ such that $N^i = N^{*j}$ but $l_i > l_j^*$ and $T^{*j} = Y^i_{l_j^*+2}$, then $\mathscr{A}$ sets $p_j = 0$.

(e) Else if $p'_j \in \mathbb{Z}_{\geq 0}$ be such that $\mathsf{pad}(C^{*j}_{m_j^*-k}) = \mathsf{pad}(C^i_{m_i-k}), \forall k \in [0, p'_j)$ but $\mathsf{pad}(C^{*j}_{m_j^*-p'_j}) \neq \mathsf{pad}(C^i_{m_i-p'_j})$, then

(i) $\mathscr{A}$ defines $p_j = \begin{cases} p'_j + 1 & \text{if } if \lceil \mathsf{pad}(C^{*j}_{m_j^*-p'_j}) \rceil_{\frac{n}{2}} = \lceil \mathsf{pad}(C^i_{m_i-p'_j}) \rceil_{\frac{n}{2}} \\ p'_j + 2 & \text{otherwise .} \end{cases}$

(ii) $\mathscr{A}$ defines

$$\Delta^j := \lfloor \mathsf{pad}(C^{*j}_{m_j^*-p_j+1}) \rfloor_{\frac{n}{2}} \oplus \lfloor \mathsf{pad}(C^i_{m_i-p_j+1}) \rfloor_{\frac{n}{2}}.$$

B. If $p_j = -1, 0$; $\mathscr{A}$ computes Y_k^{*j} for all $k \in [0, l_j^* - p_j]$ and else computes Y_k^{*j} for all $k \in [0, l_j^* - p_j + 1]$ with the help of $\mathcal{CH}$ following the restriction

that no more than μ queries to $\tilde{E}$ have the same input. In that case $\mathscr{A}$ aborts.

Remark 6. *If there exists a common prefix between* (N^i, A^i, C^i) *and* (N^{*j}, A^{*j}, C^{*j}), *then* $\mathscr{A}$ *already has computed up to the common prefix length during encryption query and thus need not send any new encryption query to* $\mathcal{CH}$ *for computation up to that point.*

Phase 2 (Commitment):

For each $j \in [1, q_d]$,

I. If $p_j = -1$, then,

A. Note that $\mathscr{A}$ knows $Y^{*j}_{l^*_j+1}$ from Phase 1.

B. $\mathscr{A}$ sets commitment of the form $((N^{*j}, l^*_j + 2), \lceil Y^{*j}_{l^*_j+1} \rceil_{\frac{n}{2}}, \lfloor T^{*j} \rfloor_{\frac{n}{2}})$.

II. If $p_j = 0$, then,

A. $\mathscr{A}$ sets $\lfloor Y^{*j}_{l^*_j+1} \rfloor_{\frac{n}{2}} = \lfloor Y^{i}_{l^*_j+1} \oplus D^{i}_{l^*_j+1} \oplus \delta_{M^{*j}} \rfloor_{\frac{n}{2}}$, where

$$D^{i}_{l^*_j+1} = \begin{cases} Y^{i}_{l^*_j+1} \oplus A^{i}_{l^*_j+2} & \text{if } l^*_j < a_i \\ \delta_{A^i} \quad \text{if } l^*_j = a_i \\ C^{i}_{i^*_j - a + 1} \quad \text{if } l^*_j > a_i. \end{cases}$$

B. $\mathscr{A}$ sets commitment of the form $((N^{*j}, l_j^* + 1), \lceil C_{m_j^*}^{*j} \rceil_{\frac{n}{2}}, \lfloor Y_{l_j^*+1}^{*j} \rfloor_{\frac{n}{2}})$.

III. If $p_j \neq 0, -1$, then $\mathscr{A}$ makes p_j commitments of the form

$$(tw_k^{*j}, x_k^{*j}, y_k^{*j})_{k \in [l_j^* - p_j + 2, l_j^* + 1]},$$

where

$$tw_k^{*j} = (N^{*j}, k); x_k^{*j} = \lceil C_{m_j^* - p_j + 1}^{*j} \rceil_{\frac{n}{2}};$$

$$y_k^{*j} = \begin{cases} \lfloor Y_k^i \rfloor_{\frac{n}{2}} \oplus \Delta^j \text{ if } k = l_j^* - p_j + 2 \\ \lfloor Y_k^i \rfloor_{\frac{n}{2}} \text{ Otherwise.} \end{cases}$$

Phase 2 (Prediction):

For each $j \in [1, q_d]$

I. If $p_j = -1$, then

A. It calculates $X_{l_j^*+2}^{*j} = Y_{l_j^*+1}^{*j} \oplus \delta_{M^{*j}}$.

B. It sends prediction query of the form $((N^{*j}, l_j^* + 2), X_{l_j^*+2}^{*j})$.

II. If $p_j = 0$, then

A. Note that $\mathscr{A}$ knows $Y_{l_j^*}^{*j}$ from Phase 1.

B. $\mathscr{A}$ then sets $X_{l_j^*+1}^{*j} = (0^{\frac{n}{2}} \| \lfloor Y_{l_j^*}^{*j} \rfloor_{\frac{n}{2}}) \oplus C_{m_j^*}^{*j}$.

C. finally $\mathscr{A}$ send $((N^{*j}, l_j^* + 1), X_{l_j^*+1}^{*j})$ as a prediction query.

III. If $p_j \neq 0, -1$, then

A. Note that, $\mathscr{A}$ knows $Y^{*j}_{l^*_j - p_j + 1}$ from PHASE 1.

B. for $k = l^*_j - p_j + 2$ to $l^*_j + 1$,

(a) $\mathscr{A}$ knows the value of Y^{*j}_{k-1}.

(b) $\mathscr{A}$ then sets $X^{*j}_k = (0^{\frac{n}{2}} \| \lfloor Y^{*j}_{lk-1} \rfloor_{\frac{n}{2}}) \oplus C^{*j}_{k-1}$.

(c) It sends (tw^{*j}_k, X^{*j}_k) as a prediction query and recieves Y^{*j}_k.

Understanding the Reduction Game

The adversary $\mathscr{A}$'s actions on receiving an encryption query are quite simple. To each decryption query, $\mathscr{A}$ simply responds $\perp$. Now, we try to understand how the adversary $\mathscr{A}$ generates the commitments and the predictions depending upon the queries of $\mathscr{B}$. Notice that for each decryption query by $\mathscr{B}$, $\mathscr{A}$ sets an integer flag p taking values in $[-1, m_i]$. Moreover, it makes at least one commitment and at least one prediction for each decryption query.

For simplicity, we assume that $\mathscr{B}$ makes only one decryption query of the form (N^*, A^*, C^*, T^*).

Here, we only discuss the most complex case, i.e., when there exists an encryption query of the form (N, A, M) with response (C, T), such that $N^* = N$; $l^* = l$ and $T^* = T$. The adversary looks for the maximum possible common suffix between C

and C^*. Assume that the last p' blocks of C and C^* are identical. Then, depending on whether the most significant half of the last non-identical blocks of C and C' are identical or not, the flag is set to $p'+2$ or $p'+1$, respectively. With the help of $\mathcal{CH}$, $\mathscr{A}$ simulates the mF decryption protocol to compute Y^*_{l-p+1} before exiting PHASE 1.

Note that adversary $\mathscr{A}$ knows all the $\{Y_{l-p+2}, \cdots, Y_{l+1}\}$ values from the encryption transcript generated for $\mathscr{B}$. The adversary simply sets p commitments of the form $((N, l-p+2), \lceil C_{m^*-p+1} \rceil_{\frac{n}{2}}, \lfloor Y_{l-p+2} \rfloor_{\frac{n}{2}}), \cdots, ((N, l+1), \lceil C_{m^*} \rceil_{\frac{n}{2}}, \lfloor Y_{l+1} \rfloor_{\frac{n}{2}})$. Finally, $\mathscr{A}$ returns to simulating the decryption protocol starting from Y^*_{l-p+1} by sending prediction queries of the form $((N, l-p+k), X^*_{l-p+k})_{k\in[2,p+1]}$ to $\mathcal{CH}$.

Remark 7. *When $\mathscr{B}$ makes more than one decryption query, $\mathscr{A}$ doesn't make any prediction query before generating commitments corresponding to all the decryption queries.*

Let CBAD denote the event that $\mathscr{A}$ receives an encryption query of the form (N^i, A^i, M^i) to output a response of the form (C^i, T^i), such that, for some $1 \leq c \leq l_i$ we have $\lfloor X^i_c \rfloor_{\frac{n}{2}} = \lfloor Y^i_{c-1} \rfloor_{\frac{n}{2}} \oplus \lfloor \delta_M \rfloor_{\frac{n}{2}}$ for some arbitrary M.

Lemma 14.

$$\Pr[\texttt{CBAD}] \leq \frac{2\sigma_e}{2^{\frac{n}{2}}} + \mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}(\mathscr{C}').$$

Proof. Since CBAD occurs only during the encryption queries, by a standard reduction technique there exists a μ-TPRP adversary $\mathscr{C}'$ such that

$$\Pr[\texttt{CBAD}] \leq \Pr[\texttt{CBAD}_P] + \mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}(\mathscr{C}')$$

where $\texttt{CBAD}_P$ denotes the event CBAD when $\mathscr{A}$ has oracle access to P.

Now, Let $\mathscr{A}$ have oracle access to P. Then, during the i-th encryption query, we have $\lfloor X_c^i \rfloor_{\frac{n}{2}} = \begin{cases} \lfloor A_c^i \rfloor_{\frac{n}{2}} & \text{if } c \leq a_i \\ \lfloor Y_{c-1}^i \rfloor_{\frac{n}{2}} \oplus \lfloor \delta_{A^i} \rfloor_{\frac{n}{2}} & \text{if } c = a_i + 1 \\ \lfloor M_c^i \rfloor_{\frac{n}{2}} & \text{if } c > a_i + 1. \end{cases}$.

Now since all the Y_j^i values are generated uniformly at random and $\lfloor \delta_{A^i} \rfloor_{\frac{n}{2}} \neq \lfloor \delta_M \rfloor_{\frac{n}{2}}$ for any M. Hence, for any $i \in (q]$ and any M, $\Pr[\texttt{CBAD}] \leq \frac{1}{2^{\frac{n}{2}}}$. Since, δ_M takes at most 2 values, depending on whether $n|M$ or not, varying over all i, j, we have the lemma. □

Corollary 12. *If the event* CBAD *doesn't hold, then for any decryption query* $(N^{*j}, A^{*j}, C^{*j}, T^{*j})$*, such that* $N^{*j} = N^i$*,* $l_i > l_j^*$ *and* $T^{*j} = Y_{l_j^*+2}^i$*, then* $X_{l_j^*+1}^{*j} \neq X_{l_j^*+1}^i$*.*

Proof. Note that, $T^{*j} = Y_{l_j^*+2}^i \implies X_{l_j^*+2}^{*j} = X_{l_j^*+2}^i$. Now, since $Y_{l_j^*+1}^{*j} = X_{l_j^*+2}^{*j} \oplus \delta_{M^{*j}}$ and since CBAD doesn't hold, $Y_{l_j^*+1}^{*j} \neq Y_{l_j^*+1}^i$, which implies $X_{l_j^*+1}^{*j} \neq X_{l_j^*+1}^i$.

□

Proposition 22. *Suppose* $\mathscr{A}$ *never Aborts and* CBAD *never occurs. If* $(N^{*j}, A^{*j}, C^{*j}, T^{*j})$ *is a valid forgery, for some* $j \in [1, q_d]$ *then for some* $k \in [-1, p_j]$ *we have* (tw_k^{*j}, X_k^{*j}) *is a successful prediction query tuple.*

We postpone the proof of Proposition 22 to Subsection 6.4.3.

Proof of Theorem 8

For all encryption query of the form (N^i, A^i, M^i), $\mathscr{A}$ can correctly simulate as it has access to $\tilde{E}_K$.

Note that Proposition 22 means, that, given $\mathscr{A}$ doesn't abort and CBAD doesn't occur for any encryption query by $\mathscr{B}$, the (μ, σ_d)-mcp adversary $\mathscr{A}$ makes a valid prediction whenever the forging adversary $\mathscr{B}$ makes a successful forgery. Hence, by Proposition 22,

$$\Pr[\mathscr{A} \text{ wins } (\mu,\sigma_d)\text{-mcp game}]$$

$$\geq \Pr\left[\mathscr{B} \text{ Forges } i\text{-th query for some } i \in [1, q_d] | \mathscr{A} \text{ doesn't Abort } \cap \overline{\texttt{CBAD}}\right]$$

Hence,

$$\begin{aligned}
\Pr[\mathscr{B} \text{ Forges}] &\leq \Pr\left[\mathscr{B} \text{ Forges } i\text{-th query for some } i \in [1, q_d] | \mathscr{A} \text{ doesn't Abort} \cap \overline{\texttt{CBAD}}\right] \\
&\quad + \Pr[\mathscr{A} \text{ Aborts}] + \Pr[\texttt{CBAD}] \\
&\leq \Pr[\mathscr{A} \text{ wins } (\mu,\sigma_d)\text{-mcp game}] + \Pr[\mathscr{A} \text{ Aborts}] + \Pr[\texttt{CBAD}] \\
&\leq \mathbf{Adv}_{\tilde{E}}^{(\mu,\sigma_d)\text{-mcp}}(\mathscr{A}) + 2 \cdot \mathbf{Adv}_{\tilde{E}}^{\mu\text{-tprp}}(\mathscr{C}') \\
&\quad + \sigma\left(1 + \frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^{\mu} + \frac{2\sigma_e}{2^{\frac{n}{2}}}.
\end{aligned}$$

6.4.3 Proof of Proposition 22

Let $(N^{*j}, A^{*j}, C^{*j}, T^{*j})$ be a valid forgery. Depending on the value of p_j, we divide it into three cases.

CASE-1: If $p_j = -1$.

In the commitment phase the adversary $\mathscr{A}$ commits $((N^{*j}, l_j^* + 2), \lceil Y_{l_j^*+1}^{*j} \rceil_{\frac{n}{2}}, \lfloor T^{*j} \rfloor_{\frac{n}{2}})$ as described above.

Notice that if $N^i \neq N^{*j}$ for all encryption query of the form (N^i, A^i, M^i) then $(N^*j, l_j^* + 2)$ is fresh.

If $N^i = N^{*j}$ and $l_j^* = l_i$ but $T^i \neq T^{*j}$, then since $(N^*j, l_j^* + 2) = (N^i, l_i + 2)$, we must have $((N^{*j}, l_j^* + 2), X_{l_j^*+2}^{*j})$ is fresh.

If $N^i = N^{*j}$ and $l_j^* > l_i$ then we again have $(N^*j, l_j^* + 2)$ is fresh.

Let $N^i = N^{*j}$ and $l_j^* < l_i$. If $T^{*j} \neq Y_{l_j^*+2}$ then we have $((N^{*j}, l_j^* + 2), X_{l_j^*+2}^{*j})$ is fresh.

Hence, if any of the above condition is satisfied then $((N^{*j}, l_j^*+2), X_{l_j^*+2}^{*j})$ is fresh i.e. $((N^{*j}, l_j^* + 2), X_{l_j^*+2}^{*j})$ has never been queried before by $\mathscr{A}$ to $\mathcal{CH}$, $\lceil X_{l_j^*+2}^{*j} \rceil_{\frac{n}{2}} = \lceil Y_{l_j^*+1}^{*j} \rceil_{\frac{n}{2}}$ and $\tilde{E}_K((N^{*j}, l_j^* + 2), X_{l_j^*+2}^{*j}) = T^{*j}$. Hence, we see that $((N^{*j}, l_j^* + 2), X_{l_j^*+2}^{*j})$ is a valid prediction query with respect to the commitment $((N^{*j}, l_j^* + 2), \lceil Y_{l_j^*+1}^{*j} \rceil_{\frac{n}{2}}, \lfloor T^{*j} \rfloor_{\frac{n}{2}})$.

CASE-2: If $p_j = 0$

We have, $N^i = N^{*j}$, $l_j^* < l_i$ and $T^{*j} = Y_{l_j^*+2}$. Then, we must have $X_{l_j^*+2}^{*j} = X_{l_j^*+2}^{i}$ and

$$\lfloor Y_{l_j^*+1}^{*j} \rfloor_{\frac{n}{2}} = \lfloor Y_{l_j^*+1}^{i} \rfloor_{\frac{n}{2}} \oplus \lfloor D_{l_j^*+1}^{i} \rfloor_{\frac{n}{2}} \oplus \lfloor \delta_{M^{*j}} \rfloor_{\frac{n}{2}}$$

where $D_{l_j^*+1}^{i}$ is as defined in PHASE 2.

In the commitment phase, the adversary $\mathscr{A}$ commits $((N^{*j}, l_j^*+1), \lceil C_{m_j^*}^{*j} \rceil_{\frac{n}{2}}, \lfloor Y_{l_j^*+1}^{*j} \rfloor_{\frac{n}{2}})$ as described above. Now, by Proposition 12, we have $((N^{*j}, l_j^* + 1), X_{l_j^*+1}^{*j})$ is fresh. Moreover, it is a valid prediction query.

CASE-3: If $p_j \neq -1, 0$

There exist an $i \in [1, q_e]$ such that $N^{*j} = N^i, a_j^* + m_j^* = a_i + m_i = l_j^*, T^{*j} = T^i$.

Now consider the two cases :

I. First, let $p'_j \in \mathbb{Z}_{\geq 0}$ be such that $C^{*j}_{m_j^*-k} = C^i_{m_i-k}, \forall k \in [0, p'_j)$ and $C^{*j}_{m_j^*-p'_j} \neq C^i_{m_i-p'_j}$ but $\lceil C^{*j}_{m_j^*-p'_j} \rceil_{\frac{n}{2}} = \lceil C^i_{m_i-p'_j} \rceil_{\frac{n}{2}}$. In this case, $p_j = p'_j+2$. We have, by suffix property, $\Delta^j \neq 0$ and

$$\lfloor Y^{*j}_{l_j-p_j+2} \rfloor_{\frac{n}{2}} = \lfloor Y^i_{l_j-p_j+2} \rfloor_{\frac{n}{2}} \oplus \Delta^j.$$

$$\text{i.e. } X^{*j}_{l_j-p_j+2} \neq X^i_{l_j-p_j+2}.$$

II. Now, let $p'_j \in \mathbb{Z}_{\geq 0}$ be such that $C^{*j}_{m_j^*-k} = C^i_{m_i-k}, \forall k \in [0, p_j)$ and $\lceil C^{*j}_{m_j^*-p'_j} \rceil_{\frac{n}{2}} \neq \lceil C^i_{m_i-p'_j} \rceil_{\frac{n}{2}}$. Then $p_j = p'_j + 1$ and, by the suffix property,

$$\lfloor Y^{*j}_{l_j^*-p_j+2} \rfloor_{\frac{n}{2}} = \lfloor Y^i_{l_j^*-p_j+2} \rfloor_{\frac{n}{2}}.$$

Since $\lceil C^{*j}_{m_j^*-p_j+1} \rceil_{\frac{n}{2}} \neq \lceil C^i_{m_i-p_j+1} \rceil_{\frac{n}{2}}$,

$$X^{*j}_{l_j^*-p_j+2} \neq X^i_{l_j^*-p_j+2}.$$

Hence, we conclude that $(tw^{*j}_{l_j^*-p_j+2}, X^{*j}_{l_j^*-p_j+2})$ is fresh in both cases.

In the commitment phase, the adversary commits $(tw_k^{*j}, x_k^{*j}, y_k^{*j})$ for all $k \in [l_j^* - p_j + 2, l_j^* + 1]$.

If $(tw_{l_j - p_j + 2}^{*j}, X_{l_j^* - p_j + 2}^{*j})$ is a valid prediction query with respect to $(tw_{l_j^* - p_j + 2}^{*j}, x_{l_j^* - p_j + 2}^{*j}, y_{l_j^* - p_j + 1}^{*j})$, we are done.

If not, then $C_{m_j^* - p_j + 2}^{*j} = C_{m_i - p_j + 2}^{i}$.

$$\lfloor Y_{l_j^* - p_j + 2}^{*j} \rfloor_{\frac{n}{2}} \neq \lfloor Y_{l_j^* - p_j + 2}^{i} \rfloor_{\frac{n}{2}}.$$

$$\text{i.e. } X_{l_j^* - p_j + 3}^{*j} \neq X_{l_j^* - p_j + 3}^{i}.$$

Hence, we have $(tw_{l_j^* - p_j + 3}^{*j}, X_{l_j^* - p_j + 3}^{*j})$ is fresh.

Inductively, suppose $(tw_{l_j^*}^{*j}, X_{l_j^*}^{*j})$ is not a valid prediction query. Then as $C_{m_j^*}^{*j} = C_{m_i}^{i}$,

$$\lfloor Y_{l_j^*}^{*j} \rfloor_{\frac{n}{2}} \neq \lfloor Y_{l_j^*}^{i} \rfloor_{\frac{n}{2}}$$

$$\text{i.e. } X_{l_j^* + 1}^{*j} \neq X_{l_j^* + 1}^{i}$$

Hence $(tw_{l_j^* + 1}^{*j}, X_{l_j^* + 1}^{*j})$ is fresh.

Since $N^{*j} = N^i, a_j^* + m_j^* = a_i + m_i = l_j^*, T^{*j} = T^i \implies X_{l_j^* + 2}^{*j} = X_{l_j^* + 2}^{i}$. Hence,

$$Y_{l_j^* + 1}^{*j} = Y_{l_j^* + 1}^{i}.$$

Finally, since $(N^{*j}, A^{*j}, C^{*j}, T^{*j})$ is a valid forgery it must be that

$$\tilde{E}_K(tw^{*j}_{l^*_j+1}, X^{*j}_{l^*_j+1}) = Y^{*j}_{l^*_j+1}.$$

Hence, $(tw^{*j}_{l^*_j+1}, X^{*j}_{l^*_j+1})$ must be a valid prediction query. □

6.5 A Block Cipher-based Tweakable Block Cipher Construction

Let $\rho : \{0,1\}^n \to \{0,1\}^n$ be any bijective function and ρ^i denotes i consecutive applications of ρ. We call ρ, the key updation function (KUF) of the tweakable block cipher.

Definition 7. *Given any fixed KUF ρ, define*

$$\nu_\rho := \max_{l<2^n} \frac{1}{l} \cdot \Pr\left[r_K \leq l \middle| K \xleftarrow{\$} \{0,1\}^n\right].$$

Where for all $K \in \{0,1\}^n$, r_K is defined as the smallest positive integer such that $\rho^{r_K}(K) = K$.

Notice that if $\rho(K) = \alpha \cdot K$, where α is a primitive polynomial of degree n, then $\nu_\rho = 0$. Leurent et al. [75] showed that if ρ is the 11-th round-key function in the AES key scheduling algorithm, then $\nu_\rho \geq \frac{0.44}{14018661024}$.

Consider a block cipher $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. Then, for any integer $t > n$, we define the tweakable block cipher $\tilde{E} : \{0,1\}^n \times \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$ as, $\tilde{E}(K, tw, X) := E(K_{tw}, X)$, where $K_{tw} := \rho^i(E(K, N))$. Here, we parse tw to get $N = \lfloor tw \rfloor_n$; i is the decimal integer representation of $\lceil tw \rceil_{t-n}$.

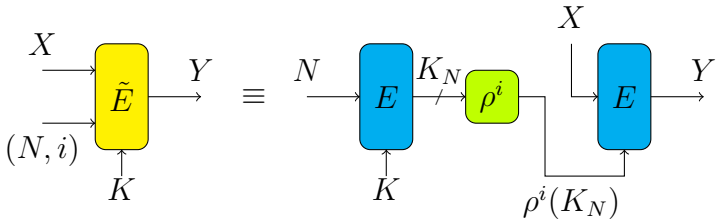


Figure 6-5: A block cipher based tweakable block cipher construction.

.

Remark 8. *If the key size κ of the block cipher is less than the state size n, then we can take the ρ function with domain and range $\{0,1\}^\kappa$ and chop K_N appropriately. If $\kappa > n$, we can generate the updated key suitably by multiple applications of ρ. Since there exist many popular block ciphers with $\kappa = n$, in this paper we restrict our analysis to these types of block ciphers only.*

6.5.1 Bounding μ-TPRP Security of $\tilde{E}$

Here we try to bound the μ-TPRP security of the tweakable block cipher $\tilde{E}$. Let $\mathscr{A}$ be any μ-respecting adversary playing the μ-TPRP game that makes at most t primitive queries and d online queries.
We assume that the adversary doesn't make repetitive or redundant queries.

The Ideal World and Analysis of Bad Events

Let $\mathcal{P}$ and $\mathcal{E}$ denote the index set of primitive queries and encryption queries respectively.

In ideal world, the oracle chooses random functions $P : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ and $Q : T \times \{0,1\}^n \to \{0,1\}^n$ such that for all $K \in \{0,1\}^n$ we have $P(K, \star)$ as a random permutation and for all $tw \in T$ we have $Q(tw, \star)$ as a random permutation.

PRIMITIVE QUERY: In the Ideal world for the i-th primitive query of the form (K^i, X^i) it computes $Y^i = P(K^i, X^i)$ and sends it as a response.

Define $\omega_t = (K^i, X^i, Y^i)_{i \in \mathcal{P}}$ to be the primitive transcript.

ONLINE QUERY: On receiving the i-th input query of the form $((N^i, j^i), X^i)$ it computes $Y^i = Q((N^i, j^i), X^i)$ and sends it as the response.

OFFLINE COMPUTATION : Oracle Chooses $K \in \{0,1\}^n$ uniformly at random. It then chooses a permutation $\Pi : \{0,1\}^n \rightarrow \{0,1\}^n$ uniformly at random from the set of all permutations over $\{0,1\}^n$. It then defines $K_{N^i} := \lfloor \Pi(N^i) \rfloor_n$ and $K^i = \rho^{j^i}(K_{N^i})$.

Define $\omega_d = (K, ((N^i, j^i), X^i, Y^i, K^i)_{i \in \mathcal{E}},)$ to be the online transcript.

Define $\omega = (\omega_t, \omega_d)$ be the transcript for the adversary in the ideal world.

Bad Events: We now look at the ideal world transcript ω. We identify all the possible events where there is an input or output collision between different types of query-response tuples in ω i.e. between the inputs or outputs of $(K^i, X_i, Y^i)_{i \in \mathcal{P}}$, $(K, tw^i, K^i)_{i \in \mathcal{E}}$ and $(K^i, X^i, Y^i)_{i \in \mathcal{E}}$. The 6 possible input collisions are:

(I_1) $(K^i, X^i) = (K^{i'}, X^{i'}), i, i' \in \mathcal{P}$

(I_2) $(K, tw^i) = (K, tw^{i'}), i, i' \in \mathcal{E}$

(I_3) $(K^i, X^i) = (K^{i'}, X^{i'}), i, i' \in \mathcal{E}$

(I_4) $(K, tw^i) = (K^{i'}, X^{i'}), i \in \mathcal{E}, i' \in \mathcal{P}$

(I_5) $(K, tw^i) = (K^{i'}, X^{i'}), i, i' \in \mathcal{E}$

(I_6) $(K^i, X^i) = (K^{i'}, X^{i'}), i \in \mathcal{E}, i' \in \mathcal{P}$.

Similarly, the 6 possible output collisions are :

(O_1) $(K^i, Y^i) = (K^{i'}, Y^{i'}), i, i' \in \mathcal{P}$

(O_2) $(K, K^i) = (K, K^{i'}), i, i' \in \mathcal{E}$

(O_3) $(K^i, Y^i) = (K^{i'}, Y^{i'}), i, i' \in \mathcal{E}$

(O_4) $(K, K^i) = (K^{i'}, Y^{i'}), i \in \mathcal{E}, i' \in \mathcal{P}$

(O_5) $(K, K^i) = (K^{i'}, Y^{i'}), i, i' \in \mathcal{E}$

(O_6) $(K^i, Y^i) = (K^{i'}, Y^{i'}), i \in \mathcal{E}, i' \in \mathcal{P}$.

We ignore cases I_1 and O_1 as the adversary doesn't make redundant queries. Similarly, we also ignore case I_2 as it simply means that the adversary has made multiple encryption queries with the same tweak. We consider the cases I_4, I_5, O_4 and O_5 as subcases of the event that $K = K^i$ for some $i \in \mathcal{E} \cup \mathcal{P}$. We call this event as `BAD1`. Similarly, we consider cases I_3, O_2 and O_3 as subcases of the event that for some $i, i' \in \mathcal{E}$ we have $K^i = K^{i'}$. Since the subcase where $tw^i = tw^{i'}$ is already considered in `BAD1` it is enough to consider the subcase where $tw^i \neq tw^{i'}$. Define this event as `BAD2`. We denote the event that case I_6 occurs as `BAD3`. Finally, we denote the event that case O_6 occurs as `BAD4`.

In notation:

`BAD1`: For some $i \in \mathcal{E} \cup \mathcal{P}$, we have $K^i = K$.

`BAD2`: For some $i_1 \neq i_2 \in \mathcal{E}$, we have $(N^{i_1}, j^{i_1}) \neq (N^{i_2}, j^{i_2})$ but $K^{i_1} = K^{i_2}$.

`BAD3`: For some $i \in \mathcal{E}$ and $i' \in \mathcal{P}$, we have $(K^i, X^i) = (K^{i'}, X^{i'})$.

`BAD4`:For some $i \in \mathcal{E}$ and $i' \in \mathcal{P}$, we have $(K^i, Y^i) = (K^{i'}, Y^{i'})$.

Note that, in the ideal world, BAD1 implies that the adversary couldn't guess the sectionret key. Further BAD2, BAD3 and BAD4 means that the input-output tuples in ω_t and ω_d are distinct i.e. permutation compatible.

Definition 8.

$$\texttt{BAD} = \bigcup_{i=1}^{4} \texttt{BADi}.$$

We call a transcript ω bad if event BAD occurs.

Lemma 15.

$$\Pr\left[\texttt{BAD}\right] \leq d \cdot \nu_\rho + \frac{t+d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{2\mu t}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu}.$$

Proof. Here, we try to bound the distinct bad events defined above.

Bounding BAD1: Fix $i \in \mathcal{P} \cup \mathcal{E}$. Since K is chosen uniformly at random, we have probability that $K^i = K$ is at most $\frac{1}{2^n}$. similar varying over all i,

$$\Pr\left[\texttt{BAD1}\right] \leq \frac{d+t}{2^n}$$

.

Bounding BAD2: This event can be divided into the following cases.

Case 1: $(N^{i_1} \neq N^{i_2})$ In this case, since Π is a random permutation, $K_{N^{i_1}} \neq K_{N^{i_2}}$ are distinct and independent. Hence probability that $K^{i_1} = K^{i_2}$ is at most $\frac{1}{2^n}$. Varying over all $i_1, i_2 \in \mathcal{E}$ we have,

$$\Pr\left[\textsc{Case}\ 1\right] \leq \frac{d^2}{2^{n+1}}.$$

CASE 2: $(N^{i_1} = N^{i_2}; j^{i_1} \neq j^{i_2})$ In this case we have $K_{N^{i_1}} = K_{N^{i_2}}$.

Hence CASE 2 event occurs if and only if, $\rho^{(j^{i_1} - j^{i_2})}(K_{N^i}) = K_{N^i}$ i.e., $(j^{i_1} - j^{i_2})$ is divisible by the periodicity of K_{N^i} (say $r_{K_{N^i}}$).

Note that queries of this form arise due to the encryption query of $\mathscr{B}$ with nonce N^i in the privacy game.

Let l_i denote the number of blocks in the encryption query of $\mathscr{B}$ with nonce N^i. Then for all i_i, i_2, such that $N^{i_1} = N^{i_2} = N^i$, we have $|j^{i_1} - j^{i_2}| \leq l_i$ i.e. $r_i \leq l_i$, and by Definition 7, probability that, CASE 2 holds is at most $l_i \cdot \nu_\rho$.

Now, varying over all possible i, and from the observation that $\sum_i l_i \leq d$, we have,

$$\Pr\left[\textsc{Case}\ 2\right] \leq \sum_i l_i \cdot \nu_\rho \leq d \cdot \nu_\rho.$$

Since the above two cases are mutually exclusive, we have

$$\Pr\left[\texttt{BAD2}\right] \leq \frac{d^2}{2^{n+1}} + d \cdot \nu_\rho.$$

BOUNDING `BAD3`: For a given $i' \in \mathcal{P}$, let the adversary make the primitive query $(K^{i'}, X^{i'})$. Then, there can be at most μ encryption query of the form $((N^{i_k}, j^{i_k}), X^{i'})_{k \in [1,\mu], i_k \in \mathcal{E}}$,

and hence, at most μ $(K^{i_k}, X^{i'})_{k\in[1,\mu],i_k\in\mathcal{E}}$ tuples. Now, since K^{i_k} are chosen uniformly at random during encryption query, we have, for a given $i_k \in \mathcal{E}$, probability that $K^{i_k} = K^{i'}$ is at most $\frac{1}{2^n}$. Hence, for a given $i' \in \mathcal{P}$, probability that $\exists i \in \mathcal{E}$ s.t. $(K^i, X^i) = (K^{i'}, X^{i'})$ is at most $\frac{\mu}{2^n}$. Varying over all i', we have

$$\Pr\left[\texttt{BAD3}\right] \leq \frac{\mu t}{2^n}.$$

BOUNDING BAD4: To bound BAD4 we first define an event BADY as follows:

BADY: $\exists i_1, \ldots i_{\mu+1} \in \mathcal{E}$ s.t. $Y^{i_k} = Y^{i_l} \quad \forall k, l \in [1, \mu+1]$.

Then by union bound we have

$$\Pr\left[\texttt{BAD4}\right] \leq \Pr\left[\texttt{BADY}\right] + \Pr\left[\texttt{BAD4}|\overline{\texttt{BADY}}\right].$$

BOUNDING BADY: Since for each $i \in \mathcal{E}$, Y^i is chosen uniformly at random, given $i_1, \ldots, i_{\mu+1} \in \mathcal{E}$, probability that $Y^{i_j} = Y^{i_j}$, for all $j \in [1, \mu+1]$ is at most $\frac{1}{(2^n)^\mu}$. Hence, varying over all choices of $i_1, \ldots, i_{\mu+1}$, we have

$$\Pr\left[\texttt{BADY}\right] \leq \frac{\binom{d}{\mu+1}}{(2^n)^\mu}.$$

BOUNDING BAD4$|\overline{\texttt{BADY}}$: For a given $i' \in \mathcal{P}$, let the adversary's primitive transcript be $(K^{i'}, \star, Y^{i'})$. Then, there can be at most μ encryption transcripts of the form

$((N^{i_k}, j^{i_k}), \star, Y^{i'})_{k\in[1,\mu],i_k\in\mathcal{E}}$, and hence, at most μ $(K^{i_k}, Y^{i'})_{k\in[1,\mu],i_k\in\mathcal{E}}$ tuples. Since K^{i_k} are chosen uniformly at random during encryption query, we have for a given $i_k \in \mathcal{E}$, probability that $K^{i_k} = K^{i'}$ is at most $\frac{1}{2^n}$. Hence, for a given $i' \in \mathcal{P}$ probability, that $\exists i \in \mathcal{E}$ s.t. $(K^i, Y^i) = (K^{i'}, Y^{i'})$ is at most $\frac{\mu}{2^n}$. Varying over all i', we have

$$\Pr\left[\texttt{BAD4}|\overline{\texttt{BADY}}\right] \leq \frac{\mu t}{2^n}.$$

Hence we get

$$\Pr\left[\texttt{BAD4}\right] \leq \frac{\binom{d}{\mu+1}}{(2^n)^\mu} + \frac{\mu t}{2^n}.$$

Finally, adding all the probabilities, we get the Lemma. □

Real World and Good Transcript Analysis

The real world has oracle E_K. All the primitive queries and the encryption queries are responded to based on the responses of E_K.

By good transcript, we mean any transcript which is not bad. Now consider a good transcript $\omega = (\omega_t, \omega_d)$. Let Θ_0 and Θ_1 be the transcript random variable obtained in the ideal world and real world respectively.

Note that by definition of the good transcript, the input-outputs of ω_t and ω_d in the ideal world are independent yet permutation compatible. Hence, we have

$$\Pr\left[\Theta_0 = \omega\right] = \prod_{t_i} \frac{1}{(2^n)_{t_i}} \times \frac{1}{2^n} \times \frac{1}{(2^n)_d} \times \frac{1}{(2^n)_d},$$

where t_i denotes the number of primitive queries with the key $K'_i \in \{0,1\}^n$. i.e, $\sum_i t_i = t$.

Now, note that in the real world the primitive queries and online queries are permutations compatible.

Hence, we have $\Pr[\Theta_1 = \omega] = \prod_{k_i} \frac{1}{(2^n)_{k_i}} \times \frac{1}{2^n} \times \frac{1}{(2^n)_d}$, where $k_i = d_i + t_i$ such that t_i denotes the number of primitive queries with key K_i and d_i denotes the number of encryption queries of the form (N^l, j^l, X) such that $K^l = K_i$. Note that $\sum_i k_i = d + t$.

Hence

$$\begin{aligned}\frac{\Pr[\Theta_1]}{\Pr[\Theta_0]} &= \frac{\prod_{t_i}(2^n)_{t_i} \times 2^n \times (2^n)_d \times (2^n)_d}{\prod_{k_i}(2^n)_{k_i} \times 2^n \times (2^n)_d} \\ &= \frac{\prod_{i}(2^n)_{t_i} \times (2^n)_d}{\prod_{i}(2^n)_{t_i+d_i}} \\ &= \frac{(2^n)_d}{\prod_i (2^n - t_i)_{d_i}} > 1.\end{aligned}$$

Hence by H-coefficient technique, we have Theorem 9.

Theorem 9.

$$\mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}(d,t) \leq d \cdot \nu_\rho + \frac{t+d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{2\mu t}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu}.$$

6.5.2 Bounding (μ, λ)-*mcp* Security of $\tilde{E}$

Here we try to bound the advantage of a μ-respecting adversary $\mathscr{A}$ making t primitive queries and d online queries playing the (μ, λ)-multi commitment prediction game with a challenger $\mathcal{CH}$. We assume that the adversary doesn't make repetitive or redundant queries.

Game 0:

We define the original (μ, λ)-*mcp* security game between $\mathscr{A}$ and $\mathcal{CH}$ as Game 0. Define $\mathcal{P}, \mathcal{E}_1$ and $\mathcal{E}_2$ respectively as the set of query indices of PHASE 1 primitive queries, PHASE 1 encryption queries and PHASE 2 encryption queries.

PHASE 1:

PRIMITIVE QUERY: For the i-th primitive query of the form $(K^i, X^i)_{i\in\mathcal{P}}$, $\mathcal{CH}$ computes $Y^i = E(K^i, X^i)$ and sends it as a response.

Define $\omega_t = (K^i, X^i, Y^i)_{i\in\mathcal{P}}$ to be the primitive transcript.

ONLINE QUERY: Oracle Chooses $K \in \{0,1\}^n$ uniformly at random. On receiving the i-th input query of the form $((N^i, j^i), X^i)_{i\in\mathcal{E}_1}$ if the query is μ-respecting then $\mathcal{CH}$ computes $K_{N^i} = E(K, N^i)$, $K^i = \rho^{j^i}(K_{N^i})$ and outputs $Y^i = E(K^i, X)$ as response. Else, it aborts.

PHASE 2:

COMMITMENT GENERATION: $\mathscr{A}$ sends λ commitments of the form $(tw^i, x^i, y^i)_{i\in[1,\lambda]}$ to $\mathcal{CH}$.

PRIMITIVE QUERIES: $\mathscr{A}$ doesn't make any primitive query in phase 2 .

PREDICTION QUERIES: Whenever $\mathscr{A}$ makes a fresh prediction query of the form $((N^i, j^i), X^i)$ for some $i \in \mathcal{E}_2$, $\mathcal{CH}$ computes $K_{N^i} = E(K, N^i)$, $K^i = \rho^{j^i}(K_{N^i})$ and outputs $Y^i = E(K^i, X)$ as response.

Let $\omega_{e_1} = ((N^i, j^i), X^i, Y^i)_{i\in\mathcal{E}_1}$ and $\omega_{e_2} = ((N^i, j^i), X^i, Y^i)_{i\in\mathcal{E}_2}$ be the phase 1 and phase 2 online transcript of the adversary.

Define $\omega_e = \omega_{e_1} \cup \omega_{e_2}$ as the overall online transcript of the adversary. Define $\omega = (\omega_t, \omega_e)$ as the transcript of $\mathscr{A}$.

Game 1:

We now define a newly modified security game called Game 1. Here, $\mathcal{CH}$ chooses random a function $Q : T \times \{0,1\}^n \to \{0,1\}^n$ such that for all $tw \in T$ we have $Q(tw, \star)$ is a random permutation. $\mathcal{CH}$ acts similar to a Game 0 challenger except in the case of phase-1 online queries.

PHASE 1 ONLINE QUERY: On receiving the i-th input query of the form $((N^i, j^i), X^i)_{i\in\mathcal{E}_1}$ if the query is μ-respecting then $\mathcal{CH}$ computes $Y^i = Q((N^i, j^i), X^i)$ and sends it as the response. Else, it aborts.

We say that any adversary $\mathscr{A}$ wins Game 1 if for some prediction query tuple $((N^k, j^k), X^k)$ there exist a commitment tuple (tw^i, x^i, y^i) such that

$$tw_i = (N^k, j^k);\ x_i = \lceil X^k \rceil_{\frac{n}{2}};\ \lfloor \tilde{E}_K(tw^i, X^k) \rfloor_{\frac{n}{2}} = y^i.$$

$$\mathbf{Adv}_{\tilde{E}}^{\text{Game 1}}(d,t) = \max_{\mathscr{A}} \Pr\left[\mathscr{A}^{\tilde{E}} \text{wins Game 1}\right].$$

where the maximum is taken over all adversaries $\mathscr{A}$ running in time t making at most d queries.

Proposition 23. *Given any (d,t)-adversary $\mathscr{A}$ playing Game 0 (or Game 1) there exists a $(d, t+2d)$-adversary $\mathscr{B}$ playing μ-TPRP security game such that,*

$$\Pr[\mathscr{A} \textit{ wins Game 0 }] \leq \Pr[\mathscr{A} \textit{ wins Game 1 }] + \mathbf{Adv}_{\tilde{E}}^{\mu\text{-tprp}}(\mathscr{B}).$$

Proof. We construct the $(d, t+2d)$-adversary $\mathscr{B}$ playing against an a μ-TPRP challenger $\mathcal{CH}$ as follows :

- Whenever $\mathscr{A}$ makes a phase-1 primitive query $\mathscr{B}$ makes the same primitive query to the μ-TPRP challenger $\mathcal{CH}$ and forwards the response to $\mathscr{A}$.

- $\mathscr{B}$ chooses $K \in \{0,1\}^n$ uniformly at random.

- Whenever $\mathscr{A}$ makes a phase-1 encryption query $\mathscr{B}$ makes the same encryption query to the μ-TPRP challenger $\mathcal{CH}$ and forwards the response to $\mathscr{A}$.

- On receiving the commitments from $\mathscr{A}$, $\mathscr{B}$ does nothing.

- Whenever $\mathscr{A}$ makes a prediction query of the form $((N^i, j^i), X^i)$ for some $i \in \mathcal{E}_2$, $\mathscr{B}$ makes a primitive query to $\mathcal{CH}$ to receive $K_{N^i} = E(K, N^i)$. It then conputes $K^i = \rho^{j^i}(K_{N^i})$ and makes a second primitive query to output $Y^i = E(K^i, X)$ as response.

- Whenever $\mathscr{A}$ wins (resp. loses) $\mathscr{B}$ sends 1 (resp. 0) to $\mathcal{CH}$.

From the construction, it is clear that whenever $\mathcal{CH}$ is a real (resp. ideal) oracle adversary $\mathscr{B}$ simulates perfectly as a Game 0 (resp. Game 1) oracle to the adversary $\mathscr{A}$. Hence

$$\Pr\left[\mathscr{A} \text{ wins Game 0}\right] = \Pr\left[\mathscr{B} \rightarrow 1 \mid \mathcal{CH} \text{ real }\right]$$

$$\Pr\left[\mathscr{A} \text{ wins Game 1}\right] = \Pr\left[\mathscr{B} \rightarrow 1 \mid \mathcal{CH} \text{ ideal }\right]$$

Hence,

$$\begin{aligned}\Big|\Pr\left[\mathscr{A} \text{ wins Game 0}\right] - \Pr\left[\mathscr{A} \text{ wins Game 1}\right]\Big| &= \Big|\Pr\left[\mathscr{B} \to 1 \mid \mathcal{CH} \text{ real}\right] \\ &\qquad - \Pr\left[\mathscr{B} \to 1 \mid \mathcal{CH} \text{ ideal }\right]\Big| \\ &= \mathbf{Adv}^{\mu\text{-tprp}}_{\tilde{E}}(\mathscr{B}).\end{aligned}$$

□

Proposition 24.

$$\mathbf{Adv}^{\mathit{Game\ 1}}_{\tilde{E}}(d,t) \leq \frac{\lambda t}{2^{\frac{3n}{2}}} + \frac{\lambda}{2^{\frac{n}{2}-1}}.$$

Proof. Consider the following event due to PHASE 2.

BAD5: For some $i \in [1,\lambda]$ and $i' \in \mathcal{P}$, we have a commitment $((N^i, j^i), x^i, y^i)$ such that $(K^i, x^i) = (K^{i'}, \lfloor X^{i'} \rfloor_{\frac{n}{2}})$ where $K^i := \rho^{j^i} \cdot E_K(N^i)$.

Claim 3.

$$\Pr\left[\mathtt{BAD5}\right] \leq \frac{\lambda t}{2^{\frac{3n}{2}}}.$$

Proof. Fix $i \in [1,\lambda]$ and $i' \in \mathcal{P}$. Since K_{N^i} is distributed uniformly at random, and there is no primitive query after commitment, we have probability that $(K^i, x^i) = (K^{i'}, \lceil X^{i'} \rceil_{\frac{n}{2}})$ is at most $\frac{1}{2^{\frac{3n}{2}}}$. varying over all i, i', we have the claim. □

Claim 4.

$$\Pr\left[\mathscr{A} \textit{ wins Game 1} \;\middle|\; \overline{\mathtt{BAD5}}\right] \leq \frac{\lambda}{2^{\frac{n}{2}-1}}.$$

Proof. Suppose $((N^i, j^i), X^i)$ is a valid prediction for some $i \in \mathcal{E}_2$. Let (tw^j, x^j, y^j) be the commitment corresponding to this prediction. Since BAD5 doesn't occur there is no primitive query of the form (K^i, X^i) in phase-1. Now suppose there are κ_i many primitive queries in phase-1 of the form $(K^i, \star)$. Then the probability that $\lfloor Y^i \rfloor_{\frac{n}{2}} = y_i$ is bounded by $\frac{2^{\frac{n}{2}}}{2^n - \kappa}$. Since $\kappa \leq t$, assuming $t \leq 2^{n-1}$ and varying over all i we have the claim.

□

Proposition 24 follows from Claims 3 and 4.

□

Theorem 10.

$$\mathbf{Adv}_{\tilde{E}}^{(\mu,\lambda)\text{-mcp}}(d,t) \leq d \cdot \nu_\rho + \frac{t+3d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{2\mu(t+2d)}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu} + \frac{\lambda t}{2^{\frac{3n}{2}}} + \frac{\lambda}{2^{\frac{n}{2}-1}}.$$

Proof. From Proposition 23 we have for any (d,t)-adversary $\mathscr{A}$ we have we have a $(d, t+2d)$-adversary $\mathscr{B}$ such that

$$\mathbf{Adv}^{\text{Game } 0}(\mathscr{A}) \leq \mathbf{Adv}^{\text{Game } 1}(\mathscr{A}) + \mathbf{Adv}_{\tilde{E}}^{\mu\text{-tprp}}(\mathscr{B})$$

Taking maximum over all such (d,t)-adversaries $\mathscr{A}$ we have

$$\mathbf{Adv}^{\text{Game } 0}(d,t) \leq \mathbf{Adv}^{\text{Game } 1}(d,t) + \mathbf{Adv}_{\tilde{E}}^{\mu\text{-tprp}}(d, t+2d).$$

Now plugging in the appropriate values from Proposition 24 and Theorem 7 we have Theorem 10. □

6.5.3 Some Instantiation of the New TBC

Consider a block cipher AES' which is a variation of AES [92] in the sense that, unlike the original scheme, it calls the AES-MixColumn operation in the last round and also outputs the 11-th round key in the AES key schedule. It then uses the new key to process the next data block. Since the key outputs only depend on the previous key input and are independent of the data inputs, this operation can be run in parallel and mixFeed can be viewed as an mF construction with $n = 128$ and the 11-th round key function in the AES key scheduling algorithm as the KUF.

Table 6.1: The RCON Values

i	1	2	3	4	5	6	7	8	9	10	11
RCON(i)	01	02	04	08	10	20	40	80	1b	36	6c

Next consider a TBC which uses AES' as the underlying block cipher and with abuse of notation call it AES'. Khairallah [72] observed that AES' is prone to practical forgery due to the existence of small periodic cycles in AES key schedule algorithm. Later, Leurent et. al. [75] confirmed Mustafa's observations by giving a practical attack with a success probability of 0.44(with data complexity 220GB). In our notations, if $l = 14018661024$, then $\Pr\left[r_K \leq l, K \xleftarrow{\$} \{0,1\}^{128}\right] \approx 0.44$. Plugging in this value in Definition 7, we get, for AES' TBC, $\sigma\nu_\rho \geq \frac{2^{46}\times 0.44}{14018661024} \gg 1$ in Theorems 9 and 10.

Algorithm 5 AES' Block Cipher. To apply a chain of block cipher, we perform an extra round of AES' Key-Schedule and use that round key as the initial key of the next call of AES'. As described in the Introduction the second output of Emodule only depends on the first input K and we define this function as $\phi(K)$.

function $\mathsf{E}(K; X)$
  $(W_{47}, \ldots, W_0) \leftarrow \mathsf{KeyGen}(K)$
  for $i = 1$ **to** 10 **do**
    $X \leftarrow X \oplus (W_{4i-1}, W_{4i-2}, W_{4i-3}, W_{4i-4})$
    $X \leftarrow \mathsf{SubBytes}(X)$
    $X \leftarrow \mathsf{ShiftRows}(X)$
    $X \leftarrow \mathsf{MixColumns}(X)$
  $X \leftarrow X \oplus (W_{43}, W_{42}, W_{41}, W_{40})$
  $K \leftarrow (W_{47}, W_{46}, W_{45}, W_{44})$
  return (X, K)

function $\mathsf{KeyGen}(K)$
  $(K_{15}, \ldots, K_0) \xleftarrow{8} K$
  for $i = 0$ **to** 3 **do**
    $W_i \leftarrow (K_{4i+3}, K_{4i+2}, K_{4i+1}, K_{4i})$
  for $i = 4$ **to** 47 **do**
    $Y \leftarrow W_{i-1}$
    if $i\%4 = 0$ **then**
      $Y \leftarrow \mathsf{SubWords}(Y \lll 8)$
      $Y \leftarrow Y \oplus \mathsf{RCON}_{i/4}$
    $W_i \leftarrow W_{i-4} \oplus Y$
  return $(W_{47}, \ldots, W_0)$

function $\mathsf{SubBytes}(X)$
  $(X_{15}, \ldots, X_0) \xleftarrow{8} X$
  for $i = 0$ **to** 15 **do**
    $X_i \leftarrow \mathrm{AS}(X_i)$
  return X

function $\mathsf{Shiftrows}(X)$
  $(X_{15}, \ldots, X_0) \xleftarrow{8} X$
  for $i = 0$ **to** 3 **do**
    for $j = 0$ **to** 3 **do**
      $Y_{4i+j} \leftarrow X_{4i+((j+i)\%4)}$
  return Y

function $\mathsf{MixColumns}(X)$
  $M \leftarrow \begin{pmatrix} 2 & 3 & 1 & 1 \\ 3 & 1 & 1 & 2 \\ 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 \end{pmatrix}$
  $Y \leftarrow M \cdot X$
  return Y

Next, we show that the weakness in AES'' TBC is only a weakness of the AES key schedule [92] and not of the TBC in general. More specifically, we describe a specific TBC construction called $\mathsf{AES}_{\mathsf{prim}}$ which is well secured.

For any primitive polynomial α of degree n, we define $\rho(x) = \alpha \cdot x \quad \forall x \in \{0,1\}^n$. Consider the TBC construction of Section 6.5 with ρ as its KUF. When AES is used as the underlying block cipher we call this TBC as $\mathsf{AES}_{\mathsf{prim}}$.

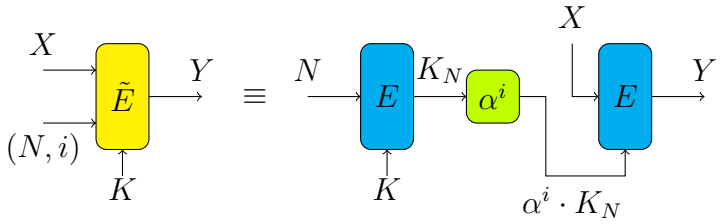


Figure 6-6: A tweakable block cipher with linear KUF. Here α is any primitive polynomial of degree n.

.

Corollary 13.

$$\mathbf{Adv}^{\mu\text{-tprp}}_{\mathit{AES}_{\mathit{prim}}}(d,t) \leq d \cdot \nu_\rho + \frac{t+d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{2\mu t}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu}.$$

Corollary 14.

$$\mathbf{Adv}^{(\mu,\lambda)\text{-mcp}}_{\mathit{AES}_{\mathit{prim}}}(d,t) \leq d \cdot \nu_\rho + \frac{t+3d}{2^n} + \frac{d^2}{2^{n+1}} + \frac{2\mu(t+2d)}{2^n} + \frac{\binom{d}{\mu+1}}{(2^n)^\mu} + \frac{\lambda t}{2^{\frac{3n}{2}}} + \frac{\lambda}{2^{\frac{n}{2}-1}}.$$

Remark 9. *If the linearity of multiplication by a primitive polynomial α becomes problematic under some block cipher designs, one could define $\rho(K) = P^{-1}(\alpha \cdot P(K))$ for some nonlinear permutation P. This preserves the cycle structure of the multiplication by α but can be arbitrarily nonlinear depending on P.*

6.6 mF Under the New TBC

In this section, we consider the mF construction under the new TBC construction defined in Section 6.5. Note that in such a case the mF mode of AEAD can be implemented as a block cipher-based construction. Unlike Figure 6-5, where each TBC call can be seen to be using two block cipher calls, we can process the nonce N with key K only once, to get K_N and then we store K_N as the initial key. The rekeying can be done in parallel, by applying the KUF (ρ) to the previous key while processing each data block. In this way, we can process the whole encryption-decryption query with only 1 extra block cipher call. Further, the number of bits processed per primitive block cipher call is asymptotical to n.

Theorem 11.

$$\mathbf{Adv}^{\text{priv}}_{\mathsf{mF},\tilde{E}}(\sigma,t) \leq \sigma\nu_\rho + \frac{t+\sigma}{2^n} + \frac{\sigma^2}{2^{n+1}} + \frac{2\mu t}{2^n} + \frac{\binom{\sigma}{\mu+1}}{(2^n)^\mu} + \sigma\left(1+\frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^\mu.$$

Theorem 12.

$$\begin{aligned}\mathbf{Adv}^{\text{forge}}_{\mathsf{mF},\tilde{E}}(\sigma,t) \leq 3\sigma\cdot\nu_\rho + \frac{3(1+2\mu)t}{2^n} + \frac{3\sigma^2}{2^{n+1}} + \frac{(5+4\mu)\sigma}{2^n} + \frac{3\binom{\sigma}{\mu+1}}{(2^n)^\mu} \\ + \frac{\sigma t}{2^{\frac{3n}{2}}} + \frac{\sigma}{2^{\frac{n}{2}-2}} + \sigma\left(1+\frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^\mu.\end{aligned}$$

Proof. Theorems 11 and 12 can be derived from Theorem 7 and 8 respectively by appropriately plugging in the security bounds for $\tilde{E}$ derived in Section 6.5. □

Corollary 15. *Consider the* mF *mode of AEAD with* AES_{prim} *TBC and call it* mF_{prim}*. For any adversary running in time t and making at most q encryption and decryption(in*

case of forgery) query with total of at most σ blocks,

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{mF}_{prim}}(\sigma,t) \leq \frac{t+\sigma}{2^n} + \frac{\sigma^2}{2^{n+1}} + \frac{2\mu t}{2^n} + \frac{\binom{\sigma}{\mu+1}}{(2^n)^\mu} + \sigma\left(1+\frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^\mu.$$

$$\begin{aligned}\mathbf{Adv}^{\mathrm{forge}}_{\mathsf{mF}_{prim},\tilde{E}}(\sigma,t) \leq{}& \frac{3(1+2\mu)t}{2^n} + \frac{3\sigma^2}{2^{n+1}} + \frac{(5+4\mu)\sigma}{2^n} + \frac{3\binom{\sigma}{\mu+1}}{(2^n)^\mu} \\ &+ \frac{\sigma t}{2^{\frac{3n}{2}}} + \frac{\sigma}{2^{\frac{n}{2}-2}} + \sigma\left(1+\frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^\mu.\end{aligned}$$

Where n is the state size and μ is the number of multi collisions allowed in the input of the tweakable block cipher. For all calculations, take $\mu \geq 4$.

6.6.1 Interpretation of the Above Bounds

According to NIST requirement, $\sigma \leq 2^{46}$ and $t \leq 2^{112}$. Following the recommendation in [38], we take $n = \kappa = 128$. Then, taking $\mu = 4$, we have $\sigma\left(1+\frac{(\mu+1)^2}{2^n}\right)\left(\frac{\sigma}{2^{\frac{n}{2}}}\right)^\mu < 2^{-25}$, and hence, the dominating term is $\frac{2\mu t}{2^n}$ in Theorem 11 and $\frac{5\mu t}{2^n} + \frac{3\sigma}{2^{\frac{n}{2}}}$ in Theorem 12, which are both less than 2^{-10}. Hence, by Corollary 15 we conclude that the $\mathsf{mF}_{\mathsf{prim}}$ mode is well secured within the complexity bounds specification of NIST.

6.7 mF Mode as a Lightweight AEAD

In this section, we try to give a theoretical comparison between the $\mathsf{mF}_{\mathsf{prim}}$ mode and some other tweakable block cipher designs in the NIST LwC [93] competition. Ignoring the different types of overheads required in practical implementations, we define the state size as the number of bits required to hold the key, auxiliary keys such as masking key (if any), block cipher state, and round key.

We have tabulated theoretical comparisons of different TBC-based AEAD schemes. A more practical, implementation-based comparison is beyond the scope of this paper and can be left as a future research problem.

Table 6.2: A theoretical comparison of different TBC-based lightweight AEAD schemes. Here the TBC of $\mathsf{mF}_{\mathsf{prim}}$ is considered with AES-128/128 as the underlying block cipher.

Mode	State Size (includes key-size)	Block Size	Tweak Size	# Pass	Bits Processed per primitive call	Inverse-free
Romulus-N1 [65]	512	128	384	1	128	yes
Romulus- M1 [65]	512	128	384	2	64	yes
SKINNY-AEAD (M1) [14]	640	128	384	1	128	No
QAMELEON [5]	640	128	384	1	128	No
LILLIPUT-I-128 [1]	576	128	320	1	128	No
$\mathsf{mF}_{\mathsf{prim}}$	384	128	0	1	128	yes

6.8 Conclusion

In this chapter, our aim was to construct a lightweight block-cipher-based AEAD, which attains the maximum possible message absorption rate per block cipher call using small effective state-size. We started by constructing a TBC-based AEAD scheme mF, which can be viewed as an abstraction of $\mathsf{mixFeed}$ mode. We have proven that the security of the said mF mode can be reduced to the security of its underlying TBC. We constructed a new block cipher-based TBC construction and bound the security advantages of any adversary against mF mode using this TBC. We have tried to interpret the results of [75] in our notations and confirmed the observation made in [72] that in the case of $\mathsf{mixFeed}$ the security of the underlying TBC depends on the periodicity of the AES key scheduling algorithm. Finally, to show that the said weakness is restricted to the use of AES key scheduling algorithm and that it doesn't affect the mF mode in general, we have constructed an explicit TBC construction and showed that the $\mathsf{mF}_{\mathsf{prim}}$ mode using this TBC achieves the desired security within the NIST parameters.

Conclusion

In this thesis, inspired by the immense popularity of lightweight devices, we analyzed two popular paradigms of lightweight authenticated encryption protocols, namely, Sponge-type AEADs and TBC-based AEADs. We showed that most of the Sponge types AEADs can be viewed as an instantiation of the Transform-then-Permute protocol. A tight security bound is achieved for a particular class of the TtP protocol where decryption feedback functions are invertible. For the general Sponge, improved security bound without any matching attack can be derived. Keeping in mind the ongoing NIST LwC requirements, our results show that designs like Beetle, SpoC, Ascon, Xoodyak, etc. are well secured in the lightweight sense. Next, we focused on increasing the rate of data absorption in a Sponge type construction. We showed that with a proper extra-state initialization and suitably modified full-rate feedback functions used in popular schemes such as CoFB, HyENA etc., we can construct Sponge type full-rate AEAD schemes which absorb data at the maximum possible rate and still meet desired security requirements. Note that the tightness of security bounds in all these Sponge-based constructions is based on the tightness of bounding the advantages of graph-structure adversaries defined in this thesis. Hence further analysis of these graph structures is an interesting research problem.

Finally, we turned our focus toward block cipher-based AEAD schemes. We designed and analyzed a generalized TBC-based AEAD scheme called mF. We also constructed a generalized block cipher-based tweakable block cipher, which can be used as the underlying TBC in the mF protocol so that the mF construction can be viewed as a block cipher-based AEAD scheme. Finally, we gave an instantiation called $\mathsf{mF}_{\mathsf{prim}}$ and showed that it is well secured in the lightweight sense. We also shared a theoretical comparison between $\mathsf{mF}_{\mathsf{prim}}$ and some existing lightweight TBC-based AEAD schemes.

In the end, implementation security (SCA/faults) analysis of all the above constructions is beyond the scope of this thesis and is a significant open problem for the future.

Bibliography

[1] Adomnicai, A., Berger, T. P., Clavier, C., Francq, J., Huynh, P., Lallemand, V., Le Gouguec, K., Minier, M., Reynaud, L., and Thomas, G. (2019). Lilliput-ae: a new lightweight tweakable block cipher for authenticated encryption with associated data.

[2] AlTawy, R., Gong, G., He, M., Jha, A., Mandal, K., Nandi, M., and Rohit, R. (2019). Spoc. Submission to NIST LwC Standardization Process (Round 2).

[3] AlTawy, R., Gong, G., Mandal, K., and Rohit, R. (2020). Wage: an authenticated encryption with a twist. *IACR Transactions on Symmetric Cryptology*, pages 132–159.

[4] Andreeva, E., Deprez, A., Pittevils, J., Roy, A., Bhati, A. S., and Vizár, D. (2020). New results and insighs on forkae. In *NIST LWC workshop*.

[5] Avanzi, R., Banik, S., Bogdanov, A., Dunkelman, O., Huang, S., and Regazzoni, F. (2019). Qameleon v. 1.0. Submission to NIST LwC Standardization Process (Round 1). `https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/qameleon-spec.pdf`.

[6] Baeza-Yates, R. A. and Gonnet, G. H. (1991). *Handbook of Algorithms and Data Structures in Pascal and C*. Addison-Wesley.

[7] Bagheri, N. (2015). Linear cryptanalysis of reduced-round simeck variants. In *International Conference on Cryptology in India*, pages 140–152. Springer.

[8] Banik, S., Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M., Peyrin, T., Sasaki, Y., Sim, S. M., and Todo, Y. (2020). Gift-cofb: Nist lwc second-round candidate status update. *Energy (nJ/128-bit)*, 5:0.

[9] Bao, Z., Chakraborti, A., Datta, N., Guo, J., Nandi, M., Peyrin, T., and Yasuda, K. (2019). PHOTON-Beetle. Submission to NIST LwC Standardization Process (FINALIST).

[10] Bariant, A., David, N., and Leurent, G. (2020). Cryptanalysis of forkciphers. *IACR Transactions on Symmetric Cryptology*, 2020(1):233–265.

[11] Beierle, C., Biryukov, A., Cardoso dos Santos, L., Großschädl, J., Perrin, L., Udovenko, A., Velichkov, V., and Wang, Q. (2020a). Alzette: A 64-bit arx-box. In *Annual International Cryptology Conference*, pages 419–448. Springer.

[12] Beierle, C., Biryukov, A., dos Santos, L. C., Großschädl, J., Perrin, L., Udovenko, A., Velichkov, V., and Wang, Q. (2020b). Lightweight aead and hashing using the sparkle permutation family. *IACR Transactions on Symmetric Cryptology*, pages 208–261.

[13] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., and Sim, S. M. (2020c). Skinny-aead and skinny-hash. *IACR Transactions on Symmetric Cryptology*, pages 88–131.

[14] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., and Sim, S. M. (2019). Skinny-aead and skinny-hash. Submission to NIST LwC Standardization Process (Round 2).

`https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/SKINNY-spec-round2.pdf`.

[15] Bellizia, D., Berti, F., Bronchain, O., Cassiers, G., Duval, S., Guo, C., Leander, G., Leurent, G., Levi, I., Momin, C., et al. (2020). Spook: Sponge-based leakage-resistant authenticated encryption with a masked tweakable block cipher. *IACR Transactions on Symmetric Cryptology*, 2020(S1):295–349.

[16] Bernstein, D. J., Gilbert, H., and Turan, M. S. (2020). Observations on comet. *Cryptology ePrint Archive*.

[17] Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2007). Sponge functions. In *ECRYPT Hash Workshop 2007. Proceedings*.

[18] Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2008). On the indifferentiability of the sponge construction. In *Advances in Cryptology - EUROCRYPT 2008. Proceedings*, pages 181–197.

[19] Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2010). Sponge-based pseudo-random number generators. In *Cryptographic Hardware and Embedded Systems, CHES 2010. Proceedings*, pages 33–47.

[20] Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2011a). Duplexing the sponge: Single-pass authenticated encryption and other applications. In *Selected Areas in Cryptography - 18th International Workshop, SAC 2011. Revised Selected Papers*, pages 320–337.

[21] Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2011b). On the security

of the keyed sponge construction. In *Symmetric Key Encryption Workshop 2011. Proceedings*.

[22] Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2013). Keccak. In *Advances in Cryptology - EUROCRYPT 2013. Proceedings*, pages 313–314.

[23] Bhattacharjee, A., López, C. M., List, E., and Nandi, M. (2021). The oribatida v1. 3 family of lightweight authenticated encryption schemes. *Journal of Mathematical Cryptology*, 15(1):305–344.

[24] Bogdanov, A. (2017). *Lightweight Cryptography for Security and Privacy: 5th International Workshop, LightSec 2016, Aksaray, Turkey, September 21-22, 2016, Revised Selected Papers*, volume 10098. Springer.

[25] Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., and Verbauwhede, I. (2012). Spongent: the design space of lightweight cryptographic hashing. *IEEE Transactions on Computers*, 62(10):2041–2053.

[26] Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., and Verbauwhede, I. (2013). SPONGENT: the design space of lightweight cryptographic hashing. *IEEE Trans. Computers*, 62(10):2041–2053.

[27] Cai, J., Wei, Z., Zhang, Y., Sun, S., and Hu, L. (2019). Zero-sum distinguishers for round-reduced gimli permutation. In *ICISSP*, pages 38–43.

[28] Canteaut, A., Duval, S., Leurent, G., Naya-Plasencia, M., Perrin, L., Pornin, T., and Schrottenloher, A. (2020). An update on saturnin. *NIST Lightweight Crypto Standardization process (Round 2)*.

[29] Chakraborti, A., Datta, N., Jha, A., Lopez, C. M., Nandi, M., and Sasaki, Y. (2019a). Lotus and locus aead: Hardware benchmarking and security.

[30] Chakraborti, A., Datta, N., Jha, A., Mancillas-López, C., Nandi, M., and Sasaki, Y. (2019b). Int-rup secure lightweight parallel ae modes. *IACR Transactions on Symmetric Cryptology*, pages 81–118.

[31] Chakraborti, A., Datta, N., Jha, A., Mancillas-López, C., Nandi, M., and Sasaki, Y. (2020a). Estate: A lightweight and low energy authenticated encryption mode. *IACR Transactions on Symmetric Cryptology*, pages 350–389.

[32] Chakraborti, A., Datta, N., Jha, A., Mitragotri, S., and Nandi, M. (2019c). Security analysis of hyena authenticated encryption mode.

[33] Chakraborti, A., Datta, N., Jha, A., Mitragotri, S., and Nandi, M. (2020b). From combined to hybrid: making feedback-based ae even smaller. *IACR Transactions on Symmetric Cryptology*, pages 417–445.

[34] Chakraborti, A., Datta, N., Jha, A., and Nandi, M. (2019d). HyENA. Submission to NIST LwC Standardization Process (Round 2).

[35] Chakraborti, A., Datta, N., Nandi, M., and Yasuda, K. (2018). Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241.

[36] Chakraborti, A., Iwata, T., Minematsu, K., and Nandi, M. (2017). Blockcipher-based authenticated encryption: How small can we go? In *Cryptographic Hardware and Embedded Systems - CHES 2017. Proceedings*, pages 277–298.

[37] Chakraborty, B., Jha, A., and Nandi, M. (2020). On the security of sponge-type authenticated encryption modes. *IACR Transactions on Symmetric Cryptology*, pages 93–119.

[38] Chakraborty, B. and Nandi, M. (2019a). mixFeed. Submission to NIST LwC Standardization Process (Round 2). `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/mixFeed-spec-round2.pdf`.

[39] Chakraborty, B. and Nandi, M. (2019b). ORANGE. Submission to NIST LwC Standardization Process (Round 2).

[40] Chang, D., Datta, N., Dutta, A., Mennink, B., Nandi, M., Sanadhya, S., and Sibleyras, F. (2019). Release of unverified plaintext: Tight unified model and application to anydae. *IACR Transactions on Symmetric Cryptology*, pages 119–146.

[41] Chang, D. and Turan, M. S. (2021). Recovering the key from the internal state of grain-128aead. *Cryptology ePrint Archive*.

[42] Chen, S. and Steinberger, J. P. (2014). Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014. Proceedings*, pages 327–350.

[43] Daemen, J., Mennink, B., and Assche, G. V. (2017). Full-state keyed duplex with built-in multi-user support. In *Advances in Cryptology - ASIACRYPT 2017. Proceedings, Part II*, pages 606–637.

[44] Datta, N., Jha, A., Mège, A., and Nandi, M. (2019). Breaking remus and tgif in the light of nist lightweight cryptography standardization project. `https:`

//csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2019/documents/papers/breaking-remus-and-tgif-lwc2019.pdf.

[45] Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Mennink, B., Primas, R., and Unterluggauer, T. (2020a). Isap v2. 0. *IACR Transactions on Symmetric Cryptology*, pages 390–416.

[46] Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. (2016). Ascon. CAESAR recommendation for lightweight applications.

[47] Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. (2021). Ascon v1. 2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34(3):1–42.

[48] Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. (2019a). ASCON. Submission to NIST LwC Standardization Process (FINALIST).

[49] Dobraunig, C., Mendel, F., and Mennink, B. (2019b). Round 1 official comments : ORANGE. Submission to NIST LwC Standardization Process .

[50] Dobraunig, C. and Mennink, B. (2019). Leakage resilience of the isap mode: a vulgarized summary. In *NIST Lightweight Cryptography Workshop*, volume 2019, page 23.

[51] Dobraunig, C., Rotella, Y., and Schoone, J. (2020b). Algebraic and higher-order differential cryptanalysis of pyjamask-96. *IACR Transactions on Symmetric Cryptology*, pages 289–312.

[52] Dworkin, M. J. (2015). Sha-3 standard: Permutation-based hash and extendable-output functions.

[53] Feller, W. (2008). *An introduction to probability theory and its applications, vol 2*. John Wiley & Sons.

[54] Flórez Gutiérrez, A., Leurent, G., Naya-Plasencia, M., Perrin, L., Schrottenloher, A., and Sibleyras, F. (2020). New results on gimli: full-permutation distinguishers and improved collisions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 33–63. Springer.

[55] Gonnet, G. H. (1981). Expected length of the longest probe sequence in hash code searching. *J. ACM*, 28(2):289–304.

[56] Goudarzi, D., Jean, J., Kölbl, S., Peyrin, T., Rivain, M., Sasaki, Y., and Sim, S. M. (2020). Pyjamask: Block cipher and authenticated encryption with highly efficient masked implementation. *IACR Transactions on Symmetric Cryptology*, pages 31–59.

[57] Gueron, S., Jha, A., and Nandi, M. (2019a). Comet. Submission to NIST LwC Standardization Process (Round 1). `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/comet-spec-round2.pdf`.

[58] Gueron, S., Jha, A., and Nandi, M. (2019b). On the security of comet authenticated encryption scheme. In *NIST Lightweight Cryptography Workshop*, volume 2019.

[59] Gueron, S., Jha, A., and Nandi, M. (2021). Revisiting the security of comet authenticated encryption scheme. In *International Conference on Cryptology in India*, pages 3–25. Springer.

[60] Guo, C., Khairallah, M., and Peyrin, T. (2020). Aet-lr: rate-1 leakage-resilient aead based on the romulus family. In *NIST LWC Workshop*.

[61] Hamburg, M. (2017). Cryptanalysis of 22 1/2 rounds of gimli. *Cryptology ePrint Archive*.

[62] Hao, Y., Leander, G., Meier, W., Todo, Y., and Wang, Q. (2020). Modeling for three-subset division property without unknown subset. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 466–495. Springer.

[63] Huang, S., Wang, X., Xu, G., Wang, M., and Zhao, J. (2017). Conditional cube attack on reduced-round keccak sponge function. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 259–288. Springer.

[64] Iwata, T., Khairallah, M., Minematsu, K., and Peyrin, T. (2019a). Remus. Submission to NIST LwC Standardization Process (Round 1). `https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/Remus-spec.pdf`.

[65] Iwata, T., Khairallah, M., Minematsu, K., and Peyrin, T. (2019b). Romulus. Submission to NIST LwC Standardization Process (Round 2). `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/Romulus-spec-round2.pdf`.

[66] Iwata, T., Khairallah, M., Minematsu, K., and Peyrin, T. (2020). Duel of the titans: the romulus and remus families of lightweight aead algorithms. *IACR Transactions on Symmetric Cryptology*, pages 43–120.

[67] Iwata, T., Khairallah, M., Minematsu, K., Peyrin, T., Sasaki, Y., Sim, S. M.,

and Sun, L. (2019c). Thank goodness it's friday (tgif). Submission to NIST LwC Standardization Process (Round 1). https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/TGIF-spec.pdf.

[68] Jovanovic, P., Luykx, A., and Mennink, B. (2014). Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 85–104. Springer.

[69] Jovanovic, P., Luykx, A., Mennink, B., Sasaki, Y., and Yasuda, K. (2019). Beyond conventional security in sponge-based authenticated encryption modes. *J. Cryptology*, 32(3):895–940.

[70] Kairallah, M. M. M., Rohit, R., and Sarkar, S. (2019). Round 2 official comments : ORANGE. Submission to NIST LwC Standardization Process.

[71] Khairallah, M. (2019a). Forgery attack on mixfeed in the nonce-misuse scenario. Cryptology ePrint Archive, Report 2019/457. https://eprint.iacr.org/2019/457.

[72] Khairallah, M. (2019b). Weak keys in the rekeying paradigm: Application to comet and mixfeed. Cryptology ePrint Archive, Report 2019/888. https://eprint.iacr.org/2019/888.

[73] Kraleva, L., Posteuca, R., and Rijmen, V. (2020). Cryptanalysis of the permutation based algorithm spoc. In *International Conference on Cryptology in India*, pages 273–293. Springer.

[74] Le, D.-P., Lu, R., and Ghorbani, A. A. (2022). Improved fault analysis on simeck ciphers. *Journal of Cryptographic Engineering*, 12(2):169–180.

[75] Leurent, G. and Pernot, C. (2020). New representations of the aes key schedule. Cryptology ePrint Archive, Report 2020/1253. `https://eprint.iacr.org/2020/1253`.

[76] Liskov, M., Rivest, R. L., and Wagner, D. (2002). Tweakable block ciphers. In *Annual International Cryptology Conference*, pages 31–46. Springer.

[77] Liu, F., Isobe, T., and Meier, W. (2019). Cube-based cryptanalysis of subterranean-sae. *IACR Transactions on Symmetric Cryptology*, pages 192–222.

[78] Liu, F., Isobe, T., and Meier, W. (2020a). Automatic verification of differential characteristics: Application to reduced gimli. In *Annual International Cryptology Conference*, pages 219–248. Springer.

[79] Liu, F., Isobe, T., Meier, W., and Yang, Z. (2020b). Algebraic attacks on round-reduced keccak/xoodoo. *Cryptology ePrint Archive*.

[80] Liu, J., Liu, G., and Qu, L. (2020c). A new automatic tool searching for impossible differential of nist candidate ace. *Mathematics*, 8(9):1576.

[81] Liu, Y., Sun, S., and Li, C. (2021). Rotational cryptanalysis from a differential-linear perspective: Practical distinguishers for round-reduced friet, xoodoo, and alzette. *Cryptology ePrint Archive*.

[82] Lu, J., Liu, Y., Ashur, T., Sun, B., and Li, C. (2020). Rotational-xor cryptanalysis of simon-like block ciphers. In *Australasian Conference on Information Security and Privacy*, pages 105–124. Springer.

[83] Matsaglia, G. and PH Styan, G. (1974). Equalities and inequalities for ranks of matrices. *Linear and multilinear Algebra*, 2(3):269–292.

[84] Mege, A. (2019). Slide attack on clx-128. In *Proceedings of the Lightweight Cryptography Workshop*, page 169p.

[85] Mennink, B. (2018). Key prediction security of keyed sponges. *IACR Transactions on Symmetric Cryptology*, 2018(4):128–149.

[86] Mennink, B. and Neves, S. (2017). Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In *Advances in Cryptology - CRYPTO 2017. Proceedings, Part III*, pages 556–583.

[87] Mennink, B., Reyhanitabar, R., and Vizár, D. (2015). Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In *Advances in Cryptology - ASIACRYPT 2015. Proceedings, Part II*, pages 465–489.

[88] Micciancio, D. and Ristenpart, T. (2020). *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part III*, volume 12172. Springer Nature.

[89] Mouha, N. (2015). The design space of lightweight cryptography. In *NIST Lightweight Cryptography Workshop 2015*.

[90] Naito, Y., Matsui, M., Sugawara, T., and Suzuki, D. (2019). Saeb: A lightweight blockcipher-based aead mode of operation. *Cryptology ePrint Archive*.

[91] National Institute of Standards and Technology (2019). Lightweight cryptography. *Available at* `https://csrc.nist.gov/Projects/lightweight-cryptography`.

[92] NIST (2001). Announcing the ADVANCED ENCRYPTION STANDARD (AES). Fedral Information Processing Standards Publication FIPS 197, National Institute of Standards and Technology, U. S. Department of Commerce.

[93] NIST (2018). Submission requirements and evaluation criteria for the Lightweight Cryptography Standardization Process. `https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf`.

[94] Patarin, J. (1991). *Etude des Générateurs de Permutations Pseudo-aléatoires Basés sur le Schéma du DES*. PhD thesis, Université de Paris.

[95] Patarin, J. (2008). The "coefficients H" technique. In *Selected Areas in Cryptography - SAC 2008. Revised Selected Papers*, pages 328–345.

[96] Raab, M. and Steger, A. (1998). "balls into bins" - A simple and tight analysis. In *Randomization and Approximation Techniques in Computer Science, Second International Workshop, RANDOM'98. Proceedings*, pages 159–170.

[97] Ramezanpour, K., Abdulgadir, A., Diehl, W., Kaps, J.-P., and Ampadu, P. (2020). Active and passive side-channel key recovery attacks on ascon. In *Proc. NIST Lightweight Cryptogr. Workshop*, pages 1–27.

[98] Rogaway, P. (2002). Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 98–107.

[99] Rogaway, P. (2004). Efficient instantiations of tweakable blockciphers and refinements to modes ocb and pmac. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 16–31. Springer.

[100] Saha, D., Sasaki, Y., Shi, D., Sibleyras, F., Sun, S., and Zhang, Y. (2020). On

the security margin of tinyjambu with refined differential and linear cryptanalysis. *IACR Transactions on Symmetric Cryptology*, pages 152–174.

[101] Sedgewick, R. and Flajolet, P. (1996). *An introduction to the analysis of algorithms*. Addison-Wesley-Longman.

[102] Shi, T., Wu, W., Hu, B., Guan, J., and Wang, S. (2021). Breaking lwc candidates: sestate and elephant in quantum setting. *Designs, Codes and Cryptography*, 89(7):1405–1432.

[103] Song, L., Tu, Y., Shi, D., and Hu, L. (2021). Security analysis of subterranean 2.0. *Designs, Codes and Cryptography*, 89(8):1875–1905.

[104] Sun, L., Wang, W., and Wang, M. Q. (2020). Milp-aided bit-based division property for primitives with non-bit-permutation linear layers. *IET Information Security*, 14(1):12–20.

[105] Tezcan, C. (2019). Distinguishers for reduced round ascon, drygascon, and shamash permutations. In *NIST Lightweight Cryptography Workshop*.

[106] Turan, M. S., McKay, K., Chang, D., Calik, C., Bassham, L., Kang, J., Kelsey, J., et al. (2021). Status report on the second round of the nist lightweight cryptography standardization process. *National Institute of Standards and Technology Internal Report*, 8369(10.6028).

[107] Turan, M. S., McKay, K. A., Çalik, Ç., Chang, D., Bassham, L., et al. (2019). Status report on the first round of the nist lightweight cryptography standardization process. *National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Rep.(NISTIR)*.

[108] Wu, H. (2011). The hash function jh. SHA-3 candidate submitted to NIST.

[109] Wu, H. and Huang, T. (2014). Jambu lightweight authenticated encryption mode and aes-jambu. *CAESAR competition proposal*.

[110] Zhang, G. and Liu, M. (2017). A distinguisher on present-like permutations with application to spongent. *Science China Information Sciences*, 60(7):1–13.

[111] Zhang, W., Ding, T., Zhou, C., and Ji, F. (2020). Security analysis of knot-aead and knot-hash. In *NIST Lightweight Cryptography Workshop*.

[112] Zhou, H., Li, Z., Dong, X., Jia, K., and Meier, W. (2020). Practical key-recovery attacks on round-reduced ketje jr, xoodoo-ae and xoodyak. *The Computer Journal*, 63(8):1231–1246.

[113] Zhou, H., Zong, R., Dong, X., Jia, K., and Meier, W. (2021). Interpolation attacks on round-reduced elephant, kravatte and xoofff. *The Computer Journal*, 64(4):628–638.