Security of COFB against Chosen Ciphertext Attacks

Mustafa Khairallah

Temasek Labs @ NTU
Nanyang Technological University, Singapore, Singapore
mustafa.khairallah@ntu.edu.sg

Abstract. COFB is a lightweight authenticated encryption (AE) mode based on block ciphers, proposed in CHES 2017 and is the basis for GIFT-COFB, a finalist in the NIST lightweight standardization project. It comes with provable security results that guarantee its security up to the birthday bound in the nonce-respecting model. However, the designers offer multiple versions of this analysis with different details and the implications of attacks against the scheme are not discussed deeply. In this article, we look at different possible attacks against COFB-like designs against both forgery and confidentiality. We show that the security for both forgery and confidentiality is bounded by the amount of forgery attempts. In particular, we show the existence of forgery and confidentiality attacks with success probability $q_f/2^{n/2}$, given q_f forgery attempts. In particular, we show that both forgery and confidentiality can be broken with $2^{n/2}$ attempts using only a single known-plaintext encryption query. While these attacks do not contradict the claims made by the GIFT-COFB designers, it shows its limitations in terms of the number of forgery attempts. It also shows that while GIFT-COFB generates a 128-bit tag it behaves in a very similar manner to an AE scheme with 64-bit tag. As an independent result, our analysis provides a contradiction to main in theorem of *Journal of Cryptology volume 33, pages 703–741 (2020)*, which is an includes an improved security proof of COFB compared to the CHES 2017 version. Finally, we discuss the term $nq_f/2^{n/2}$ that appears in the security proof of GIFT-COFB and CHES 2017, showing why this term is unlikely to be tight and it is likely that $q_f/2^{n/2}$ is sufficient. We emphasize that the results in this article do not threaten the security of GIFT-COFB in the scope of the NIST lightweight cryptography requirements or the claims made by the designers in the specification of the design.

Keywords: COFB · GIFT · Block Cipher · NIST · AEAD · Authenticated Encryption · Forgery.

1 Introduction

Over the past few years, the National Institute for Standardization and Technology (NIST), USA, have been running a lightweight cryptography standardization. The project called for Authenticated Encryption with Associated Data

(AEAD) algorithms where the amount of data that can be processed under one key is not less than $2^{50}-1$ bytes and the cryptanalytic attacks against the algorithms are of at least 2^{112} computational complexity [1]. The project received 57 proposals, 56 of them were selected as round 1 candidates, then narrowed down to 32 in round 2. In March 2021, 10 proposals were announced as finalists. Among these candidates, GIFT-COFB [2] is a block cipher-based proposal and will be the focus of this article.

The GIFT-COFB mode (depicted in Figure 1) is an instance of the COmbined FeedBack, which is an AEAD mode proposed in CHES 2017 [6] as a lightweight algorithm based on Block Ciphers (BC). It is is claimed to be secure up to offer integrity up to $2^{n/2}/n$ forgery attempts and confidentiality up to $2^{n/2}$ data complexity in the nonce-respecting mode. The integrity limit comes from a bound on the adversary's success probability on the form of $nq_f/2^{n/2}$ where q_f is the number of forgery attempts made by the attacker. Interestingly, this bound relies only on the number of forgery attempts and is independent of the amount of data encrypted by the algorithm or the computational abilities of the adversary. It is typical to see such similar terms when it comes to generic attacks based on the authentication tag size. In particular, an AEAD scheme that generates a τ-bit tag can be attacked by simply guessing the correct tag corresponding to a ciphertext. The attack's success probability relies on the number of forgery attempts $(q_f/2^\tau)$ and after 2^τ the adversary would have guessed the correct tag. However, GIFT-COFB has a tag size of n bits. so the appearance of $q_f/2^{n/2}$ raises some research questions. Most notably:

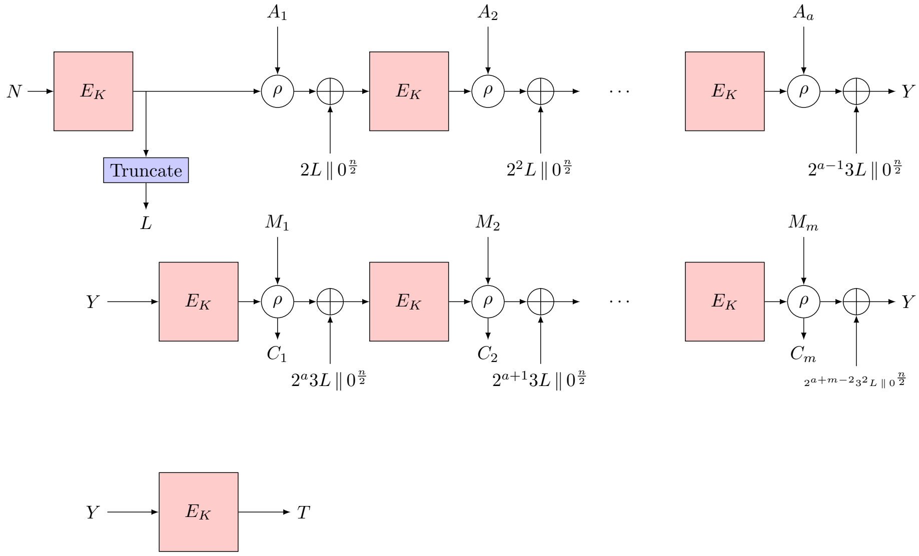
A_1
A_2
A_a
N
E_K
ρ
E_K
ρ
$\cdots$
E_K
ρ
Y
Truncate
$2L \parallel 0^{\frac{n}{2}}$
$2^2L \parallel 0^{\frac{n}{2}}$
$2^{a-1}3L \parallel 0^{\frac{n}{2}}$
L
M_1
M_2
M_m
Y
E_K
ρ
E_K
ρ
$\cdots$
E_K
ρ
Y
C_1
C_2
C_m
$2^a3L \parallel 0^{\frac{n}{2}}$
$2^{a+1}3L \parallel 0^{\frac{n}{2}}$
$2^{a+m-2}3^2L \parallel 0^{\frac{n}{2}}$
Y
E_K
T

Fig. 1. The COFB mode of operation.

1. *Can we break the GIFT-COFB algorithm with only $2^{n/2}/n$ forgery attempts and negligible (or 0) encryption queries?*
2. *Can we show that GIFT-COFB behaves as a scheme with a tag that is shorter than n bits, even when an n-bit tag is generated?*

These two questions are not answered by the security proofs of GIFT-COFB. Provable security is a critical tool in studying the security of new designs. It provides mathematical guarantees for their security. However, it does not always take the attacker's point of view and it often times does not consider what happens when the provable security bounds are reached. It may lead to conservative bounds that cannot be matched by attacks in practice. Besides, analyzing the schemes helps understand and verify the security proofs, understand the different assumptions that the designers may have used or implied, and identify errors, if any.

We also note that not all AEAD modes that are secure up to the Birthday Bound (upBB) suffer from such issues. For example, the GCM [10] is probably the most famous BC-based AEAD mode secure upBB but the security bound in on the form of $\sigma^2/2^n$, where σ is the total amount of data processed by the algorithm given a certain key. While $\sigma/2^{n/2} \approx \sigma^2/2^n$ when $\sigma \approx 2^{n/2}$, $\sigma/2^{n/2}$ is significantly larger than $\sigma^2/2^n$ when σ is small.

Given a BC with n-bit block and k-bit key, it expands the internal state by only $n/2$ bits compared to the state of the BC ($n+k$ bits). The designers assume that the BC is secure in the standard Pseudo-Random Permutation (PRP) model and that it behaves as a Pseudo-Random Function (PRF) up to the bound derived in the PRP-PRF switching lemma [4]. They assume that an adversary makes q_e encryption queries that involve σ_e blocks (σ_e invocations of the BC) and q_f forgery attempts that involve σ_f blocks. Given these assumptions, the authors presented a provable security bound that suggests that the success probability single-key attacks against COFB as an AEAD scheme, when the underlying cipher is replaced by a PRF, is bounded by

$$\Pr[\text{CHES 2017}] \leq \frac{4\sigma_e + 0.5nq_f}{2^{n/2}} + \frac{q_f + (q_e + \sigma_e + \sigma_f)\sigma_e}{2^n}.$$

Subsequently, an extended version of COFB has been published in the Journal of Cryptology (JoC) in 2020 [7]. The provable security bound in this version implies a different probability bound:

$$\Pr[\text{JoC 2020}] \leq \frac{4\sigma_e}{2^{n/2}} + \frac{q_f + (q_e + \sigma_e + 2\sigma_f)\sigma_f}{2^n}.$$

As part of the efforts surrounding the NIST lightweight cryptography project, the designers of GIFT-COFB [2] provided the following bound on the *Cryptology ePrint Archive report 2020/738*:

$$\Pr[\text{GIFT} - \text{COFB}] \leq \frac{1}{2^{n/2}} + \frac{(n+4)q_f}{2^{n/2+1}} + \frac{q_f + \sigma_e^2 + (q_e + \sigma_e + \sigma_f)\sigma_e}{2^n}.$$

While the three bounds share a lot of similarities, some of the strategies used in each security proof are different, which leads to the differences. More importantly, Pr[JoC 2020] is dominated by terms on the form of $\sigma^2/2^n$. If this bound is correct, then the question is if it is possible to adopt that bound for GIFT-COFB, improving its security claims.

Contributions In this article, we analyze the security of GIFT-COFB. We present the first IND-CCA attack with complexity $2^{n/2}$ and the first forgery attack with complexity $2^{n/2}$ to operate with a single encryption query. We show that the behaviour of GIFT-COFB is very close to that of an algorithm with half the tag size. We present an oversight in [7] and discuss the implications of our attack and open questions on the security of GIFT-COFB. Interestingly, the attack in section 5 represents a structural weakness in the mode itself, *i.e.* the attack works even if the underlying BC is replaced by an ideal PRF. The article is organized as follows: In Section 2 we give some needed definitions of authenticated encryption, the feedback function of GIFT-COFB. We also discuss the generic IND-CCA attack presented in [11]. We discuss existing forgery attacks against GIFT-COFB in Section 3 and a possible IND-CPA attack in Section 4. We present our main IND-CCA attack with discussions in Section 5. The IND-CCA and the observation on [7] have been communicated to and confirmed by the designers of the COFB mode.

2 Preliminaries

2.1 Nonce-based Authenticated Encryption (NAE)

Bellare and Namprempre [3] provide an in-depth discussion of the security notions and definitions of authenticated encryption. In this section, we focus on three main notions: IND-CPA, INT-CTXT and IND-CCA. An NAE scheme ensures both the confidentiality and authenticity of input data as long as the nonce, an auxiliary parameter associated with each query, is never repeated in two different encryption queries. More precisely, an NAE scheme Π consists of two algorithms:

- $(C,T) \leftarrow \Pi.\mathsf{Enc}(N,A,M)$: It takes as input a unique public nonce N, a public input string called associated data A and a private input string M. It returns a string C, such that $|C| = |M|$ and a string T such that $|T| = \tau$, where τ is a constant value and $|X|$ is the bit length of X.
- M or $\bot \leftarrow \Pi.\mathsf{Dec}(N,A,C,T)$: It takes as input N, A, C and T, and should return $\bot$ if the pair (C,T) is not generated by an earlier call to $\Pi.\mathsf{Enc}(N,A,M)$. Otherwise, it returns the corresponding M.

Notably, the definition prevents the repetition of N in two different encryption queries, but does not limit the repetition of the nonce in decryption queries or in different decryption and encryption queries. An adversary $\mathcal{A}$ has two goals. In order to break the confidentiality of the scheme it is sufficient to distinguish the outputs of $\mathsf{\Pi.Enc}$ from uniformly sampled random string from the space of all binary string of length $|M| + \tau$. Such adversary can make q_e queries to the oracle in question on the condition that the nonces used in these queries are pairwise distinct. Such goal is captured by the Chosen-Plaintext Attack (CPA) indistinguishability security notion (IND-CPA). Bounding $\mathcal{A}$'s advantage in this case leads to the confidentiality/privacy security bound. Besides, in order for $\mathcal{A}$ to break the integrity of the scheme, it is sufficient to make $\mathsf{\Pi.Dec}$ output a value other than $\perp$ for a pair (C, T) that has not been obtained from a legitimate call to $\mathsf{\Pi.Enc}$. To do so, $\mathcal{A}$ is allowed to make q_e to $\mathsf{\Pi.Enc}$ of size σ_e and q_f queries to $\mathsf{\Pi.Dec}$ of size σ_f, with the overall data size of the queries $\sigma = \sigma_e + \sigma_f$. This is captured by integrity of ciphertext security notion (INT-CTXT) and the setting is known as the Adaptive Forgery Attempts (AFA) setting. $\mathcal{A}$'s advantage in this setting leads to the integrity security bounds. Depending on the details of the scheme other parameters may be considered. For example, in block cipher-based NAE, the number of queries made by $\mathcal{A}$ to the block cipher is relevant to the security. However, we leave such details for simplicity as they are not related to the attacks in this article. The overall security of an AEAD algorithm is a combination of both notions. In other words, a secure AEAD scheme must be *at least* IND-CPA-secure and INT-CTXT-secure. Besides, Bellare and Namprempre [3] showed that if an AEAD mode is both IND-CPA-secure and INT-CTXT-secure, this implies it is also achieves indistinguishability against Chosen Ciphertext Attacks (IND-CCA).

2.2 Combined Feedback

The combined feedback function ρ used in the GIFT-COFB mode is a linear transformation from $2n$ bits to $2n$ bits. Given an output of the BC X_i and a plaintext block M_i, it outputs $C_i = X_i \oplus M_i$ and $S_i = M_i \oplus G(X_i)$, where G is a linear permutation over n bits. During decryption, it takes a ciphertext block C_i instead and outputs $M_i = X_i \oplus C_i$ and $S_i = X_i \oplus G(X_i) \oplus C_i$. In case an adversary has access to a known-plaintext-ciphertext pair (M_i, C_i), X_i and S_i can be found by $M_i \oplus C_i$ and $M_i \oplus G(M_i \oplus C_i)$ respectively. Before applying the next BC call, S_i is masked to Y_i using the mask L as shown in Figure 1. Note that for simplicity, we ignore N and A when we index X_i, S_i and Y_i, where $X_1 = E_K(Y_0)$ and $C_1 = X_1 \oplus M_1$.

2.3 Generic IND-CCA attack against AEAD modes.

An adversary can break the INT-CTXT security by simply guessing the tag T corresponding to (N, A, C). Such adversary has an advantage of $1/2^\tau$. After $q_f = O(2^\tau)$ forgery attempts, there is a high probability of success. The probability of success of this type of attacks is upper bounded by $q_f/2^\tau$. Notably, this attack strategy requires no calls to $\mathsf{\Pi.Enc}$, *i.e.*, $q_e = 0$. In a discussion on the NIST lightweight cryptography forum [11], Alexandre Mège proposed an IND-CCA attack against NAE with tag size τ. The attack works against a BC-based AEAD mode with n-bit blocks as follows:

1. $\mathcal{A}$ selects a triplet (N, A, C) that has never seen before, where C is C is longer than n bits.
2. $\mathcal{A}$ selects a potential tag T^* and asks for the decryption of (N, A, C, T^*).
3. If the decryption is successful, $\mathcal{A}$ receives M. Otherwise, the adversary tries again.
4. When M is received, $\mathcal{A}$ asks for the encryption of $(N, A, M^{'})$, such that M and $M^{'}$ share the first n bits.
5. When $C^{'}$ and $T^{'}$ are received, $\mathcal{A}$ check whether $C^{'}$ and C share the first n bits. Ideally, this should only happen with probability 2^{-n}, but it happens with probability 1 when an NAE algorithm is used.

The attack requires 2^{τ} forgery attempts and no encryption queries besides the challenge query. While the attack is natural, due to the relation between INT-CTXT and IND-CCA, it is only worrying when the tag length τ is small. In general, it is not always possible or clear how to convert a forgery attack into an IND-CCA attack.

3 Forgery Attacks against GIFT-COFB

Two forgery attacks have been presented by Khairallah against GIFT-COFB in [9] and [8]. The attacks take advantage of special relations on the mask L used in the mode (see Figure 1). In [9], the attack works as follows:

1. $\mathcal{A}$ asks for the encryption of (N, A, M), where the length of M is at least $2n$. In particular, M consists of m blocks $M_1 \| M_2 \| \cdots \| M_m$.
2. $\mathcal{A}$ assumes $L = 0$. Note that L is secret and never revealed to the adversary.
3. Using the linearity of the feedback function ρ, $\mathcal{A}$ can find a block $C_x = M_2 \oplus M_1 \oplus C_1 \oplus G(M_1 \oplus C_1) \oplus G(M_2 \oplus C_2)$.
4. $\mathcal{A}$ asks for the decryption of $(N, A, C^{'}, T)$ where $C^{'} = C_x \| C3 \| \cdots \| C_m$.
5. If the forgery is unsuccessful, $\mathcal{A}$ repeats with a different N.

Since the attack relies on $L = 0$, and L is randomly generated, it has a success probability of $2^{-n/2}$. In other words, it needs $2^{n/2}$ short encryptions and $2^{n/2}$ short decryptions to have success probability close to 1. In [8], the attack works as follows:

1. $\mathcal{A}$ asks $2^{n/4}$ encryptions of (N^i, A, M^i), where the length of M is at least n. In particular, M^i consists of m blocks $M_1^i \| M_2^i \| \cdots \| M_m^i$.
2. $\mathcal{A}$ picks two outputs (N^i, A, C^i, T^i) and (N^j, A, C^j, T^j) randomly from the set of all received encryption outputs.
3. Using the linearity of the feedback function ρ, $\mathcal{A}$ can find a block $C_x = M_1^i \oplus C_1^i \oplus G(M_1^i \oplus C_1^i) \oplus M_1^j \oplus G(M_1^j \oplus C_1^j)$.
4. $\mathcal{A}$asks for the decryption of $(N^i, A^i, C^{'}, T^j)$ where $C^{'} = C_x \| C2 \| \cdots \| C_m$.
5. If the forgery is unsuccessful, $\mathcal{A}$ two different outputs (N^i, A, C^i, T^i) and (N^j, A, C^j, T^j) randomly from the set of all received encryption outputs.

The attack relies on a collision $L^i = L^j$. Since L is an $n/2$-bit random variable, the probability of such collision after $2^{n/4}$ encryptions is close to 1. Assuming such collision exists in the encryption queries, finding it requires sampling the correct pair, which has a probability of $2^{-n/2}$. The attack needs $2^{n/4}$ short encryptions and $2^{n/2}$ short decryptions to have success probability close to 1. Both attacks require a non-negligible amount of encryption queries. Moreover, they cannot be directly extended to IND-CCA attacks as both attacks rely on asking for an encryption with the target forgery nonce before performing forgery.

4 IND-CPA Attack Against GIFT-COFB

It is easy to see that an IND-CPA attack exists with probability $\sigma_e^2/2^n$. For example, in an IND-CPA attack, the adversary observes the outputs of all BC calls corresponding to ciphertext blocks, where $X_i = M_i \oplus C_i$. Let $\mathbf{X} = \{X_i^j\}$, such that $X_i^j = M_i \oplus C_i$. After $2^{n/2}$ CPA plaintext blocks, the probability of a collision in $\mathbf{X}$ would be close to 1. If the ciphertext blocks are uniformly distributed over the set of all possible n-bit strings and the collision is on the form $X_i^j = X_{i'}^{j'}$, then

$$\Pr[X_{i-1}^j = X_{i'-1}^{j'}] = 2^{-n}.$$

However, in the case of GIFT-COFB, a collision $X_i^j = X_{i'}^{j'}$ implies a collision $Y_{i-1}^j = Y_{i'-1}^{j'}$. While the adversary does not observe the full input of the BC calls, due to masking, the unmasked half of the block is still observable. Hence, in the case of GIFT-COFB,

$$\Pr[X_{i-1}^j = X_{i'-1}^{j'}] = 2^{-n/2}.$$

Hence, an attack can work as follows:

1. $\mathcal{A}$ asks for a set of encryption queries where the total size of the ciphertexts is at least $2^{n/2}$ blocks.
2. $\mathcal{A}$ then generates the set $\mathbf{X}$ and identifies a collision.
3. One a collision $X_i^j = X_{i'}^{j'}$ is found, the adversary checks if the unmasked halves of X_{i-1}^j and $X_{i'-1}^{j'}$ are identical, and decides the ciphertext is generated using GIFT-COFB if they are.

The last step happens with probability $2^{-n/2}$ if the ciphertexts are uniformly distributed and with probability 1 in the case of GIFT-COFB. The attack has success probability of

$$\frac{\sigma_e^2}{2^n} - \frac{\sigma_e^2}{2^{3n/2}}$$

which is close to 1 at $\sigma_e = 2^{n/2}$. Note that here we abuse the notation a little, as σ_e should also include the nonces and associated data blocks, but the adversary can limit their effect to only 2 blocks per query, and use long queries, making the overall complexity only slightly larger than $2^{n/2}$. This attack, however, is not relevant to the bounds discussed in Section 1 as it relies on the fact that the underlying cipher may only behave as a PRF up to the birthday bound, which we excluded from the bounds we present.

5 New IND-CCA Attack Against GIFT-COFB

All the previous INT-CTXT attacks in Section 3 rely on the fact that the chosen plaintext queries give the adversary a somewhat restricted access to the underlying tweakable BC defined by $Y = E_K(X \oplus \text{mask}(L, i))$ where i is the index of the BC call. Hence, an adversary would need to satisfy a condition on the mask L in order to be able to predict when the same input-output values can be reused. The crucial idea was to either find $L = 0$ or to find $L_1 = L_2$. The first condition requires $O(2^{n/2})$ encryptions and the second requires $O(2^{n/4})$ encryptions. We observe that the encryption complexity of satisfying the required condition on L translates to encryption complexity, while the decryption complexity is almost fixed at $O(2^{n/2})$. Besides, these attacks cannot be used to directly launch and IND-CCA attack as they do not leak any information about fresh nonces that have not been used in previous encryption queries.

In order to find an INT-CTXT or an IND-CCA attack with negligible encryption complexity, we start by asking two questions:

1. Can we find an attack that does not requires any special condition on the value (or values) of L?
2. What can we achieve if we completely knew what is the value of L?

We observe that if L is leaked, then the adversary gets full access to any CPA block corresponding to the query that uses such L. In other words, not only can the adversary observe the outputs of the BC calls, but also the inputs. Using such information proves to be critical in constructing forgeries using nonces that may have never appeared in any encryption queries. Assume the adversary know that $V = E_K(P)$. The adversary can use P as a nonce and would know that the input the initial state of the algorithm before absorbing A or M, as well as the initial value of the mask L. An attacker can choose A and C to be one block each, such that if $L^* = \mathit{truncate}(V)$ is the leftmost $n/2$ bits of V, then

$$P = 3L^* \| 0^{n/2} \oplus G(V) \oplus A_1$$

and

$$P = 3^2 L^* \| 0^{n/2} \oplus G(V) \oplus V \oplus C_1.$$

The decryption query (N^*, A^*, C^*, T^*), where $N^* = P$, $A^* = A_1$, $C^* = C_1$, $T^* = V$, will succeed with probability 1, outputting M_1. Once M_1 is retrieved, the adversary can ask for the encryption query $(P, A_1, M^{'})$, where $M^{'} = M_1 \| M_2$ and M_2 can take any value. If the ciphertexts are uniformly distributed, then the probability of the first n-bit block to be equal to C_1 is 2^{-n}, while the probability is 1 in the case of GIFT-COFB. The challenge with this attack is that the adversary does not have such access to the inputs of the BC calls during encryption. The adversary can go around this by guessing the value of L. The full attack operates as follows:

1. $\mathcal{A}$ asks for the encryption query (N, A, M) where M consists of one block and for simplicity $A = \epsilon$ (empty string).
2. $\mathcal{A}$ guesses the value 3^2L, and stores $V = T$, $P = G(M_1 \oplus C_1) \oplus M_1 \oplus 3^2L \| 0^{n/2}$ and $L^* = truncate(V)$
3. $\mathcal{A}$ assigns $N^* = P$, $A^* = P \oplus 3L^* \| 0^{n/2} \oplus G(V)$, $C^* = P \oplus 3^2L^* \| 0^{n/2} \oplus G(V) \oplus V$ and $T^* = V$.
4. $\mathcal{A}$ asks for the decryption of (N^*, A^*, C^*, T^*) and if the forgery is unsuccessful, repeats from step 2.

It is easy to see that if the guess of 3^2L is correct, then the attack succeeds. The critical observation is that if the guess is incorrect, step 1 does not need to be repeated. The adversary can simply guess a different value. The probability of success of this forgery attack is $q_f/2^{n/2}$. After $2^{n/2}$ guesses the adversary is bound to have guessed the correct value. Consequently, if $N^* = P$ has never been used in any encryption query, the adversary can perform the following attack:

1. $\mathcal{A}$ asks for the encryption of $(N^*, A^*, M^{'})$ where $M^{'} = M^* \| M_2$ and M_2 can take any value.
2. If the first n bits of the ciphertext are identical to the C^* then $\mathcal{A}$ decides the ciphertext was generated using GIFT-COFB.

The probability that $N^* = P$ is high and in some scenarios can be forced to 1 by the adversary, while the overall complexity of the attack is a single encryption query and $2^{n/2}$ forgery attempts.

5.1 Analysis and Discussions of the IND-CCA Attack

Effective Tag Size of GIFT-COFB The attack presented in this section holds some resemblance to the attack described in Section 2.3 [11]. They mainly differ in two points:

1. The attack in [11] targets algorithms with short tags. It works with $2^{n/2}$ forgery attempts against algorithms with $n/2$-bit tags. Our attack complexity is a function of the mask size rather than the tag size. Given an $n/2$-bit mask, the attack works with $2^{n/2}$ forgery attempts, even if the tag size is larger than $n/2$ bits.
2. The attack in [11] requires only decryption queries, and no encryption queries except the challenge query. Our attack requires one encryption query at the beginning. However, this encryption query may consist of one block and the plaintext can only be known, not necessarily chosen. In practice, this limitation is very mild and the adversary can achieve it in many cases.

Given these two differences, it seems that the tag size of GIFT-COFB offers little immunity compared to algorithms with half the tag size, and by keeping the tag size n instead of truncating it to $n/2$ (or a value in between) seems to offer very minimal security advantage.

Forgery with no encryption queries It can be shown that it is impossible to perform successful forgery with non-negligible probability with no encryption queries at all, while keeping the forgery attempts close to $2^{n/2}$. Assume a forgery attempt uses the tag T^* and $\mathbf{X}$ is defined as in Section 4. Since the underlying cipher is a permutation, if $T^* \notin \mathbf{X}$, then the internal state during decryption corresponding Y_m^* (where $T^* = E_K(Y_m^*)$ has never appeared in any block cipher call during encryption. Assuming the underlying cipher is a PRP, then the probability that the adversary can guess T^* is bounded by

$$\frac{1}{2^n - \sigma_e}$$

By setting $\sigma_e = 0$, the probability is simply 2^{-n}. However, as shown this limitation is simply bypassed by a single known-plaintext query.

Relation to existing security bounds As discussed in Section 1, there are three security proofs in the literature that cover GIFT-COFB. The attack proposed in this paper do not contradict the proofs in [2] and [6], whose bound is dominated by the term $nq_f/2^{n/2}$. However, it does contradict Theorem 2 in [7]. This is due to an error in calculating the probability of a collision between the internal state values in encryption and decryption queries. Let Y_i^j be the input to the BC at block i in the encryption query j and $Y_{i'}^{j'}$ be the input to the BC at block i' in the decryption query j'. The proof of Theorem 2 of [7] bounds the probability of a collision on the form $Y_i^j = Y_{i'}^{j'}$ by

$$\Pr[\exists i, j, i', j', s.t. Y_i^j = Y_{i'}^{j'}] \leq \frac{(q_e + \sigma_e)\sigma_f}{2^n}$$

However, this assumes that the internal states are completely random. In reality, the adversary has almost full control over $n/2$ bits of the state. For example, the adversary can force a collision on half the state by forcing half the input of the BC during a decryption query to a value that has been observed during encryption. However, once the adversary makes such decision, the probability of a full collision becomes $2^{-n/2}$. The adversary can keep changing the masked half of the state during decryption until the guess is correct. Hence, the probability is bounded by

$$\Pr[\exists i, j, i', j', s.t. Y_i^j = Y_{i'}^{j'}] \leq \frac{q_f}{2^{n/2}}.$$

This observation have been communicated to the authors of [7] and has been verified by them.

Potential Remedy The IND-CCA attack presented requires the ability to predict both the internal state and mask corresponding to a nonce N^*, where a successful guess of an $n/2$ bit value leaks both. Currently, one PRF is used to generate both the mask and the initial internal state. In order prevent this, we can use two different PRF constructions to generate each value. For example, we can use $L = \text{truncate}(E_K(N))$, which is the same as the current situation, while the initial state (the state XORed with the first associated data block) be $E_K(E_K(N))$, *i.e.* adding an extra BC call after the mask generation. However, while this may prevent the presented IND-CCA attack, it will not affect the security bounds and may introduce additional problems. Hence, this issue requires an independent study, outside the scope of this article.

Generalization to many forgeries One of the goals of this article is challenge the limits of the security of GIFT-COFB. However, the attack can be enhanced to allow many forgeries from a single plaintext query. If the encryption query (step 1) includes m plaintext blocks, then after successfully guessing L, *i.e.*, after the first successful forgery, the attacker has m choices for N^* and can forge many blocks of A^* and C^*. Even with the single block encryption query, the attacker has only a single choice for N^*, but can use it with different lengths of A and M. In that ascpect, GIFT-COFB is even worse than a generic scheme with $n/2$-bit tag, as such scheme allows 1 forgery per $2^{n/2}$ attempts, while GIFT-COFB allows many forgeries once the $2^{n/2}$-attempt threshold is met.

Open questions in terms of the tightness of GIFT-COFB security bounds. Focusing on [2] and [6], we note that while our attack shows the tightness of the bound on the form $q_f/2^{n/2}$, the bounds on the form $\sigma_e/2^{n/2}$ and $nq_f/2^n$. It is likely that the latter bound is an artefact of the proof methodology and can be eliminated, as we observe that whatever the attack strategy is the adversary needs to guess a random $n/2$-bit value. The questions of whether we can find an IND-CPA attack whose probability is bounded by $\sigma_e/2^{n/2}$ or whether we can find a forgery attack with small σ_f remain unsolved.

Applicability to other COFB-designs HyENA [5] was a round 2 candidate of the NIST lightweight cryptography standardization project and it holds a lot of resemblance to the COFB mode. The main differences are:

1. It uses a different feedback function known as hybrid feedback.
2. Before calling the BC for the last time the internal state is split in two halves and swapped.
3. The nonce is limited to r bits concatenated with an $(n-r)$-bit constant during the first call to the BC.

The first two differences do not affect the applicability of the proposed attack. The third difference makes the attack function of finding an internal state on the form $N \| B$ where B is the $(n-r)$-bit constant needed in the first call. The probability of this event is 2^{r-n}. The adversary can increase the length of the first encryption query to 2^{n-r} blocks, such that when L is guessed correctly, there is a high probability that one of the blocks can be used during initialization of the forgery attempt, which increases the encryption complexity from 1 to 2^{n-r}. However, since r is typically large, the complexity is still acceptable. For example, in the NIST specification of HyENA, $r = 96$ and the attack would need one encryption query of length 2^{32} and 2^{64} short forgery attempts.

6 Conclusions

In this article, we have analyzed the GIFT-COFB algorithm showing that it is secure against IND-CCA adversaries at most up to $2^{n/2}$ forgery attempts. We presented a new forgery attack and the first IND-CCA attack against GIFT-COFB with negligible encryption complexity and $2^{n/2}$ forgery attempts. We show that GIFT-COFB behaves in a similar manner to a generic AEAD scheme with $n/2$-bit tag, and in some scenarios even worse. As a byproduct, we have identified an oversight in [7]. However, we emphasize that the attacks do not threaten [2], [6] or the security of GIFT-COFB according to the requirements of the NIST lightweight cryptography standardization project.

Acknowledgment

We would like to thank Thomas Peyrin from the GIFT-COFB team for discussions on the design and the designers of the COFB mode: Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, and Mridul Nandi for providing feedback on the main result of this article and on [7].

References

1. Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process (2018), https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf
2. Banik, S., Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT-COFB. Cryptology ePrint Archive, Report 2020/738 (2020), https://eprint.iacr.org/2020/738
3. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) Advances in Cryptology — ASIACRYPT 2000. pp. 531–545. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
4. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) Advances in Cryptology - EUROCRYPT 2006. pp. 409–426. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
5. Chakraborti, A., Datta, N., Jha, A., Nandi, M.: HyENA. NIST Lightweight Cryptography Project (2019), https://csrc.nist.gov/Projects/Lightweight-Cryptography/Round-1-Candidates

6. Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M.: Blockcipher-based authenticated encryption: How small can we go? In: International Conference on Cryptographic Hardware and Embedded Systems. pp. 277–298. Springer (2017), https://link.springer.com/chapter/10.1007/978-3-319-66787-4_14
7. Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M.: Blockcipher-based authenticated encryption: How small can we go? In: Journal of Cryptology. pp. 703–741. Springer (2020), https://link.springer.com/article/10.1007/s00145-019-09325-z
8. Khairallah, M.: Observations on the tightness of the security bounds of gift-cofb and hyena. Cryptology ePrint Archive, Report 2020/1463 (2020), https://eprint.iacr.org/2020/1463
9. Khairallah, M.: Weak keys in the rekeying paradigm: Application to comet and mixfeed. IACR Transactions on Symmetric Cryptology **2019**(4), 272–289 (Jan 2020). https://doi.org/10.13154/tosc.v2019.i4.272-289, https://tosc.iacr.org/index.php/ToSC/article/view/8465
10. McGrew, D., Viega, J.: The galois/counter mode of operation (gcm). submission to NIST Modes of Operation Process **20**, 0278–0070 (2004)
11. Mège, A.: (Nov 2019), https://groups.google.com/a/list.nist.gov/g/lwc-forum/c/2a0H-HQHgqU/m/EtjdRFSmBQAJ