IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 0, No. 0, pp. 0–0. DOI:10.13154/tosc.v0.i0.0-0

Linear Cryptanalyses of Three AEADs with GIFT-128 as Underlying Primitives

Ling Sun[1,2], Wei Wang[1,2] and Meiqin Wang[1,2](✉)

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
[2] School of Cyber Science and Technology, Shandong University, Qingdao, China
lingsun@sdu.edu.cn, weiwangsdu@sdu.edu.cn, mqwang@sdu.edu.cn

Abstract. This paper considers the linear cryptanalyses of Authenticated Encryptions with Associated Data (AEADs) GIFT-COFB, SUNDAE-GIFT, and HyENA. All of these proposals take GIFT-128 as underlying primitives. The automatic search with the Boolean satisfiability problem (SAT) method is implemented to search for linear approximations that match the attack settings concerning these primitives. With the newly identified approximations, we launch key-recovery attacks on GIFT-COFB, SUNDAE-GIFT, and HyENA when the underlying primitives are replaced with 16-round, 17-round, and 16-round versions of GIFT-128. The resistance of GIFT-128 against linear cryptanalysis is also evaluated. We present a 24-round key-recovery attack on GIFT-128 with a newly obtained 19-round linear approximation. We note that the attack results in this paper are far from threatening the security of GIFT-COFB, SUNDAE-GIFT, HyENA, and GIFT-128.

Keywords: Linear cryptanalysis · GIFT-128 · GIFT-COFB · SUNDAE-GIFT · HyENA

1 Introduction

Linear cryptanalysis [Mat93] is one of the most fundamental methods to evaluate the security of symmetric-key primitives. This method pays attention to the linear relationship among the input, the key, and the output of the objective function. Compared to differential cryptanalysis [BS90], we find that linear cryptanalysis is more suitable to the cryptanalysis of the Authenticated Encryption with Associated Data (AEAD) since it works under the known-plaintext attack setting.

This paper manages to evaluate the security of GIFT-COFB, SUNDAE-GIFT, and HyENA with the linear method. The targets are set as the encryption functions in the ciphertext generating phases. According to the individual features of these proposals, we implement automatic search with the Boolean satisfiability problem (SAT) method in [SWW18] to search for linear approximations satisfying specific restrictions. We propose three 10-round linear approximations that fit the attack settings of these primitives. With the newly obtained distinguisher, we realise linear key-recovery attacks on GIFT-COFB, SUNDAE-GIFT, and HyENA when the underlying primitives are replaced with round-reduced versions of GIFT-128. An overview of our results is shown in Table 1. Also, the resistance of GIFT-128 against linear cryptanalysis is checked. A 24-round key-recovery attack is proposed with the newly identified 19-round linear approximation. Please find in Table 1 a sketch of the 24-round attack.

Organisation. In Sect. 2, we introduce the primitives analysed in the paper. Also, the automatic technique for the search of linear approximations is briefly recalled. The main

Table 1: Summary of the cryptanalytic results on the three AEADs and GIFT-128

Algorithm	Attack	Rounds	Time	Data	Memory	Success probability	Ref.
GIFT-COFB	Linear	15	$2^{90.70}$	$2^{62.00}$	2^{96}	-	[ZDC+21]
		16	$\mathbf{2^{122.80}}$	$\mathbf{2^{62.10}}$	$\mathbf{2^{47}}$	**80.01%**	**Sect. 4.1**
SUNDAE-GIFT	Linear	16	$2^{91.20}$	$2^{60.00}$	2^{96}	-	[ZDC+21]
		17	$\mathbf{2^{123.38}}$	$\mathbf{2^{61.51}}$	$\mathbf{2^{49}}$	**80.01%**	**Sect. 4.2**
HyENA	Linear	**16**	$\mathbf{2^{122.00}}$	$\mathbf{2^{61.51}}$	$\mathbf{2^{52}}$	**80.01%**	**Sect. 4.3**
GIFT-128	Differential	23	$2^{120.00}$	$2^{120.00}$	2^{86}	-	[ZDY19]
		26	$2^{124.42}$	$2^{124.42}$	2^{109}	-	[LWZZ19]
		26	$2^{123.25}$	$2^{123.25}$	2^{109}	-	[JZZD20]
		27	$2^{124.83}$	$2^{123.53}$	2^{80}	-	[ZDC+21]
	Linear	22	$2^{117.00}$	$2^{117.00}$	2^{78}	-	[ZDC+21]
		24	$\mathbf{2^{124.45}}$	$\mathbf{2^{122.55}}$	$\mathbf{2^{105}}$	**80.01%**	**Sect. 5.2**

method to estimate the complexity of the linear attack and the linear approximations exploited in the key-recovery procedure are presented in Sect. 3. In Sect. 4, we launch linear attacks for three AEADs with GIFT-128 as underlying primitives. The security of GIFT-128 regarding linear cryptanalysis is also considered, and a 24-round linear attack is proposed in Sect. 5. Sect. 6 concludes the paper.

2 Preliminaries

In this section, we first introduce GIFT-128, which works as the building blocks for the three subsequent AEAD algorithms. Then, the overall structures of GIFT-COFB, SUNDAE-GIFT, and HyENA are presented. After that, we briefly recall the automatic method for the search of linear approximations.

2.1 Description of GIFT-128

GIFT-128 is one version of GIFT [BPP+17] that exploits the Substitution-Permutation Network (SPN). GIFT-128 is a 40-round cipher with 128-bit inputs. The plaintext is initialised as $b_0 b_1 \cdots b_{127}$, and b_0 stands for the most significant bit. The cipher also receives a 128-bit key $K = k_0 \| k_1 \| \cdots \| k_7$, where k_i's are 16-bit words. Each round of GIFT-128 is composed of three steps: SubCells, PermBits, and AddRoundKey.

SubCells GIFT-128 employs the same invertible 4-bit S-box GS.

x	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
$GS(x)$	0x1	0xa	0x4	0xc	0x6	0xf	0x3	0x9	0x2	0xd	0xb	0x7	0x5	0x0	0x8	0xe

The S-box is applied to every nibble of the inner state.

PermBits The bit permutation maps the bit from bit position i of the cipher state to bit position $P(i)$, i.e.,

$$b_{P(i)} \leftarrow b_i, i \in \{0, 1, \ldots, 127\}.$$

The value of $P(i)$ can be calculated as

$$127 - \left\{ 4 \left\lfloor \frac{127-i}{16} \right\rfloor + 32 \left\lfloor 3 \left\lfloor \frac{(127-i) \bmod 16}{4} \right\rfloor + (127-i) \bmod 16 \right\rfloor + (127-i) \bmod 4 \right\} \bmod 128.$$

AddRoundKey This step consists of adding the round key and the round constant. After extracting a 64-bit round key RK from the key state, we partition it into two 32-bit words as $RK = U\|V = u_0u_1\cdots u_{31}\|v_0v_1\cdots v_{31}$. U and V are XORed with the cipher state as follows

$$b_{4\cdot i+1} \leftarrow b_{4\cdot i+1} \oplus u_i,\ b_{4\cdot i+2} \leftarrow b_{4\cdot i+2} \oplus v_i,\ i \in \{0, 1, \ldots, 31\}.$$

The adding round constant operation is not introduced here as it does not affect the validity of the attacks in this paper.

To minimise the hardware area and maintain the software friendly simultaneously, the designers only implement state rotation and bit rotation operations in the key schedule.

Key schedule Note that the round key should be extracted before the update of the key state. In each round, the 64-bit round key $RK = U\|V$ is firstly assigned as

$$U \leftarrow k_2\|k_3,\ V \leftarrow k_6\|k_7.$$

Then, the key state is updated as follows,

$$k_0\|k_1\|\cdots\|k_7 \leftarrow (k_6 \ggg 2)\|(k_7 \ggg 12)\|k_0\|\cdots\|k_4\|k_5.$$

2.2 Three AEADs with GIFT-128 as Underlying Primitives

This paper investigates the linear attacks for GIFT-COFB, SUNDAE-GIFT, and HyENA. In this subsection, we recall the overall structures of these primitives and refer readers to [BCI+20, BBP+19, CDJN19] for more details.

2.2.1 GIFT-COFB

GIFT-COFB [BCI+20] is an Authenticated Encryption with Associated Data (AEAD) that instantiates the COmbined FeedBack (COFB) mode [CIMN17] with GIFT-128. As in Figure 1, the encryption algorithm takes the following data as inputs:

- an encryption key $K \in \{0,1\}^{128}$;
- a nonce $N \in \{0,1\}^{128}$;
- associated data and message $A, M \in \{0,1\}^*$.

The algorithm outputs the following data:

- a ciphertext $C \in \{0,1\}^{|M|}$;
- a tag $T \in \{0,1\}^{128}$.

The E_K functions in Figure 1 are referred to as the cipher GIFT-128. The feedback function $G : \{0,1\}^{128} \mapsto \{0,1\}^{128}$ is defined as $G(Y_0\|Y_1) = Y_1\|(Y_0 \lll 1)$, where $Y_0, Y_1 \in \{0,1\}^{64}$. The 64-bit value L depends on the values of N and K and thus is unknown. Furthermore, L is applied to generate masks for all the subsequent E_K functions. The designers claim that GIFT-COFB achieves 64 bits IND-CPA security under the nonce respecting scenario.

2.2.2 SUNDAE-GIFT

SUNDAE-GIFT [BBP+19] is a family of AEAD schemes that exploit the AEAD scheme SUNDAE [BBLT18] with GIFT-128 as the underlying block cipher. The encryption algorithm, which is illustrated in Figure 2, takes as input a key $K \in \{0,1\}^{128}$, an associated data $A \in \{0,1\}^*$, and a message $M \in \{0,1\}^*$. For variants accepting a fixed-length nonce N, the nonce is prepended to and regarded as a part of the associated data A. The output of the encryption is a ciphertext $C \in \{0,1\}^{|M|}$ and a tag T. The operation '$\times$' in Figure 2 is the multiplication by 2 or 4, which depends on the length of the last blocks of A and M. For more information about the primitive, please refer to [BBP+19].

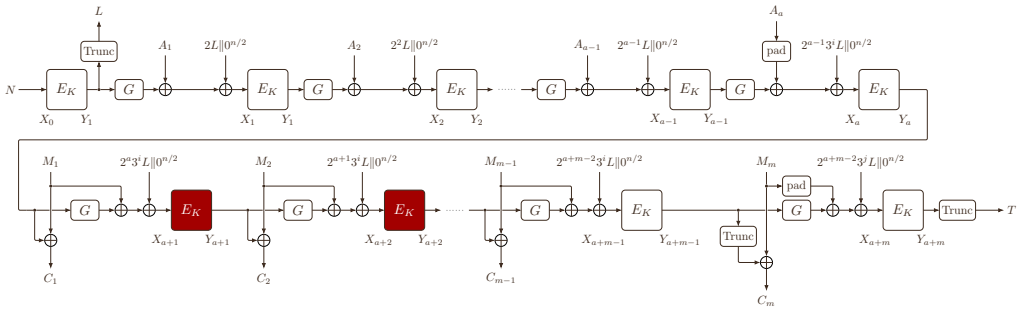


Figure 1: Encryption of GIFT-COFB.

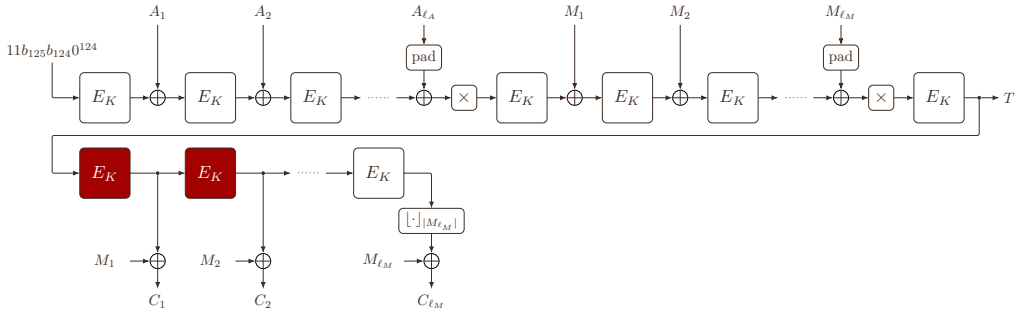


Figure 2: Encryption of `SUNDAE-GIFT`.

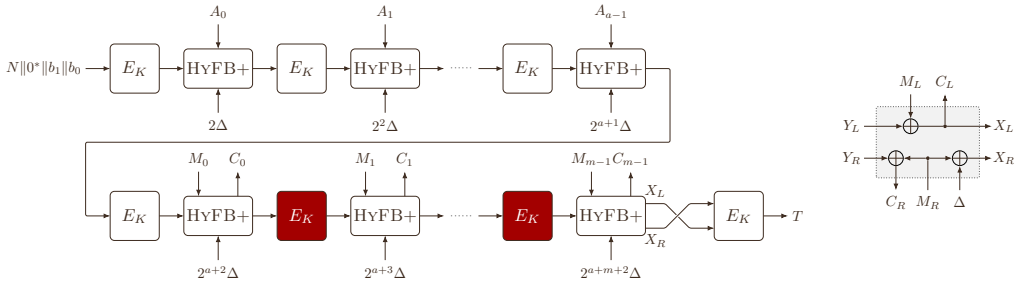
2.2.3 HyENA

The word HyENA in [CDJN19] has two meanings. For the one thing, it stands for the HYbrid feedback-based ENcryption with Authentication mode of operation that provides Nonce-based Authenticated Encryption with Associated Data (NAEAD) functionality. It also indicates the instantiation of the mode of operation mentioned above with the cipher `GIFT-128`. In this paper, we take the second meaning when we refer to HyENA.

The encryption of HyENA is shown in Figure 3, which takes an encryption key $K \in \{0,1\}^{128}$, a nonce $N \in \{0,1\}^{96}$, an associated data $A \in \{0,1\}^*$, and a message $M \in \{0,1\}^*$ as the input and returns a ciphertext $C \in \{0,1\}^{|M|}$ and a tag $T \in \{0,1\}^{128}$. Similar to the case in GIFT-COFB, HyENA also creates a 64-bit unknown value Δ before the associated data processing phase. This value assists in masking half of the input state for all of the following E_K functions. Under the nonce respecting scenario, the designers claim that for a valid attack on HyENA, the data requirement should be less than 2^{64}, and the time complexity is bounded by 2^{128}.

2.3 Automatic Method for the Search of Linear Approximations

The most fundamental step to launch a linear attack is to find a linear distinguisher. This paper utilises the automatic tool with the Boolean satisfiability problem (SAT) method in [SWW18] to search for linear approximations of `GIFT-128`. In different settings, we start with searching the linear trail with the absolute value of the correlation being no less than the predetermined value c. Then, we select the distinguisher that matches the attack setting of the primitive under consideration. After fixing the input and output masks of the linear approximation, we manage to discover all linear trails, whose absolute values of correlations are no less than $c \cdot 2^{-20}$, in this approximation.



(a) Encryption of HYENA. (b) HYFB+ module.

Figure 3: Overall structure of HYENA.

To enable the SAT problem to realise the search of linear trails, we should use Boolean formulas to represent the linear property of the cipher. Benefiting from the simple but elegant structure of GIFT-128, describing the propagation of the linear mask inside the cipher boils down to tracing the propagation of the linear mask across the S-box. In other words, we should create a set of Boolean expressions that incorporates the full information of the Linear Approximation Table (LAT) of the S-box.

Denote $\boldsymbol{x} \in \mathbb{F}_2^4$ and $\boldsymbol{y} \in \mathbb{F}_2^4$ the input and output masks of the S-box. Since the absolute values of the entries in the LAT belong to the set $\{0, 2, 4, 8\}$, the absolute value of the correlation for the possible propagation only has three possibilities, i.e., 2^{-2}, 2^{-1}, and 1. To encode the absolute value of the correlation $|cor|$, we employ two Boolean variables ε_0 and ε_1 so that $\varepsilon_0 + \varepsilon_1$ equals the opposite number of the binary logarithm of $|cor|$. Specifically, if $|cor|$ equals 2^{-2}, $\varepsilon_0\|\varepsilon_1$ is set as $(1, 1)$; if $|cor|$ equals 2^{-1}, $\varepsilon_0\|\varepsilon_1$ is set as $(0, 1)$; if the propagation is deterministic, $\varepsilon_0\|\varepsilon_1$ is set as $(0, 0)$. Next, following the method in [SWW18], we define a 10-bit Boolean function $f(\boldsymbol{x}\|\boldsymbol{y}\|\varepsilon_0\|\varepsilon_1)$ as

$$f(\boldsymbol{x}\|\boldsymbol{y}\|\varepsilon_0\|\varepsilon_1) = \begin{cases} 1, & \text{if } \boldsymbol{x} \to \boldsymbol{y} \text{ is a possible propagation with } |cor| = 2^{-(\varepsilon_0+\varepsilon_1)} \\ 0, & \text{otherwise} \end{cases}.$$

After simplifying the expression of f with the off-the-shelf software Logic Friday[1], we get a set of Boolean formulas that precisely depicts the relation among $\boldsymbol{x}$, $\boldsymbol{y}$, ε_0, and ε_1.

Aside from tracking the propagation of linear mask inside the cipher, the SAT problem should clarify the correlation of the targeted linear trail. Suppose we intend to search for the r-round trail with the absolute value of the correlation being no less than $c = 2^{-\xi}$, where ξ is a positive integer. Let $\varepsilon_k^{(i,j)}$ be the auxiliary variable regarding the j-th S-box in the i-th round, where $0 \leqslant i \leqslant r-1$, $0 \leqslant j \leqslant 31$, $0 \leqslant k \leqslant 1$. The valid linear trail should satisfy the condition

$$\sum_{i=0}^{r-1}\sum_{j=0}^{31}\sum_{k=0}^{1}\varepsilon_k^{(i,j)} \leqslant \xi.$$

As in [SWW18], we apply the sequential encoding method [Sin05] to convert this inequality constraint into a sequence of Boolean expressions.

So far, the problems of searching for linear trails can be converted into SAT problems, and we invoke the SAT solver Cryptominisat5 [SNC09] to solve the problems in this paper. At last, we refer readers to [SWW18] for more information about the automatic technique.

[1] https://web.archive.org/web/20131022021257/http://www.sontrak.com/

3 Linear Distinguishers in the Attacks

In this section, we first introduce the method to evaluate the complexity of the linear attack. After that, we propose the linear approximations employed in the attacks.

3.1 Complexity Analysis of the Linear Attack

Let $u \xrightarrow{r\text{-round}} v$ be an r-round linear approximation of an iterated block cipher with block size n bits. Denote the absolute value of the correlation regarding the dominating characteristic $\tau = (\tau_0, \tau_1, \ldots, \tau_r)$ with $\tau_0 = u$ and $\tau_r = v$ of this linear approximation as c. The expected linear potential $ELP(u, v)$ of linear approximation equals the quadratic sum of the correlations for all characteristics belonging to the linear approximation.

In the linear attack based on this approximation, we perform partial encryption and decryption and estimate the linear approximation's empirical correlation by guessing the values of some round keys. The key candidate is accepted if its empirical correlation is greater than the predefined value of the threshold Θ.

Denote the probability that the correct key survives as P_S, which is called the success probability of the attack. 2^{-a} represents the proportion of keys that are discarded in the screening process, and we call the exponent a the advantage [Sel08] of the attack. In the hypothesis test, the probabilities for the two types of errors are calculated as

$$\alpha_0 = 2^{-a} \text{ and } \alpha_1 = 1 - P_S,$$

where α_0 is the probability that a wrong key candidate is accepted, and α_1 is the probability that the correct key is rejected.

Suppose that N known plaintexts participate in the key-recovery attack. The threshold value is set as

$$\Theta = \sqrt{1/N} \cdot \Phi^{-1}\left(1 - 2^{-(a+1)}\right),$$

where Φ stands for the cumulative distribution function of the standard normal distribution. With the method in [BN17], the success probability of the linear attack is

$$P_S \approx \Phi\left(\frac{c \cdot \sqrt{N} - \Phi^{-1}\left(1 - 2^{-(a+1)}\right) \cdot \sqrt{1 + N \cdot 2^{-n}}}{\sqrt{1 + N \cdot (ELP(u, v) - c^2)}}\right). \tag{1}$$

3.2 Linear Approximations in the Attacks on Three AEADs

In the test phase, we observe that the absolute value of the correlation for the optimal 11-round linear trail regarding `GIFT-128` is 2^{-31}. Given the linear hull effect of `GIFT-128` is relatively weak, we guess that the data requirements for linear attacks with 11-round approximations may be larger than 2^{64}, which is the common upper bound for the three AEADs analysed in this paper. Thus, we utilise 10-round linear approximations to realise key-recovery attacks for the three AEADs.

Beyond that, in the test, we notice that the maximum absolute value of the correlation for the 10-round linear trail is 2^{-26}. However, a further investigation reveals that all the 16384 10-round trails with this optimal correlation do not result in good performances in the key-recovery attacks. That is to say, the number of appended rounds concerning the optimal trail in the key-recovery attack is shorter than those regarding some 10-round distinguishers with a bit lower correlations. Therefore, the dominating linear characteristics of the three 10-round linear approximations in this section are not the optimal ones.

3.2.1 10-Round Linear Approximation in the Attack on GIFT-COFB

Since the designers claim 64 bits IND-CPA security under the nonce respecting scenario, the data requirement of a valid attack on GIFT-COFB should be lower than 2^{64}. Besides, as the most significant 64 bits of the input for the E_K functions in the data processing phase are masked by the unknown value L, the verification of the linear relation should be irrelevance with these bits. To accomplish the search of linear approximations fulfilling this restriction, we attempt to encode it with Boolean equations. These extra Boolean equations are integrated into the original SAT problem in Sect. 2.3 to create a specialised SAT problem targeting the conditional linear trail. As a result, the outcome of the specialised SAT problem returned by the SAT solver will automatically coordinate with the attack setting.

Specialised SAT problems Given that GIFT-128 achieves full diffusion after four rounds, we conjecture the maximum number of rounds annexed before the linear distinguisher in the attack on GIFT-COFB is three. Regarding the three rounds extended before the linear approximation, we introduce extra variables to locate the bits involved in verifying the linear relation. According to the functionality, the extra Boolean equations in the specialised SAT problem can be divided into three parts.

Part I: Identifying the necessary bits for the calculation of the linear relation For each S-box in the three appended rounds, we introduce four Boolean variables $(\mu_0, \mu_1, \mu_2, \mu_3)$ to signify whether the four values of the input bits (x_0, x_1, x_2, x_3) should be known for checking the linear relation, respectively. To be explicit, for $0 \leqslant i \leqslant 3$, we set μ_i as

$$\mu_i = \begin{cases} 1, & \text{if the value of } x_i \text{ should be known for the verification of the linear relation} \\ 0, & \text{otherwise} \end{cases}.$$

Likewise, we utilise four Boolean variables $(\nu_0, \nu_1, \nu_2, \nu_3)$ to stand for whether the four values of the output bits (y_0, y_1, y_2, y_3) are the necessary bits for calculating the linear relation. Since the S-box is a non-linear operation, the four values of the input bits must be known if any of the four output bits turn into necessary bits. Consequently, the newly included variables should satisfy the following constraint

$$\mu_0 = \mu_1 = \mu_2 = \mu_3 = (\nu_0 \vee \nu_1 \vee \nu_2 \vee \nu_3). \tag{2}$$

Then, we consider an 8-bit Boolean function

$$f'(\mu_0 \| \mu_1 \| \mu_2 \| \mu_3 \| \nu_0 \| \nu_1 \| \nu_2 \| \nu_3) = \begin{cases} 1, & \text{if the inputs validate Eq. (2)} \\ 0, & \text{otherwise} \end{cases}.$$

Note that this constraint can be converted into Boolean expressions with the method in [SWW18], which is also recalled in Sect. 2.3. These expressions constitute the first part of extra Boolean equations in the specialised SAT problem.

Part II: Ensuring the irrelevance with the most significant 64 input bits Denote $\mu_k^{(i,j)}$ and $\nu_k^{(i,j)}$ the variables for the j-th S-box in the i-th appended round before the linear approximation, where $0 \leqslant i \leqslant 2$, $0 \leqslant j \leqslant 31$, $0 \leqslant k \leqslant 3$. To make sure that the evaluation of the linear relation does not rely on the most significant 64 bits of the input, we should supplement the following 64 equations to the SAT problem

$$\overline{\mu_k^{(0,j)}} = 1,\ 0 \leqslant j \leqslant 15,\ 0 \leqslant k \leqslant 3.$$

These equations are the second part of extra Boolean equations in the specialised SAT problem.

Part III: Connecting the extended rounds with the linear trail Suppose that the input mask of the 10-round linear trail in the original SAT problem is symbolically represented as $(a_0, a_1, \ldots, a_{127})$. The values of the bits masked with $a_i = 1$ should be known so that we can estimate the validity of the linear relation. Hence, to establish the connection between the affixed three rounds and the linear trail, we generate the following 128 equations

$$\nu_k^{(2,j)} = a_{P(4\cdot j+k)},\ 0 \leqslant j \leqslant 31,\ 0 \leqslant k \leqslant 3,$$

where P is the bit permutation in the PermBits step. Equivalently, these equations can be transformed into 256 Boolean expressions

$$\nu_k^{(2,j)} \vee \overline{a_{P(4\cdot j+k)}} = 1,\ \overline{\nu_k^{(2,j)}} \vee a_{P(4\cdot j+k)} = 1,\ 0 \leqslant j \leqslant 31,\ 0 \leqslant k \leqslant 3.$$

These equations form the third part of extra Boolean equations in the specialised SAT problem.

We apply the specialised SAT problem to assist the search of linear distinguishers for GIFT-COFB. In the test phase, we find no trail satisfying the specialised SAT problem if the absolute value of the objective correlation for the linear trail is fixed as 2^{-26} or 2^{-27}. When the objective correlation is set as 2^{-28}, 16896 linear trails are returned by the SAT solver. Indeed, these trails can be forward extended by three rounds in the key-recovery phase. Nevertheless, for all the 16896 distinguishers, we also notice that appending three rounds after the distinguisher will increase the number of guessed subkey bits. The considerable time complexity disables us from performing a 16-round attack. So, we lower the objective correlation of the specialised SAT problem to 2^{-29} and discover 424320 linear trails. When we append three rounds both before and after these trails, we observe the minimum number of guessed subkey bits is 69. Furthermore, the dominating trail in the following linear approximation is the unique trail that achieves the minimum number of guessed subkey bits.

Taken together, we exploit a 10-round linear approximation $u_1 \xrightarrow{\text{10-round}} v_1$ with $ELP(u_1, v_1) = 2^{-57.68}$, where

$$\begin{aligned} u_1 &= \texttt{0x0000 0x0000 0x0000 0x0000 0xa002 0x0000 0x0000 0x5001}, \\ v_1 &= \texttt{0x0000 0x0000 0x0000 0x0000 0x0000 0x0044 0x0000 0x0022}. \end{aligned}$$

The dominating linear characteristic with correlation $c = 2^{-29}$ is exhibited in Figure 4.

3.2.2 10-Round Linear Approximation in the Attack on SUNDAE-GIFT

Unlike the case in GIFT-COFB, because there is no limitation at the input of the E_K function, we purpose to attach four rounds and three rounds before and after the distinguisher. The original SAT problem is exploited to search for the linear distinguisher of SUNDAE-GIFT. Again, we note that all the 16384 optimal 10-round trails of GIFT-128 with correlation 2^{-26} cannot derive 17-round attacks for the massive number of guessed subkey bits. Therefore, we reduce the objective correlation to 2^{-27}. As there are numerous 10-round trails with correlation 2^{-27}, the SAT solver outputs 919882 solutions and terminates the search for memory error. We explore the feasibility of employing the 919882 trails to implement 17-round attacks and find that none of them can complete this task. We further lower the objective correlation to 2^{-28} and analyse the 658845 trails returned by the solver. Among the 658845 trails, we choose the unique one attaining the minimum number of guessed subkey bits, say 88, as the final distinguisher in the attack, which is the dominating trail in the following linear approximation.

For the attack on SUNDAE-GIFT, we employ a 10-round linear approximation $u_2 \xrightarrow{\text{10-round}} v_2$ with $ELP(u_2, v_2) = 2^{-55.36}$, where

$$\begin{aligned} u_2 &= \texttt{0x0000 0x0000 0x002a 0x002a 0x0000 0x0000 0x0000 0x0000}, \\ v_2 &= \texttt{0x0044 0x0000 0x0022 0x0000 0x0000 0x0000 0x0000 0x0000}. \end{aligned}$$

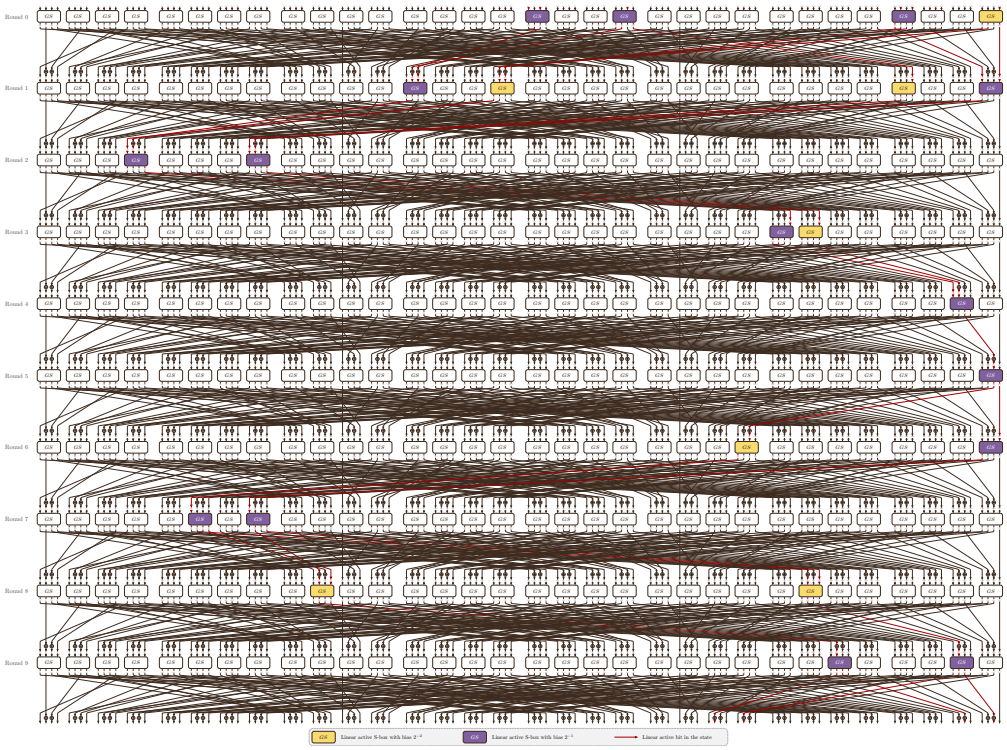


Figure 4: 10-round trail with $c = 2^{-29}$ in the approximation concerning GIFT-COFB.

The dominating linear characteristic with correlation $c = 2^{-28}$ is shown in Figure 5.

3.2.3 10-Round Linear Approximation in the Attack on HyENA

Similar to the case in GIFT-COFB, as the unknown value Δ masks the least significant 64 bits of the input for the E_K functions in the data processing phase, the estimation of the linear equation must have no relevance to these bits. Likewise, we suspect the maximum number of rounds extended before the linear distinguisher in the attack on HyENA is three, considering that four rounds of GIFT-128 accomplish full diffusion. Accordingly, we employ the specialised SAT problem in Sect. 3.2.1 and replace the second part of extra Boolean equations with the following ones

$$\overline{\mu_k^{(0,j)}} = 1,\ 16 \leqslant j \leqslant 31,\ 0 \leqslant k \leqslant 3.$$

Then, the result of this specialised SAT problem will automatically suit the attack setting.

In the test, we obtain no trail satisfying the specialised SAT problem if the absolute value of the objective correlation for the linear trail is fixed as 2^{-26} or 2^{-27}. When the objective correlation is set as 2^{-28}, the SAT solver outputs 14848 trails. We append three rounds both before and after these trails and compute the minimum number of guessed subkey bits. The minimum number of guessed subkey bits is 71, and four trails among the 14848 ones reach this minimum value. The dominating trail of the following linear approximation is one of the four trails that possesses the most significant linear hull effect.

Thus, the 10-round linear approximation $u_3 \xrightarrow{\text{10-round}} v_3$ with $ELP(u_3, v_3) = 2^{-55.36}$ is utilised, where

$$\begin{aligned} u_3 &= \texttt{0x0000 0x0000 0x008a 0x8a00 0x0000 0x0000 0x0000 0x0000}, \\ v_3 &= \texttt{0x0044 0x0000 0x0022 0x0000 0x0000 0x0000 0x0000 0x0000}. \end{aligned}$$

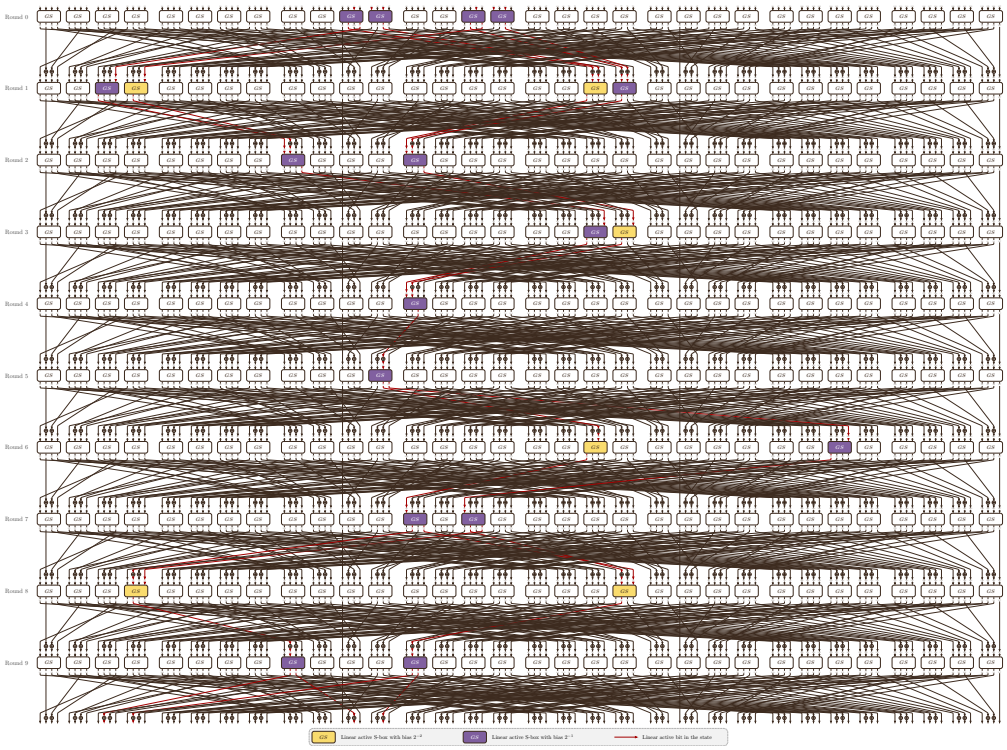


Figure 5: 10-round trail with $c = 2^{-28}$ in the approximation concerning SUNDAE-GIFT.

The dominating linear characteristic with correlation $c = 2^{-28}$ is presented in Figure 6.

4 Cryptanalyses of Three AEADs Based on GIFT-128

Based on the linear approximations in Sect. 3, we present linear attacks on the three AEADs in this section.

4.1 Linear Cryptanalysis of GIFT-COFB

We target the encryption functions in the message processing phase highlighted in red in Figure 1. Suppose we obtain a plaintext-ciphertext pair $(M_1\|M_2\|\cdots\|M_m, C_1\|C_2\|\cdots\|C_m)$ of GIFT-COFB. With this information, the structure of GIFT-COFB enables us to compute the values of the least significant 64 bits of X_{a+i} and the 128 bits of Y_{a+i} for $1 \leqslant i \leqslant m-2$. Then, we can launch a linear attack regarding the underlying primitive GIFT-128 with the $(m-2)$ pairs $\{(X_{a+i}[64\text{-}127], Y_{a+i}) \mid 1 \leqslant i \leqslant m-2\}$.

With the 10-round linear approximation $u_1 \xrightarrow{\text{10-round}} v_1$ in Sect. 3.2.1, we launch a 16-round linear attack on GIFT-128 by appending three rounds both before and after the distinguisher. As depicted in Figure 7, the key-recovery attack is realised with the following steps.

S1 We allocate a counter $\mathsf{Cnt}_1[z_1]$ for each of 2^{47} possible values of

$$z_1 = X^{14}[\mathsf{Index}^{\mathsf{S1}}(X^{14})]\|X^{15}[\mathsf{Index}^{\mathsf{S1}}(X^{15})]\|EY^{15}[\mathsf{Index}^{\mathsf{S1}}(EY^{15})]\|t_1,$$

where $\mathsf{Index}^{\mathsf{S1}}(X^{14})$, $\mathsf{Index}^{\mathsf{S1}}(X^{15})$, and $\mathsf{Index}^{\mathsf{S1}}(EY^{15})$ are index sets containing the

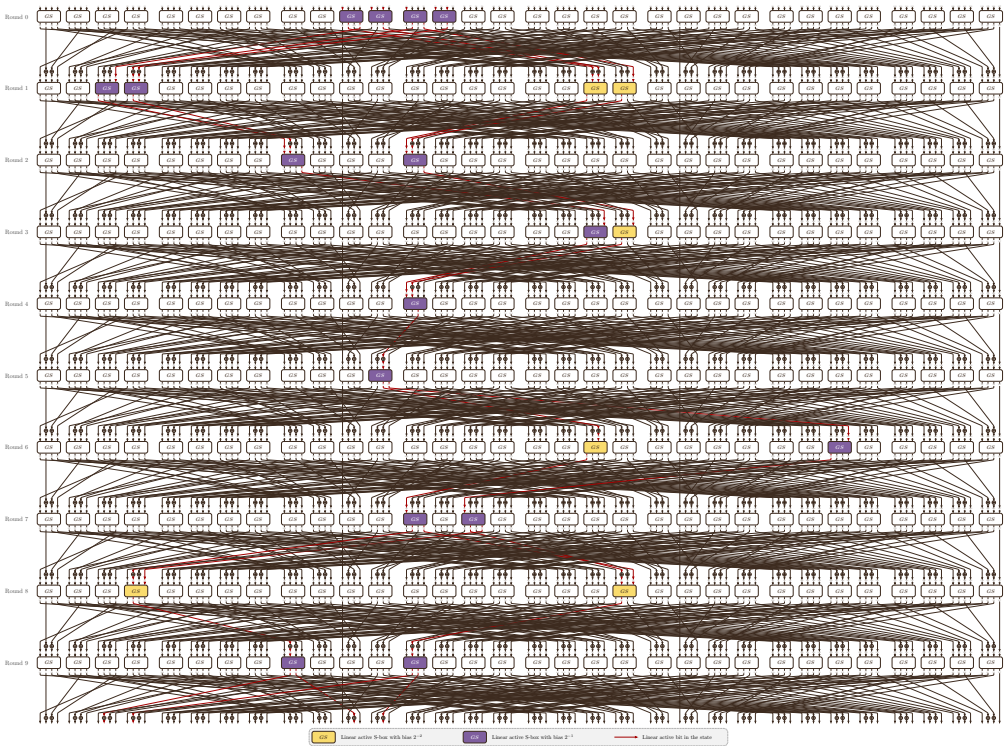


Figure 6: 10-round characteristic with $c = 2^{-28}$ in the approximation concerning HyENA.

bit positions that should be memorised,

$$\begin{aligned}
\mathsf{Index}^{\mathsf{S1}}(X^{14}) &= \{119, 126, 127\},\\
\mathsf{Index}^{\mathsf{S1}}(X^{15}) &= \{84, 86, 117, 119\},\\
\mathsf{Index}^{\mathsf{S1}}(EY^{15}) &= \{4\text{-}7, 12\text{-}15, 20\text{-}23, 36\text{-}39, 44\text{-}47, 52\text{-}55, 68\text{-}71, 76\text{-}79, 100\text{-}103, 108\text{-}111\},
\end{aligned}$$

and t_1 is a 1-bit string calculated as

$$t_1 = X^3[64] \oplus X^3[66] \oplus X^3[78] \oplus X^3[113] \oplus X^3[115] \oplus X^3[127] \oplus X^{14}[118].$$

Then, we compute the value of z_1 for each possible 42-bit subkey value

$$RK^0[\mathsf{Index}^{\mathsf{S1}}(RK^0)] \| RK^1[\mathsf{Index}^{\mathsf{S1}}(RK^1)] \| EK^{14}[\mathsf{Index}^{\mathsf{S1}}(EK^{14})] \| EK^{15}[\mathsf{Index}^{\mathsf{S1}}(EK^{15})],$$

where

$$\begin{aligned}
\mathsf{Index}^{\mathsf{S1}}(RK^0) &= \{8\text{-}15, 24\text{-}31, 56\text{-}63\},\\
\mathsf{Index}^{\mathsf{S1}}(RK^1) &= \{2, 3, 6, 7, 30, 31, 34, 35, 39, 63\},\\
\mathsf{Index}^{\mathsf{S1}}(EK^{14}) &= \{59, 63\},\\
\mathsf{Index}^{\mathsf{S1}}(EK^{15}) &= \{15, 31, 42, 46, 62, 63\},
\end{aligned}$$

by using each of the $(m-2)$ pairs $(X_{a+i}[64\text{-}127], Y_{a+i})$ and update $\mathsf{Cnt}_1[z_1]$ by $\mathsf{Cnt}_1[z_1] + 1$. The time complexity of this step is $(m-2) \cdot 2^{42} \cdot 42$ GS operations.

S2 Allocate a counter $\mathsf{Cnt}_2[z_2]$ for each of 2^{38} possible values of

$$z_2 = X^{14}[\mathsf{Index}^{\mathsf{S2}}(X^{14})] \| EY^{15}[\mathsf{Index}^{\mathsf{S2}}(EY^{15})] \| t_2,$$

where

$$\begin{aligned}\mathsf{Index}^{\mathsf{S2}}(X^{14}) &= \{93, 94, 119, 126, 127\},\\ \mathsf{Index}^{\mathsf{S2}}(EY^{15}) &= \{\text{4-7}, \text{12-15}, \text{36-39}, \text{44-47}, \text{68-71}, \text{76-79}, \text{100-103}, \text{108-111}\},\\ t_2 &= t_1 \oplus X^{14}[85] \oplus X^{14}[86].\end{aligned}$$

For each possible 5-bit subkey value $EK^{14}[47]\|EK^{15}[10, 11, 26, 27]$, we compute the value of z_2 and update $\mathsf{Cnt}_2[z_2]$ by $\mathsf{Cnt}_2[z_2] + \mathsf{Cnt}_1[z_1]$. The time complexity of this step is $2^{47} \cdot 2^{42} \cdot 2^5 \cdot 4\ GS$ operations.

S3 Allocate a counter $\mathsf{Cnt}_3[z_3]$ for each of 2^{25} possible values of

$$z_3 = X^{14}[52, 60, 61, 93, 94, 119, 126, 127]\|EY^{15}[\text{4-7}, \text{36-39}, \text{68-71}, \text{100-103}]\|t_3,$$

where $t_3 = t_2 \oplus X^{14}[53]$. Then, for each possible 10-bit subkey value

$$EK^{14}[26, 30]\|EK^{15}[6, 7, 22, 23, 38, 39, 54, 55],$$

we compute the value of z_3 and update $\mathsf{Cnt}_3[z_3]$ by $\mathsf{Cnt}_3[z_3] + \mathsf{Cnt}_2[z_2]$. The time complexity of this step is $2^{38} \cdot 2^{47} \cdot 2^{10} \cdot 6\ GS$ operations.

S4 Initialise a counter Σ. For each possible 12-bit subkey value

$$EK^{13}[62]\|EK^{14}[10, 11, 14]\|EK^{15}[2, 3, 18, 19, 34, 35, 50, 51],$$

we compute the value of t_4

$$t_4 = t_3 \oplus X^{13}[89] \oplus X^{13}[93] \oplus X^{13}[122] \oplus X^{13}[126].$$

If the value of t_4 equals zero, we update Σ as $\Sigma + \mathsf{Cnt}_3[z_3]$. The time complexity of this step is $2^{25} \cdot 2^{57} \cdot 2^{12} \cdot 10\ GS$ operations.

The threshold is set as Θ. The key guess will be accepted as a candidate if the value of the counter Σ validates the condition $|\Sigma/(m-2) - 0.5| > \Theta$. All master keys that are compatible with the guessed 69 subkey bits are tested exhaustively against a maximum of two plaintext-ciphertext pairs.

Complexity Analysis We set the advantage of the attack as $a = 5.20$ and the number of blocks m in the message as $2^{62.10}$, which constitutes the data complexity of this attack. With Eq. (1), we obtain the success probability $P_S = 80.01\%$. The time complexity of the attack is composed of the time complexity in the subkey enumeration phase as in Steps S1 - S4 and the time to check the remaining 59-bit value in the master key exhaustively. In this case, the total time complexity of the attack is $2^{122.80}$. Since $\mathsf{Cnt}_1[z_1]$ constitutes the largest memory, the memory complexity is roughly 2^{47}.

4.2 Linear Cryptanalysis of `SUNDAE-GIFT`

We aim at the encryption functions in the ciphertext generating phase highlighted in red in Figure 2. With a plaintext-ciphertext pair $(M_1\|M_2\|\cdots\|M_{\ell_M}, C_1\|C_2\|\cdots\|C_{\ell_M})$ of `SUNDAE-GIFT`, we can generate $(\ell_M - 1)$ plaintext-ciphertext pairs $\{(M_i \oplus C_i, M_{i+1} \oplus C_{i+1}) \mid 1 \leqslant i \leqslant \ell_M - 1\}$ for the underlying primitive `GIFT-128`.

With the 10-round linear approximation $u_2 \xrightarrow{\text{10-round}} v_2$ in Sect. 3.2.2, we launch a 17-round linear attack on `GIFT-128` by appending four rounds and three rounds before and after the distinguisher, respectively. The key-recovery procedure is illustrated in Figure 8. We adopt the following steps to accomplish the key-recovery attack.

S1 We allocate a counter $\mathsf{Cnt}_1[z_1]$ for each of 2^{49} possible values of

$$z_1 = Z^1[\text{64-79, 96-111}] \| EY^{15}[\text{8-11, 40-43, 72-75, 104-107}] \| t_1,$$

where $t_1 = Y^3[34] \oplus Y^3[98] \oplus X^{14}[9] \oplus X^{14}[13]$. Then, we compute the value of z_1 for each possible 64-bit subkey value

$$RK^0[\text{0-31}] \| RK^1[\text{0-7}] \| RK^2[49] \| EK^{15}[\mathsf{Index}^{\mathsf{S1}}(EK^{15})] \| EK^{16}[\mathsf{Index}^{\mathsf{S1}}(EK^{16})],$$

by using each of the $(\ell_M - 1)$ pairs $(M_i \oplus C_i, M_{i+1} \oplus C_{i+1})$ and update $\mathsf{Cnt}_1[z_1]$ by $\mathsf{Cnt}_1[z_1] + 1$, where

$$\begin{aligned} \mathsf{Index}^{\mathsf{S1}}(EK^{15}) &= \{0, 16, 17, 32, 33, 48, 49\}, \\ \mathsf{Index}^{\mathsf{S1}}(EK^{16}) &= \{4, 5, 12, 13, 20, 21, 28, 29, 36, 37, 44, 45, 52, 53, 60, 61\}. \end{aligned}$$

The time complexity of this step is $(\ell_M - 1) \cdot 2^{64} \cdot 76$ GS operations.

S2 Allocate a counter $\mathsf{Cnt}_2[z_2]$ for each of 2^{33} possible values of

$$z_2 = Z^1[\text{96-111}] \| EY^{15}[\text{8-11, 40-43, 72-75, 104-107}] \| t_2,$$

where $t_2 = t_1 \oplus Y^3[50] \oplus Y^3[114]$. We compute the value of z_2 for each possible 10-bit subkey value $RK^1[\text{32-39}] \| RK^2[56, 57]$ and update $\mathsf{Cnt}_2[z_2]$ by $\mathsf{Cnt}_2[z_2] + \mathsf{Cnt}_1[z_1]$. The time complexity of this step is $2^{49} \cdot 2^{64} \cdot 2^{10} \cdot 6$ GS operations.

S3 Allocate a counter $\mathsf{Cnt}_3[z_3]$ for each of 2^{17} possible values of

$$z_3 = EY^{15}[\text{8-11, 40-43, 72-75, 104-107}] \| t_3, \ t_3 = t_2 \oplus Y^3[56] \oplus Y^3[120].$$

For each possible 10-bit subkey value $RK^1[\text{48-55}] \| RK^2[60, 61]$, we compute the value of z_3 and update $\mathsf{Cnt}_3[z_3]$ by $\mathsf{Cnt}_3[z_3] + \mathsf{Cnt}_2[z_2]$. The time complexity of this step is $2^{33} \cdot 2^{74} \cdot 2^{10} \cdot 6$ GS operations.

S4 Initialise a counter Σ. For each possible 4-bit subkey value $EK^{15}[5, 21, 36, 52]$, we compute the value of $t_4 = t_3 \oplus X^{14}[42] \oplus X^{14}[46]$. If t_4 equals zero, we update Σ as $\Sigma + \mathsf{Cnt}_3[z_3]$. The time complexity of this step is $2^{17} \cdot 2^{84} \cdot 2^4 \cdot 6$ GS operations.

The threshold is set as Θ. The key guess will be accepted as a candidate if the value of the counter Σ validates the condition $|\Sigma/(\ell_M - 1) - 0.5| > \Theta$. All master keys that are compatible with the guessed 88 subkey bits are tested exhaustively against a maximum of two plaintext-ciphertext pairs.

Complexity Analysis We set the advantage of the attack as $a = 6.00$ and the number of blocks ℓ_M in the message as $2^{61.51}$, which constitutes the data complexity of this attack. With Eq. (1), we obtain the success probability $P_S = 80.01\%$. The time complexity of the attack is composed of the time complexity in the subkey enumeration phase as in Steps S1 - S4 and the time to check the remaining 40-bit value in the master key exhaustively. In this case, the total time complexity of the attack is $2^{123.38}$. Since $\mathsf{Cnt}_1[z_1]$ constitutes the largest memory, the memory complexity is roughly 2^{49}.

4.3 Linear Cryptanalysis of HyENA

The target of this attack is the encryption functions in the message processing phase highlighted in red in Figure 3. Given a pair $(M_0 \| M_1 \| \cdots \| M_{m-1}, C_0 \| C_1 \| \cdots \| C_{m-1})$ of HyENA, the values of the most significant 64 bits of the input and the full state of the output for the E_K functions in red can be generated. These pairs can be used to launch a linear attack on the underlying primitive GIFT-128.

With the 10-round linear approximation $u_3 \xrightarrow{\text{10-round}} v_3$ in Sect. 3.2.3, we launch a 16-round linear attack on GIFT-128 by appending three rounds both before and after the distinguisher. The key-recovery attack is performed as follows, which is also in Figure 9.

S1 Allocate a counter $\mathsf{Cnt}_1[z_1]$ for each of 2^{52} possible values of

$$z_1 = X^{14}[32, 40, 41] \| EY^{15}[\mathsf{Index}^{\mathsf{S1}}(EY^{15})] \| t_1,$$

where

$$\begin{aligned} \mathsf{Index}^{\mathsf{S1}}(EY^{15}) &= \left\{ \begin{array}{l} \text{0-3, 16-19, 24-27, 32-35, 48-51, 56-59, 64-67,} \\ \text{80-83, 88-91, 96-99, 112-115, 120-123} \end{array} \right\}, \\ t_1 &= Y^2[40] \oplus Y^2[50] \oplus Y^2[56] \oplus Y^2[72] \oplus Y^2[82] \oplus Y^2[88]. \end{aligned}$$

Then, we compute the value of z_1 for each possible 41-bit subkey value

$$RK^0[\mathsf{Index}^{\mathsf{S1}}(RK^0)] \| RK^1[\mathsf{Index}^{\mathsf{S1}}(RK^1)] \| EK^{14}[17, 21] \| EK^{15}[4, 37, 53],$$

by using each of the $(m-1)$ pairs and update $\mathsf{Cnt}_1[z_1]$ by $\mathsf{Cnt}_1[z_1] + 1$, where

$$\begin{aligned} \mathsf{Index}^{\mathsf{S1}}(RK^0) &= \{\text{16-23, 32-39, 48-55}\}, \\ \mathsf{Index}^{\mathsf{S1}}(RK^1) &= \{20, 21, 24, 25, 28, 29, 36, 37, 40, 41, 44, 45\}. \end{aligned}$$

The time complexity of this step is $(m-1) \cdot 2^{41} \cdot 40$ GS operations.

S2 Allocate a counter $\mathsf{Cnt}_2[z_2]$ for each of 2^{40} possible values of

$$z_2 = EY^{15}[\mathsf{Index}^{\mathsf{S2}}(EY^{15})] \| X^{14}[0, 3, 8, 11, 32, 40, 41] \| t_1,$$

where $\mathsf{Index}^{\mathsf{S2}}(EY^{15}) = \{\text{16-19, 24-27, 48-51, 56-59, 80-83, 88-91, 112-115, 120-123}\}$. For each possible 7-bit subkey value $EK^{14}[1, 5] \| EK^{15}[0, 1, 17, 33, 49]$, we compute the value of z_2 and update $\mathsf{Cnt}_2[z_2]$ by $\mathsf{Cnt}_2[z_2] + \mathsf{Cnt}_1[z_1]$. The time complexity of this step is $2^{52} \cdot 2^{41} \cdot 2^7 \cdot 6$ GS operations.

S3 Allocate a counter $\mathsf{Cnt}_3[z_3]$ for each of 2^{26} possible values of

$$z_3 = EY^{15}[\text{24-27, 56-59, 88-91, 120-123}] \| X^{14}[0, 3, 8, 11, 32, 40, 41, 73, 74] \| t_2,$$

where $t_2 = t_1 \oplus X^{14}[65] \oplus X^{14}[66]$. For each possible 11-bit subkey value

$$EK^{14}[32, 33, 36] \| EK^{15}[8, 9, 24, 25, 40, 41, 56, 57],$$

we compute the value of z_3 and update $\mathsf{Cnt}_3[z_3]$ by $\mathsf{Cnt}_3[z_3] + \mathsf{Cnt}_2[z_2]$. The time complexity of this step is $2^{40} \cdot 2^{48} \cdot 2^{11} \cdot 6$ GS operations.

S4 Initialise a counter Σ. For each possible 12-bit subkey value

$$EK^{13}[20, 23] \| EK^{14}[48, 52] \| EK^{15}[12, 13, 28, 29, 44, 45, 60, 61],$$

we compute the value of $t_3 = t_2 \oplus X^{13}[9] \oplus X^{13}[13] \oplus X^{13}[42] \oplus X^{13}[46]$. If t_3 equals zero, we update Σ as $\Sigma + \mathsf{Cnt}_3[z_3]$. The time complexity of this step is $2^{26} \cdot 2^{59} \cdot 2^{12} \cdot 10$ GS operations.

The threshold is set as Θ. The key guess will be accepted as a candidate if the value of the counter Σ validates the condition $|\Sigma/(m-1) - 0.5| > \Theta$. All master keys that are compatible with the guessed 71 subkey bits are tested exhaustively against a maximum of two plaintext-ciphertext pairs.

Complexity Analysis We set the advantage of the attack as $a = 6.00$ and the number of blocks m in the message as $2^{61.51}$, which constitutes the data complexity of this attack. With Eq. (1), we obtain the success probability $P_S = 80.01\%$. The time complexity of the attack is composed of the time complexity in the subkey enumeration phase as in Steps S1 - S4 and the time to check the remaining 57-bit value in the master key exhaustively. In this case, the total time complexity of the attack is $2^{122.00}$. Since $\mathsf{Cnt}_1[z_1]$ constitutes the largest memory, the memory complexity is roughly 2^{52}.

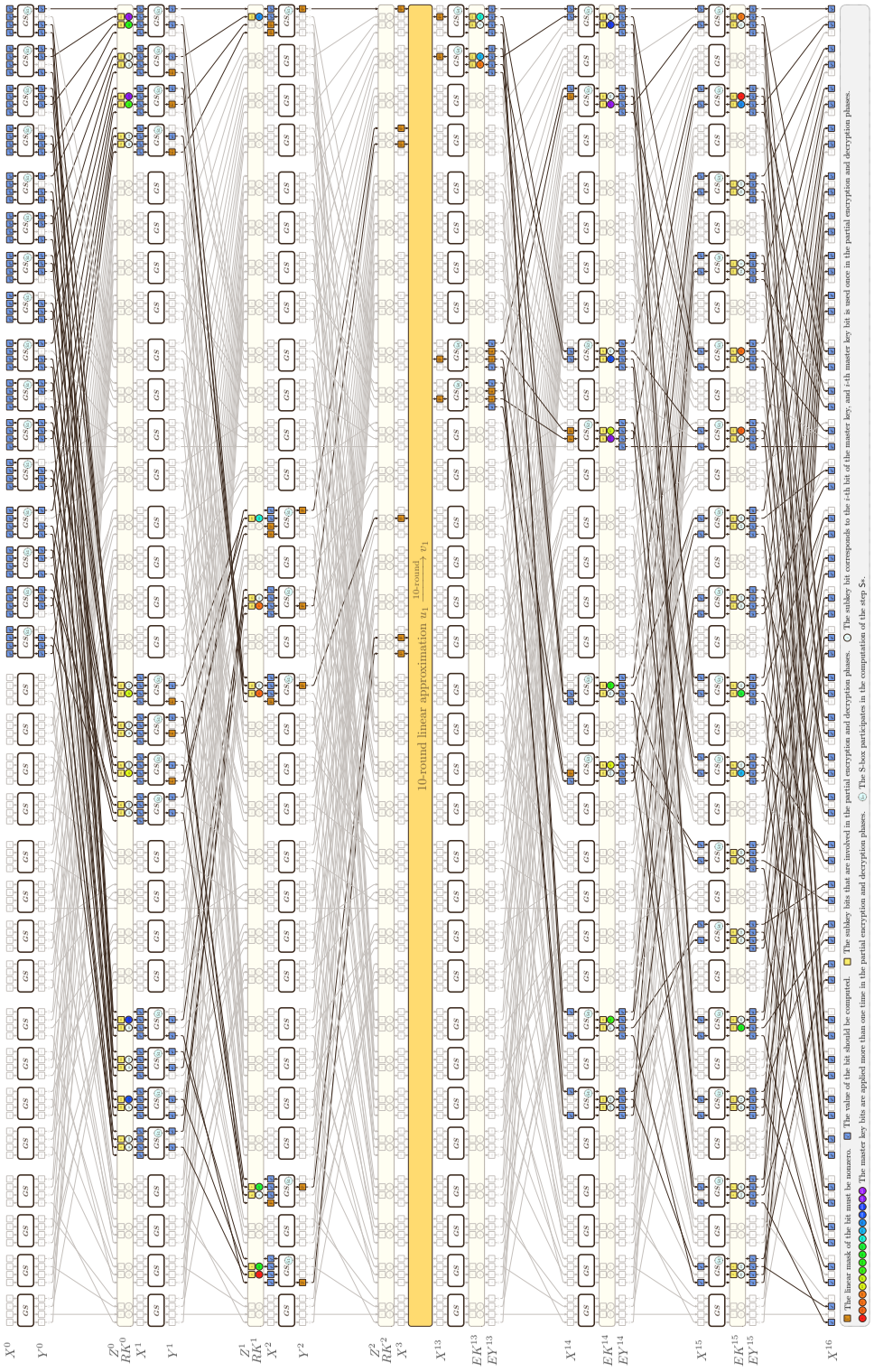


Figure 7: Key-recovery attack on 16-round GIFT-128 in GIFT-COFB.

Figure 8: Key-recovery attack on 17-round `GIFT-128` in `SUNDAE-GIFT`.

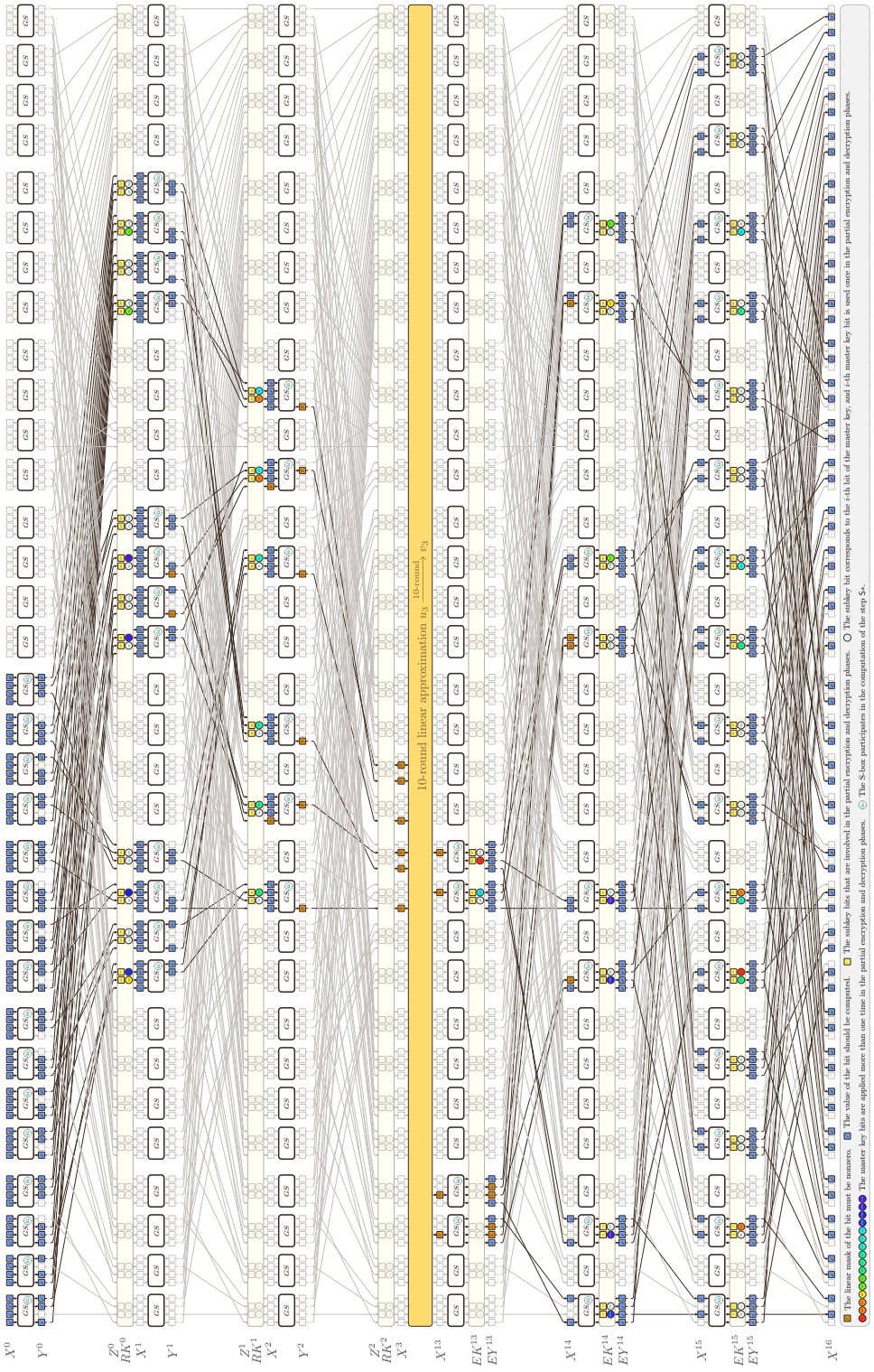


Figure 9: Key-recovery attack on 16-round GIFT-128 in HyENA.

5 24-Round Linear Attack on GIFT-128

This section provides a 19-round linear approximation for GIFT-128 first and then gives a 24-round linear attack with this linear approximation.

5.1 19-Round Linear Approximation of GIFT-128

With the experimental result, we identify that the absolute value of the correlation for the optimal 20-round linear trail is 2^{-64}. Considering the weak linear hull effect of GIFT-128, we think that the data complexity of a linear attack with a 20-round approximation may be larger than 2^{128}. Hence, we adopt 19-round linear approximations to launch the key-recovery attack.

Note that the maximum absolute value of the correlation for the 19-round linear trail is 2^{-59}, and we get 8192 trails with the optimal correlation. For all these trails, if we append three rounds both before and after the distinguisher, the time complexity will exceed 2^{128}. So, we turn to check the possibility of extending three and two rounds before and after the distinguisher and find that 24 trails enable us to give valid 24-round linear attacks. Since the linear hull effects of the 24 linear approximations are almost the same, we randomly pick one as the distinguisher in the attack.

The 24-round linear attack is based on a 19-round linear approximation $u_4 \xrightarrow{\text{19-round}} v_4$ with $ELP(u_4, v_4) = 2^{-117.43}$, where

$$\begin{aligned} u_4 &= \texttt{0x0000 0x0000 0x0000 0x0000 0x000c 0x000c 0x0600 0x0000}, \\ v_4 &= \texttt{0x0400 0x0040 0x0202 0x0000 0x0001 0x0010 0x0000 0x0000}. \end{aligned}$$

The number of characteristics belonging to this linear approximation with different correlations is demonstrated in Figure 10. The dominating linear characteristic with correlation $c = 2^{-59}$ is exhibited in Figure 11.

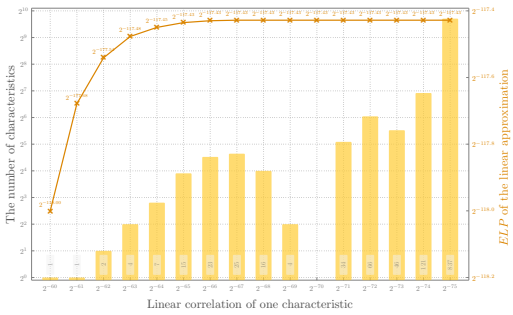


Figure 10: Distribution of characteristics belonging to the 19-round linear approximation.

5.2 Linear Attack on GIFT-128

With the 19-round linear hull, we launch a 24-round linear attack on GIFT-128 by appending three and two rounds before and after the distinguisher, respectively. The key-recovery attack is shown in Figure 12.

Suppose that the number of plaintext-ciphertext pairs utilised in the attack is N. We allocate a global counter $\mathsf{Cnt}_0[z_0]$ for each of 2^{105} possible values of

$$z_0 = Z^0[\text{80-111}] \| EY^{23}[\text{0-11}, \text{16-23}, \text{32-43}, \text{48-55}, \text{64-67}, \text{72-75}, \text{80-87}, \text{100-107}, \text{112-119}] \| t_1,$$

where t_1 is computed as

$$\begin{aligned} t_1 = \; & \left(\overline{EY^{23}[68]} \wedge EY^{23}[71]\right) \oplus EY^{23}[69] \oplus EY^{23}[70] \\ & \oplus \left(\overline{EY^{23}[96]} \wedge EY^{23}[99]\right) \oplus EY^{23}[97] \oplus EY^{23}[98]. \end{aligned}$$

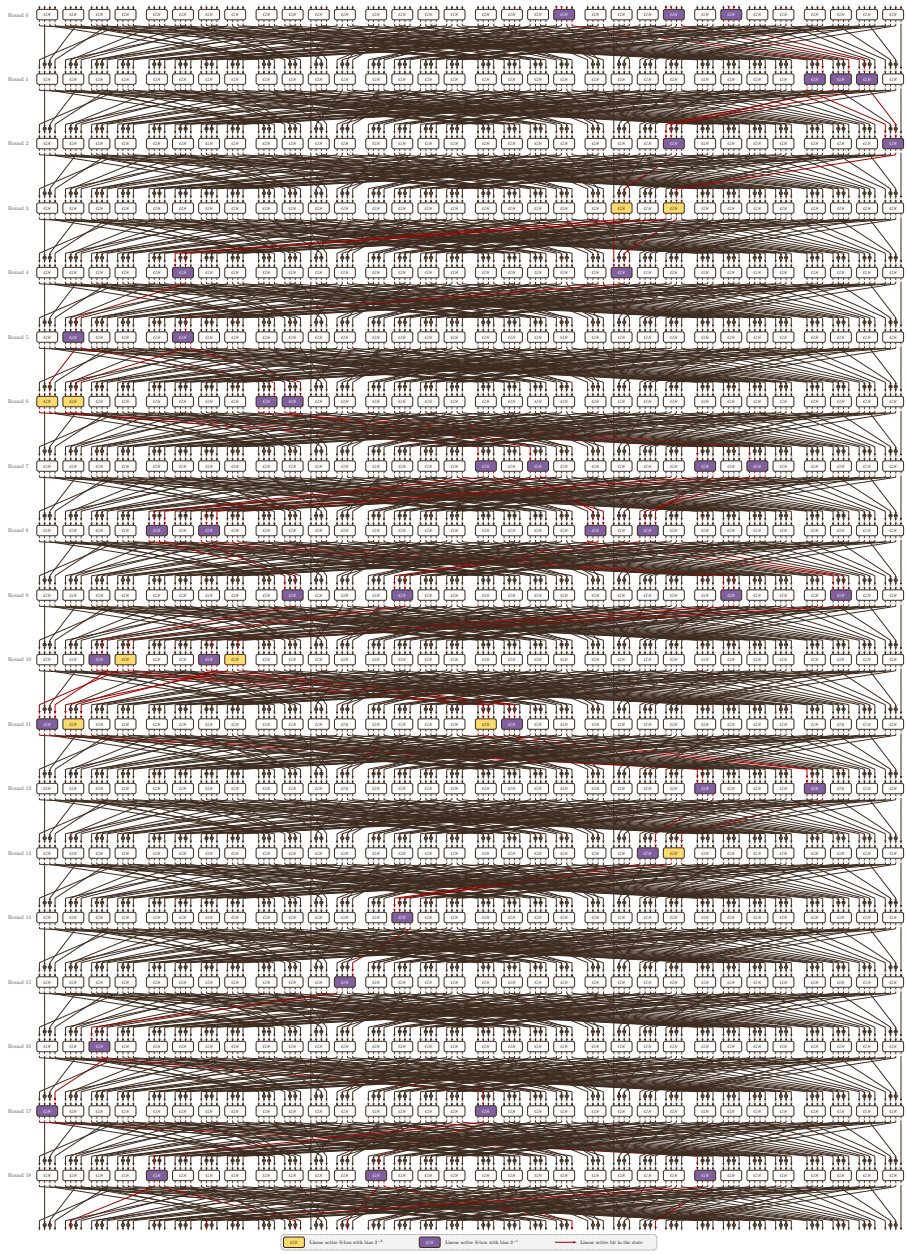


Figure 11: 19-round trail with $c = 2^{-59}$.

For each of N plaintext-ciphertext pairs, we compute the value of z_0 and update $\mathsf{Cnt}_0[z_0]$ as $\mathsf{Cnt}_0[z_0] + 1$. The time complexity of this step $\mathsf{S0}$ is N 24-round of encryptions. Then, we exploit a similar method as in Sect. 4.1 to realise the enumeration of subkey bits. The detailed information about the counters constructed in the subkey enumerating phase can be found in Table 2. After executing step $\mathsf{S20}$, we initialise a counter Σ. If t_{10} equals zero, we update the value Σ.

We set the threshold as Θ. The key guess will be accepted as a candidate if the value of the counter Σ validates the condition $|\Sigma/N - 0.5| > \Theta$. All master keys that are compatible with the guessed 62 subkey bits are tested exhaustively against a maximum of two plaintext-ciphertext pairs.

Table 2: Detailed computation of complexity.

Step	Guessed subkey	Information about the counter		Time complexity (GS operations)
		Object	Quantity	
S1	$EK^{23}[0,1]$	Z^0[80-111]$\|\|EY^{23}$[4-11, 16-23, 32-43, 48-55, 64-67, 72-75, 80-87, 100-107, 112-119]$\|\|t_2$, $t_2 = t_1 \oplus X^{23}[2]$	2^{101}	$2^{105} \cdot 2^2$
S2	$EK^{23}[50,51]$	Z^0[80-111]$\|\|EY^{23}$[4-11, 16-23, 32-43, 48-55, 64-67, 72-75, 80-87, 104-107, 112-119]$\|\|t_3$, $t_3 = t_2 \oplus X^{23[102]}$	2^{97}	$2^{101} \cdot 2^2 \cdot 2^2$
S3	$EK^{23}[16,17]$	$X^{23}[35]\|\|Z^0$[80-111]$\|\|EY^{23}$[4-11, 16-23, 36-43, 48-55, 64-67, 72-75, 80-87, 104-107, 112-119]$\|\|t_3$	2^{94}	$2^{97} \cdot 2^4 \cdot 2^2$
S4	$EK^{23}[32,33]$	Z^0[80-111]$\|\|EY^{23}$[4-11, 16-23, 36-43, 48-55, 72-75, 80-87, 104-107, 112-119]$\|\|t_4$, $t_4 = t_3 \oplus (\overline{EY^{22}[4]} \wedge EY^{22}[7])$	2^{89}	$2^{94} \cdot 2^6 \cdot 2^2 \cdot 2$
S5	$EK^{23}[2,3]$	$X^{23}[7]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 16-23, 36-43, 48-55, 72-75, 80-87, 104-107, 112-119]$\|\|t_4$	2^{86}	$2^{89} \cdot 2^8 \cdot 2^2$
S6	$EK^{23}[18,19]$	Z^0[80-111]$\|\|EY^{23}$[8-11, 16-23, 40-43, 48-55, 72-75, 80-87, 104-107, 112-119]$\|\|t_5$, $t_5 = t_4 \oplus (\overline{EY^{22}[24]} \wedge EY^{22}[27])$	2^{81}	$2^{86} \cdot 2^{10} \cdot 2^2 \cdot 2$
S7	$EK^{23}[8,9]$	$X^{23}[16]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 20-23, 40-43, 48-55, 72-75, 80-87, 104-107, 112-119]$\|\|t_5$	2^{78}	$2^{81} \cdot 2^{12} \cdot 2^2$
S8	$EK^{23}[24,25]$	$X^{23}[16,49]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 20-23, 40-43, 52-55, 72-75, 80-87, 104-107, 112-119]$\|\|t_5$	2^{75}	$2^{78} \cdot 2^{14} \cdot 2^2$
S9	$EK^{23}[40,41]$	$X^{23}[16,49,82]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 20-23, 40-43, 52-55, 72-75, 84-87, 104-107, 112-119]$\|\|t_5$	2^{72}	$2^{75} \cdot 2^{16} \cdot 2^2$
S10	$EK^{23}[56,57]$	$X^{23}[16,49,82,115]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 20-23, 40-43, 52-55, 72-75, 84-87, 104-107, 116-119]$\|\|t_5$	2^{69}	$2^{72} \cdot 2^{18} \cdot 2^2$
S11	$EK^{22}[38,39]$	Z^0[80-111]$\|\|EY^{23}$[8-11, 20-23, 40-43, 52-55, 72-75, 84-87, 104-107, 116-119]$\|\|t_6$, $t_6 = t_5 \oplus X^{22}[79]$	2^{65}	$2^{69} \cdot 2^{20} \cdot 2^2$
S12	$EK^{23}[10,11]$	$X^{23}[23]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 40-43, 52-55, 72-75, 84-87, 104-107, 116-119]$\|\|t_6$	2^{62}	$2^{65} \cdot 2^{22} \cdot 2^2$
S13	$EK^{23}[26,27]$	$X^{23}[23,52]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 40-43, 72-75, 84-87, 104-107, 116-119]$\|\|t_6$	2^{59}	$2^{62} \cdot 2^{24} \cdot 2^2$
S14	$EK^{23}[42,43]$	$X^{23}[23,52,85]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 40-43, 72-75, 104-107, 116-119]$\|\|t_6$	2^{56}	$2^{59} \cdot 2^{26} \cdot 2^2$
S15	$EK^{23}[58,59]$	$X^{23}[23,52,85,118]\|\|Z^0$[80-111]$\|\|EY^{23}$[8-11, 40-43, 72-75, 104-107]$\|\|t_6$	2^{53}	$2^{56} \cdot 2^{28} \cdot 2^2$
S16	$EK^{22}[44,45]$	Z^0[80-111]$\|\|EY^{23}$[8-11, 40-43, 72-75, 104-107]$\|\|t_7$, $t_7 = t_6 \oplus X^{22}[91]$	2^{49}	$2^{53} \cdot 2^{30} \cdot 2^2$
S17	$EK^{23}[4,5,20,21,36,37,52,53]$	$X^{23}[8,10,41,43,72,74,105,107]\|\|Z^0$[80-111]$\|\|t_7$	2^{41}	$2^{49} \cdot 2^{32} \cdot 2^8 \cdot 4$
S18	$EK^{22}[18,19,22,23]$	Z^0[80-111]$\|\|t_8$, $t_8 = t_7 \oplus X^{22}[38] \oplus X^{22}[46]$	2^{33}	$2^{41} \cdot 2^{40} \cdot 2^4 \cdot 2$
S19	RK^0[49-55]$\|\|RK^1[29]$	Z^0[80-95]$\|\|t_9$, $t_9 = t_8 \oplus Y^2[26] \oplus Y^2[57] \oplus Y^2[121]$	2^{17}	$2^{33} \cdot 2^{44} \cdot 2^8 \cdot 7$
S20	RK^0[40-43, 45-47]$\|\|RK^1[27,58,59]$	$t_{10} = t_9 \oplus Y^2[21] \oplus Y^2[52] \oplus Y^2[116]$	2^1	$2^{17} \cdot 2^{52} \cdot 2^{10} \cdot 7$
Total	-	-	-	$2^{107.47}$

Complexity Analysis We set the advantage of the attack as $a = 4.00$ and the number of pairs N as $2^{122.55}$. Thus, the data complexity of this attack is $2^{122.55}$. With Eq. (1), we obtain the success probability $P_S = 80.01\%$. The time complexity of the attack is composed of the time complexity in the subkey enumeration phase as in steps S0 - S20 and the time to check the remaining 66-bit value in the master key exhaustively. In this case, the total time complexity of the attack is $2^{124.45}$. Since $\mathsf{Cnt}_0[z_0]$ constitutes the largest memory, the memory complexity is roughly 2^{105}.

6 Conclusion

This paper first studies linear cryptanalyses of three AEADs with GIFT-128 as underlying primitives. We realise key-recovery attacks on GIFT-COFB, SUNDAE-GIFT, and HyENA when round-reduced versions of GIFT-128 replace the underlying primitives. Also, we check the security of GIFT-128 regarding to the linear attack. With a newly obtained 19-round linear approximation, we accomplish a 24-round linear attack on GIFT-128. Finally, we note that the attack results in this paper are far from threatening the security of GIFT-COFB, SUNDAE-GIFT, HyENA, and GIFT-128.

Acknowledgements.

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The research leading to these results has received funding from the National Natural Science Foundation of China (Grant No. 62002201, Grant No. 62032014), the National Key Research and Development Program of China (Grant No. 2018YFA0704702), the Major Scientific and Technological Innovation Project of Shandong Province, China (Grant No. 2019JZZY010133), the Major Basic Research Project of Natural Science Foundation of Shandong Province, China (Grant No. ZR202010220025), and the Qingdao Postdoctor Application Research Project (Grant No. 61580070311101).

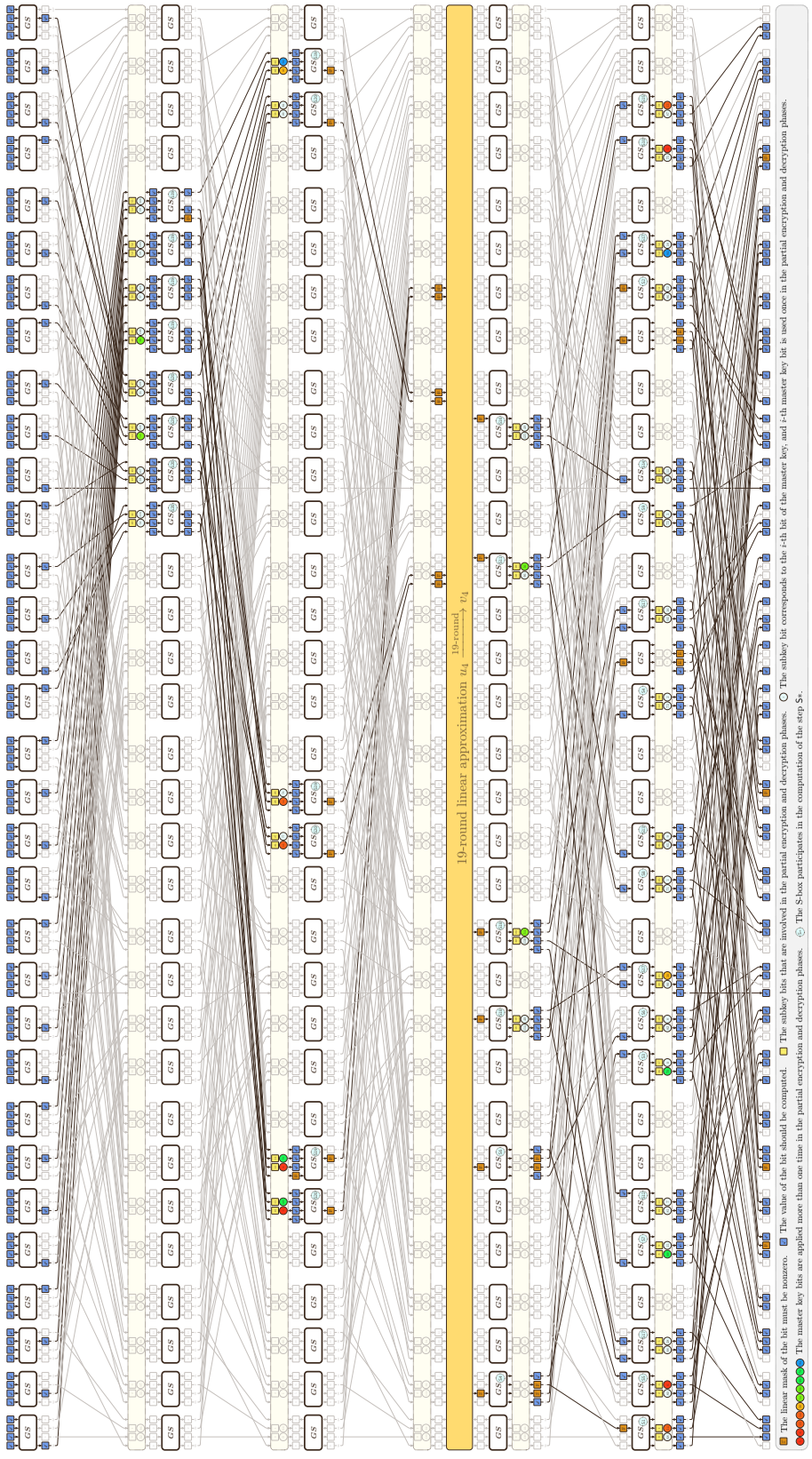


Figure 12: Key-recovery attack on 24-round GIFT-128.

References

[BBLT18] Subhadeep Banik, Andrey Bogdanov, Atul Luykx, and Elmar Tischhauser. SUNDAE: small universal deterministic authenticated encryption for the internet of things. *IACR Trans. Symmetric Cryptol.*, 2018(3):1–35, 2018.

[BBP+19] Subhadeep Banik, Andrey Bogdanov, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, Elmar Tischhauser, and Yosuke Todo. SUNDAE-GIFT. *Submission to Round*, 1, 2019.

[BCI+20] Subhadeep Banik, Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, Mridul Nandi, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT-COFB. *IACR Cryptol. ePrint Arch.*, 2020:738, 2020.

[BN17] Céline Blondeau and Kaisa Nyberg. Joint data and key distribution of simple, multiple, and multidimensional linear cryptanalysis test statistic and its impact to data complexity. *Des. Codes Cryptogr.*, 82(1-2):319–349, 2017.

[BPP+17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 321–345, 2017.

[BS90] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, pages 2–21, 1990.

[CDJN19] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, and Mridul Nandi. HYENA. *Submission to the NIST Lightweight Cryptography project*, 2019.

[CIMN17] Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, and Mridul Nandi. Blockcipher-based authenticated encryption: How small can we go? In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 277–298. Springer, 2017.

[JZZD20] Fulei Ji, Wentao Zhang, Chunning Zhou, and Tianyou Ding. Improved (related-key) differential cryptanalysis on GIFT. *IACR Cryptol. ePrint Arch.*, 2020:1242, 2020.

[LWZZ19] Lingchen Li, Wenling Wu, Yafei Zheng, and Lei Zhang. The relationship between the construction and solution of the MILP models and applications. *IACR Cryptol. ePrint Arch.*, 2019:49, 2019.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 386–397, 1993.

[Sel08] Ali Aydin Selçuk. On probability of success in linear and differential cryptanalysis. *J. Cryptol.*, 21(1):131–147, 2008.

[Sin05] Carsten Sinz. Towards an optimal CNF encoding of Boolean cardinality constraints. In *Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP 2005, Sitges, Spain, October 1-5, 2005, Proceedings*, pages 827–831, 2005.

[SNC09] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In Oliver Kullmann, editor, *Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings*, volume 5584 of *Lecture Notes in Computer Science*, pages 244–257. Springer, 2009.

[SWW18] Ling Sun, Wei Wang, and Meiqin Wang. More accurate differential properties of LED64 and Midori64. *IACR Trans. Symmetric Cryptol.*, 2018(3):93–123, 2018.

[ZDC+21] Rui Zong, Xiaoyang Dong, Huaifeng Chen, Yiyuan Luo, Si Wang, and Zheng Li. Towards key-recovery-attack friendly distinguishers: Application to GIFT-128. *IACR Transactions on Symmetric Cryptology*, 2021(1):156–184, Mar. 2021.

[ZDY19] Baoyu Zhu, Xiaoyang Dong, and Hongbo Yu. MILP-based differential attack on round-reduced GIFT. In Mitsuru Matsui, editor, *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, volume 11405 of *Lecture Notes in Computer Science*, pages 372–390. Springer, 2019.