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

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

Ling Sun[1,2,3], Wei Wang[1,3] and Meiqin Wang(✉)[1,3,4]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
[2] State Key Laboratory of Cryptology, P.O.Box 5159, Beijing, 100878, China
[3] School of Cyber Science and Technology, Shandong University, Qingdao, China
[4] Quan Cheng Shandong Laboratory, Jinan, China
lingsun@sdu.edu.cn, weiwangsdu@sdu.edu.cn, mqwang@sdu.edu.cn

Abstract. In ToSC 2021(2), Sun et al. implemented an automatic search with the Boolean satisfiability problem (SAT) method on GIFT-128 and identified a 19-round linear approximation with the expected linear potential being $2^{-117.43}$, which is utilised to launch a 24-round attack on the cipher. In this addendum, we discover a new 19-round linear approximation with a lower expected linear potential. However, in the attack, one more round can be appended after the distinguisher. As a result, we improve the previous optimal linear attack by one round and put forward a 25-round linear attack. Given that the optimal differential attack on GIFT-128, for now, covers 27-round, the resistances of the cipher against differential and linear attacks still have a 2-round gap.

Keywords: Differential cryptanalysis · Linear cryptanalysis · GIFT-128

1 Introduction

In [SWW21], Sun et al. conducted linear cryptanalyses on three Authenticated Encryptions with Associated Data (AEADs) with GIFT-128 [BPP+17] as underlying primitives and the block cipher in itself. Based on a 19-round linear approximation with the expected linear potential being $2^{-117.43}$, they launched a 24-round linear attack on GIFT-128. The dominating trail in the 19-round linear approximation is an optimal trail with the maximum correlation 2^{-59}. On the contrary, we notice that the dominating trails utilised in the attacks on three AEADs do not possess the maximum correlation, and the authors also mentioned that the optimal 10-round trails could not lead to good performances in the key-recovery attacks. Thus, we wonder about the possibility of improving the existing 24-round linear attack on GIFT-128 by a distinguisher with a lower expected linear potential.

Likewise, we exploit the automatic methods in [SWW18, SWW21] to accomplish the search for distinguishers. Since GIFT-128 achieves full diffusion after four rounds, we intend to attach four rounds and three rounds before and after the distinguisher. Firstly, the objective correlation is set as 2^{-60}, and we obtain 303616 trails in total. After that, we examine the feasibility of employing the 303616 trails to implement 26-round attacks and observe that none of the trails can actualise a valid attack for the considerable time complexity. Then, the objective correlation is reduced to 2^{-61}. The SAT solver returns 420701 trails and terminates the search for memory error. Again, none of the 420701 trails enables us to perform a 26-round attack in a reasonable amount of time. Therefore, the objective correlation is further reduced to 2^{-62}. Among the 389171 trails yielded by the solver, we choose the unique one attaining the minimum number of guessed subkey

bits as the final distinguisher in the attack, which is the dominating trail in the linear approximation in Sect. 2. However, after a careful investigation, we notice that the time complexity of the 26-round attack goes beyond 2^{128}. Thus, we finally launch a 25-round linear attack with the newly identified 19-round linear approximation. An overview of the attack result can be found in Table 1. A description of `GIFT-128` is provided in Appendix A, and more details about the method to evaluate the quality of a linear approximation are supplied in Appendix B.

Table 1: Summary of cryptanalytic results on `GIFT-128`.

Attack	Rounds	Time	Data	Memory	Success probability	Ref.
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	$2^{124.45}$	$2^{122.55}$	2^{105}	80.01%	[SWW21]
	25	$\mathbf{2^{126.77}}$	$\mathbf{2^{124.75}}$	$\mathbf{2^{96}}$	**50.00%**	**Sect. 3**
	25	$\mathbf{2^{127.77}}$	$\mathbf{2^{125.75}}$	$\mathbf{2^{96}}$	**75.00%**	**Sect. 3**

2 19-Round Linear Approximation of `GIFT-128`

The 25-round linear attack is based on a 19-round linear approximation $\alpha \xrightarrow{\text{19-round}} \beta$ with the expected linear potential being $ELP(\alpha, \beta) = 2^{-123.11}$, where

$$\begin{aligned} \alpha &= \texttt{0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x1011}, \\ \beta &= \texttt{0x0044 0x0000 0x0022 0x0000 0x0000 0x0000 0x0000 0x0000}. \end{aligned}$$

Details for the linear approximation are provided in Appendix C.

3 25-Round Linear Attack on `GIFT-128`

In this section, we propose a linear attack on 25-round `GIFT-128`. This attack utilises the 19-round linear approximation described in Sect. 2 from round 3 to 21. The key-recovery attack is demonstrated in Figure 2 of Appendix D. In the following, P is the 128-bit permutation of the PermBits operation, X^i and Y^i denote the 128-bit input and output of the SubCells operation in the i-th round ($0 \leqslant i \leqslant 25$), Z^i represents the 128-bit state $P(Y^i)$, EY^i stands for the 128-bit state $P^{-1}(X^{i+1})$, RK^i is the i-th round key, and EK^i is referred to as the equivalent round key $P^{-1}(RK^i)$. The j-th bit of a state X is denoted as $X[j]$. The attack procedure is as follows.

S0 Collect N plaintexts with corresponding ciphertexts. The time complexity of this step is equal to N 25-round encryptions.

S1 Allocate a counter $\mathsf{C}_1[z_1]$ for each of 2^{96} possible values of

$$z_1 = Z^0[48\text{-}63, 112\text{-}127] \| EY^{24}[\mathsf{Index}^{\mathsf{S1}}(EY^{24})],$$

where $\mathsf{Index}^{\mathsf{S1}}(EY^{24}) = \{8 \cdot s + b \mid s \text{ and } b \text{ are integers}, 0 \leqslant s \leqslant 15, 0 \leqslant b \leqslant 3\}$ is an index set recording the bit positions that should be memorised. Calculate the number

of plaintext-ciphertext pairs with given values z_1 and save it in $\mathsf{C}_1[z_1]$. The time complexity of this step is composed of N memory accesses, $(16 \cdot N)$ GS operations, and $\mathcal{O}(N)$ Shift and XOR operations. Following the idea in [SN14], we view one memory access to a large table as one 25-round encryption. Hence, the dominant time complexity is N memory accesses to a table with 2^{96} elements.

S2 Allocate a counter $\mathsf{C}_2[z_2]$ for each of 2^{65} values $z_2 = EY^{24}[\mathsf{Index}^{\mathsf{S1}}(EY^{24})] \| t_0$, where

$$\begin{aligned} t_0 \;=\; & \big(X^2[78] \wedge X^2[79]\big) \oplus Z^1[76] \oplus Z^1[77] \oplus X^2[78] \oplus X^2[79] \oplus \\ & \big(X^2[110] \wedge X^2[111]\big) \oplus Z^1[108] \oplus Z^1[109] \oplus X^2[110] \oplus X^2[111] \oplus \\ & \big(X^2[126] \wedge X^2[127]\big) \oplus Z^1[124] \oplus Z^1[125] \oplus X^2[126] \oplus X^2[127]. \end{aligned}$$

For each possible 19-bit subkey value $RK^0[24\text{-}31, 56\text{-}63] \| RK^1[39, 55, 63]$, compute the value of z_2 and update $\mathsf{C}_2[z_2]$ with $\mathsf{C}_2[z_2] + \mathsf{C}_1[z_1]$. Similarly to the case in S1, the dominant time complexity of this step is $2^{96} \cdot 2^{19} = 2^{115}$ memory accesses to a table with 2^{65} elements.

S3 Allocate a counter $\mathsf{C}_3[z_3]$ for each of 2^{33} possible values of

$$z_3 = EY^{23}[0\text{-}3, 8\text{-}11, 32\text{-}35, 40\text{-}43, 64\text{-}67, 72\text{-}75, 96\text{-}99, 104\text{-}107] \| t_0.$$

For each possible 24-bit subkey value

$$EK^{24}[0, 4, 5, 8, 12, 13, 16, 20, 21, 24, 28, 29, 32, 33, 37, 40, 41, 45, 48, 49, 53, 56, 57, 61],$$

compute the value of z_3 and update $\mathsf{C}_3[z_3]$ with $\mathsf{C}_3[z_3] + \mathsf{C}_2[z_2]$. The dominant time complexity of this step is $2^{65} \cdot 2^{19} \cdot 2^{24} = 2^{108}$ memory accesses to a table with 2^{33} elements.

S4 Initialise a counter Σ. For each possible 18-bit subkey value

$$EK^{22}[20, 23] \| EK^{23}[0, 1, 4, 5, 16, 17, 20, 21, 32, 33, 36, 37, 48, 49, 52, 53],$$

we compute the value of

$$\begin{aligned} t_1 \;=\; & t_0 \oplus \big(EY^{22}[8] \wedge EY^{22}[11]\big) \oplus EY^{22}[9] \oplus EY^{22}[10] \oplus EY^{22}[11] \oplus X^{22}[42] \oplus \\ & \big(EY^{22}[12] \wedge EY^{22}[15]\big) \oplus EY^{22}[13] \oplus EY^{22}[14] \oplus EY^{22}[15] \oplus X^{22}[46]. \end{aligned}$$

If t_1 equals zero, we update Σ with $\Sigma + \mathsf{C}_3[z_3]$. The time complexity of this step is composed of $(2^{33} \cdot 2^{43} \cdot 2^{18} \cdot 10)$ GS operations and $\mathcal{O}(2^{94})$ Shift, AND, and XOR operations. So, we conclude that the time complexity of this step is bounded by 2^{94} 25-round encryptions.

S5 We set the threshold as τ. The key guess will be accepted as a candidate if the condition $|\Sigma/N - 0.5| > \tau$ holds for the counter Σ. Then, all master keys compatible with the guessed 61 subkey bits are tested exhaustively against a maximum of two plaintext-ciphertext pairs.

Complexity Analysis We set the advantage of the attack as $a = 2.2$ and the number of plaintext-ciphertext pairs N as $2^{124.75}$. So, the data complexity of the attack is $2^{124.75}$. With the method from [BN17], we calculate the success probability as $P_S = 50\%$. The time complexity of steps S0 - S4 depends on the number of accesses to the memory. Following the idea in [SN14], we regard one memory access to the largest counter $\mathsf{C}_1[z_1]$ as one 25-round encryption. The time complexity of steps S0 - S4 is bounded by $(N + N + 2^{115} + 2^{108} + 2^{94})$ 25-round encryptions. The time complexity of S5 is about $2^{128} \cdot 2^{-a} \cdot \big(1 + 2^{-128}\big)$ 25-round

encryptions. Then, the time complexity of the attack is about $2^{126.77}$ 25-round encryptions. As the counters in $\mathsf{S1}$ occupy the largest memory, the memory complexity is roughly 2^{96}.

Note that the success probability of the above 25-round attack can be improved by repeating the entire work with a new group of plaintext-ciphertext pairs. Accordingly, the success probability is 75%, the data requirement is $2^{125.75}$, the time complexity is $2^{127.77}$, and the memory complexity is still 2^{96}.

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), and the Major Basic Research Project of Natural Science Foundation of Shandong Province, China (Grant No. ZR202010220025).

References

[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.

[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.

[SN14] Hadi Soleimany and Kaisa Nyberg. Zero-correlation linear cryptanalysis of reduced-round LBlock. *Des. Codes Cryptogr.*, 73(2):683–698, 2014.

[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.

[SWW21] Ling Sun, Wei Wang, and Meiqin Wang. Linear cryptanalyses of three AEADs with GIFT-128 as underlying primitives. *IACR Transactions on Symmetric Cryptology*, 2021(2):199–221, Jun. 2021.

[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 Trans. Symmetric Cryptol.*, 2021(1):156–184, 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.

A Description of GIFT-128

GIFT-128 is one version of GIFT [BPP+17] with 128-bit block size. It exploits the Substitution-Permutation Network (SPN) and iterates the round function 40 times. Denote $b_0 b_1 \cdots b_{127}$ the plaintext of the cipher, where b_0 stands for the most significant bit. Apart from the plaintext, the cipher also accepts a 128-bit key $K = k_0 \| k_1 \| \cdots \| k_7$, where k_i's are 16-bit words. The round function of GIFT-128 is composed of the following three operations.

SubCells An invertible 4-bit S-box GS, provided in the following, is applied to every nibble of the cipher state.

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

PermBits This operation maps the i-th bit of the cipher state to the $P(i)$-th bit, i.e., $b_{P(i)} \leftarrow b_i$, $i \in \{0, 1, \ldots, 127\}$. The permutation P is specified as follows.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	96	1	34	67	64	97	2	35	32	65	98	3	0	33	66	99
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	100	5	38	71	68	101	6	39	36	69	102	7	4	37	70	103
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	104	9	42	75	72	105	10	43	40	73	106	11	8	41	74	107
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	108	13	46	79	76	109	14	47	44	77	110	15	12	45	78	111
i	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
$P(i)$	112	17	50	83	80	113	18	51	48	81	114	19	16	49	82	115
i	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
$P(i)$	116	21	54	87	84	117	22	55	52	85	118	23	20	53	86	119
i	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
$P(i)$	120	25	58	91	88	121	26	59	56	89	122	27	24	57	90	123
i	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
$P(i)$	124	29	62	95	92	125	30	63	60	93	126	31	28	61	94	127

AddRoundKey This operation consists of adding the round key and the round constant. Adding the round constant does not influence the validity of the attack in this paper and is not introduced here. For the adding round key operation, we first extract a 64-bit round key RK from the key state and split it into two 32-bit words as $RK = U \| V = u_0 u_1 \cdots u_{31} \| v_0 v_1 \cdots v_{31}$. Then, U and V are XORed with the cipher state as $b_{4 \cdot i+1} \leftarrow b_{4 \cdot i+1} \oplus u_i$, $b_{4 \cdot i+2} \leftarrow b_{4 \cdot i+2} \oplus v_i$, $i \in \{0, 1, \ldots, 31\}$.

The key state is updated according to the following approach.

Key schedule In each round, the round key is first extracted as $U = k_2 \| k_3$ and $V = k_6 \| k_7$. After that, the key state is updated as follows,

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

B General Method to Select the Linear Approximation

Note that the block size n and the key size of GIFT-128 are the same. Given a linear approximation with the expected linear potential (ELP) being close to $2^{-n/2}$, the data

requirement of the corresponding linear attack should be comparable to 2^n. For this kind of attack, controlling the time complexity is an intractable task, and the critical step is to construct a counter (cf. the counter $\mathsf{C}_1[z_1]$ in Sect. 3) after gathering plaintext-ciphertext pairs. This kind of counter, denoted by $\mathsf{C}_{\mathsf{Init}}$, draws valuable information, which is used to calculate the empirical ELP of the linear approximation in the key-recovery attack, from the plaintext-ciphertext pairs. Moreover, if the size of the counter $\mathsf{C}_{\mathsf{Init}}$ is significantly smaller than the data requirement, controlling the time complexity in the subsequent subkey enumeration phase is possible. Consequently, the criterion for the linear approximation is whether we can create a counter with fewer than 2^{128} elements after obtaining plaintext-ciphertext pairs.

In this work, the search of linear trails and the sieve of candidate distinguishers are realised separately. The automatic search of linear trails is based on the method in [SWW18, SWW21], and we write another C program to accomplish the selection of distinguisher in the key-recovery attack. These programs can be found at https://github.com/SunLing134340/Addendum_GIFT-128. Note that the size of $\mathsf{C}_{\mathsf{Init}}$ is determined by the number of bits involved in the attack. Specifically, the bits are located at the output of the first SubCells operation (cf. the state Y^0 in Figure 2) and the ciphertext (cf. the state X^{25} in Figure 2). For each trail returned by the SAT solver, the selection program first tracks the bits participating in the key-recovery procedure and marks them with flags. Then, it estimates the size of $\mathsf{C}_{\mathsf{Init}}$ by counting the number of flags in states Y^0 and X^{25}. The distribution for the number of linear trails with different counter sizes is listed in Table 2. The linear approximation exploited to launch the 25-round linear attack in Sect. 3 is the unique one with $\log_2(|\mathsf{C}_{\mathsf{Init}}|) = 96$.

Table 2: Distribution for the number of linear trails.

303616 trails with correlation 2^{-60}		420701 trails with correlation 2^{-61}		389171 trails with correlation 2^{-62}	
$\log_2(\lvert\mathsf{C}_{\mathsf{Init}}\rvert)$	#{Trails}	$\log_2(\lvert\mathsf{C}_{\mathsf{Init}}\rvert)$	#{Trails}	$\log_2(\lvert\mathsf{C}_{\mathsf{Init}}\rvert)$	#{Trails}
160	1928	160	1217	96	1
176	25680	172	273	112	13
188	2048	176	20310	128	39
192	147752	188	5396	144	117
204	512	192	124083	160	1098
208	31232	204	1413	172	95
220	1536	208	49295	176	17195
224	53248	220	3858	188	2514
236	1536	224	101329	192	116587
240	36096	236	5516	204	471
252	512	240	87657	208	64737
256	1536	252	2715	220	1931
-	-	256	17639	224	106014
-	-	-	-	236	2550
-	-	-	-	240	56950
-	-	-	-	252	1068
-	-	-	-	256	17791

$\log_2(|\mathsf{C}_{\mathsf{Init}}|)$ is the binary logarithm of the size of the counter $\mathsf{C}_{\mathsf{Init}}$.
#{Trails} represents the number of trails with the specific counter size.

C 19-Round Linear Approximation

The number of characteristics belonging to this linear approximation with different correlations is illustrated in Figure 1. The dominating linear trail with correlation $c = 2^{-62}$ is given in Table 3.

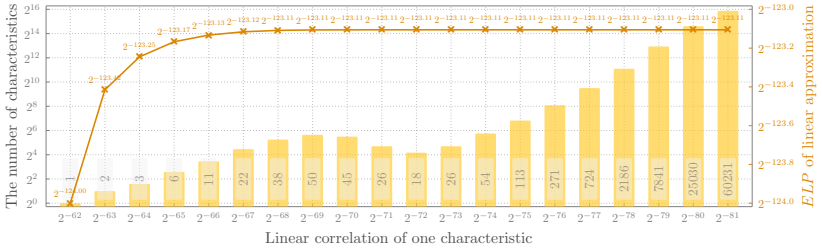

Figure 1: Distribution of characteristics belonging to the 19-round linear approximation.

Table 3: 19-round trail with correlation $c = 2^{-62}$.

State	Linear mask
$\Gamma^0_{\mathtt{in}}$	0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x1011
$\Gamma^1_{\mathtt{in}}$	0x0000 0x000c 0x0000 0x000a 0x0000 0x0000 0x0000 0x0000
$\Gamma^2_{\mathtt{in}}$	0x0000 0x0000 0x0000 0x0000 0x0202 0x0000 0x0000 0x0000
$\Gamma^3_{\mathtt{in}}$	0x0000 0xa000 0x0000 0x0000 0x0000 0xa000 0x0000 0x0000
$\Gamma^4_{\mathtt{in}}$	0x0000 0x0000 0x0200 0x0200 0x0000 0x0000 0x0000 0x0000
$\Gamma^5_{\mathtt{in}}$	0x0022 0x0000 0x0011 0x0000 0x0000 0x0000 0x0000 0x0000
$\Gamma^6_{\mathtt{in}}$	0x8080 0x0000 0x8080 0x0000 0x0000 0x0000 0x0000 0x0000
$\Gamma^7_{\mathtt{in}}$	0x5050 0x0000 0x0000 0x0000 0x5050 0x0000 0x0000 0x0000
$\Gamma^8_{\mathtt{in}}$	0x0000 0x0000 0xa000 0xa000 0x0000 0x0000 0xa000 0xa000
$\Gamma^9_{\mathtt{in}}$	0x0000 0x0000 0x0022 0x0022 0x0000 0x0000 0x0000 0x0000
$\Gamma^{10}_{\mathtt{in}}$	0x0099 0x0000 0x0000 0x0000 0x0066 0x0000 0x0000 0x0000
$\Gamma^{11}_{\mathtt{in}}$	0x0000 0x0000 0xc000 0xc000 0x0000 0x0000 0x0000 0x0000
$\Gamma^{12}_{\mathtt{in}}$	0x0000 0x0000 0x0000 0x0000 0x0011 0x0000 0x0000 0x0000
$\Gamma^{13}_{\mathtt{in}}$	0x0000 0x0000 0x0000 0xc000 0x0000 0x0000 0x0000 0x0000
$\Gamma^{14}_{\mathtt{in}}$	0x0000 0x0000 0x0002 0x0000 0x0000 0x0000 0x0000 0x0000
$\Gamma^{15}_{\mathtt{in}}$	0x0000 0x0000 0x0000 0x0000 0x0020 0x0000 0x0010 0x0000
$\Gamma^{16}_{\mathtt{in}}$	0x0000 0x0000 0x0000 0x8080 0x0000 0x0000 0x0000 0x0000
$\Gamma^{17}_{\mathtt{in}}$	0x0005 0x0000 0x0000 0x0000 0x0005 0x0000 0x0000 0x0000
$\Gamma^{18}_{\mathtt{in}}$	0x0000 0x0000 0x4000 0x4000 0x0000 0x0000 0x0000 0x0000
$\Gamma^{18}_{\mathtt{out}}$	0x0044 0x0000 0x0022 0x0000 0x0000 0x0000 0x0000 0x0000

$\Gamma^r_{\mathtt{in}}$: The input mask of the r-th round.

$\Gamma^r_{\mathtt{out}}$: The output mask of the r-th round.

D An Illustration for the 25-Round Key-Recovery Attack

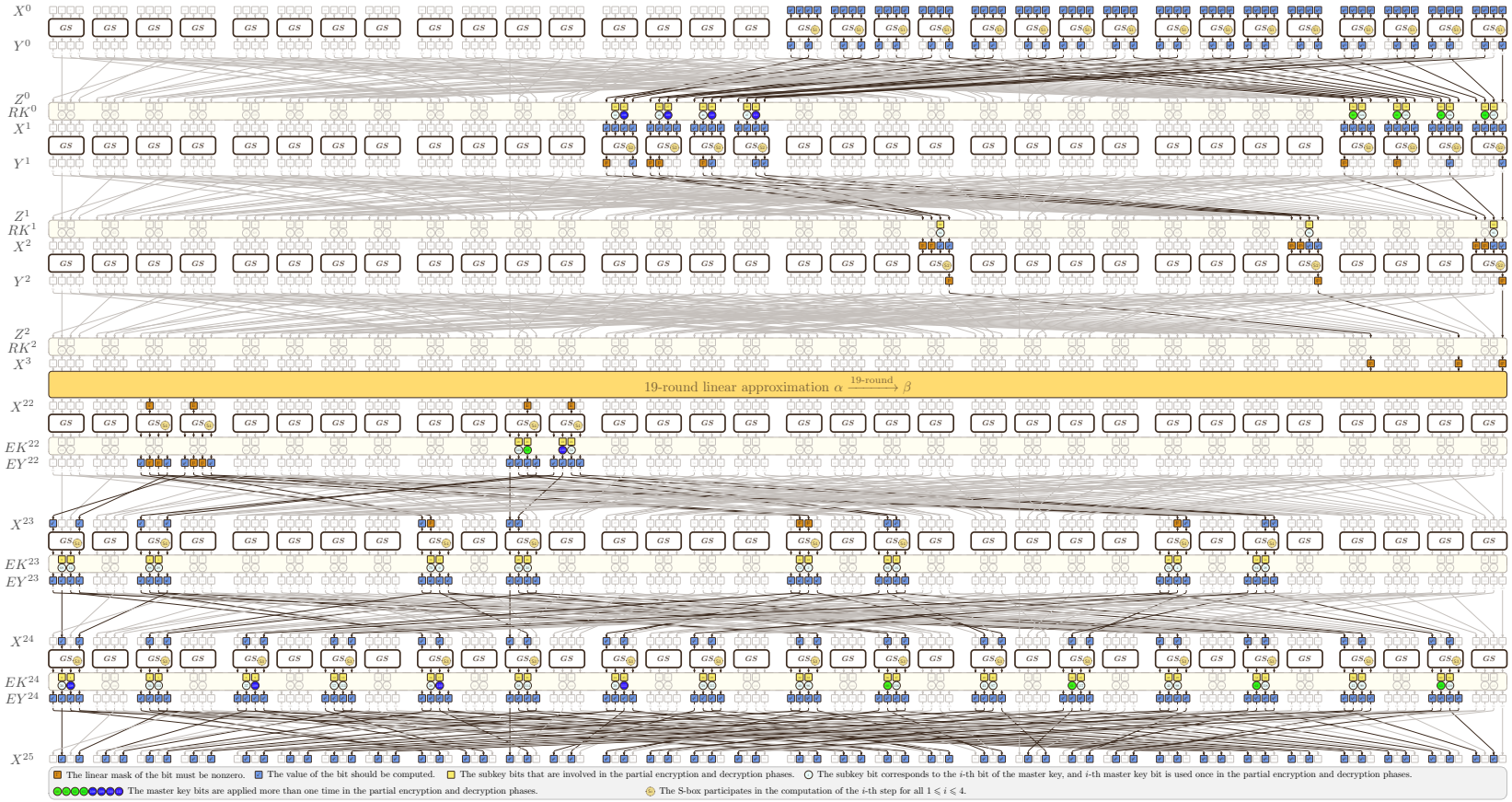

Figure 2: Key-recovery attack on 25-round GIFT-128.