Hindawi
IET Information Security
Volume 2024, Article ID 6674019, 12 pages
https://doi.org/10.1049/2024/6674019

Research Article

New Practical Attacks on GEA-1 Based on a New-Found Weakness

Zheng Wu, Lin Ding, Zhengting Li, Xinhai Wang, and Ziyu Guan

PLA SSF Information Engineering University, Zhengzhou 450001, China

Correspondence should be addressed to Lin Ding; dinglin_cipher@163.com

Received 30 July 2023; Revised 18 December 2023; Accepted 24 January 2024; Published 2 March 2024

Academic Editor: Qichun Wang

Copyright © 2024 Zheng Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

GEA-1, a proprietary stream cipher, was initially designed and used to protect against eavesdropping general packet radio service (GPRS) between the phone and the base station. Now, a variety of current mobile phones still support this standard cipher. In this paper, a structural weakness of the GEA-1 stream cipher that has not been found in previous works is discovered and analyzed. That is the probability that two different inputs of GEA-1 generate the identical keystream can be up to $2^{-7.30}$, which is quite high compared with an ideal stream cipher that generates random sequences. Based on this newfound weakness, a new practical distinguishing attack on GEA-1 is proposed, which shows that the keystreams generated by GEA-1 are far from random and can be easily distinguished with a practical time cost. After then, a new practical key recovery attack on GEA-1 is presented. It has a time complexity of $2^{21.02}$ GEA-1 encryptions and requires only seven related keys, which is much less than the existing related key attack on GEA-1. The experimental results show that GEA-1 can be broken within about 41.75 s on a common PC in the related key setting. These cryptanalytic results show that GEA-1 cannot provide enough security and should be immediately prohibited to be supported in the massive GPRS devices.

1. Introduction

General packet radio service (GPRS) is a "packet mode" wireless data system which has been standardized to operate on GSM infrastructure. GPRS was originally a standard under the European Telecommunications Standards Institute (ETSI), but was finally transferred to the 3rd Generation Partnership Project (3GPP) and released in 1998. The GPRS standard was widely used during the late 1990s and the early 2000s. To protect against eavesdropping GPRS between the base station and the phone, two proprietary stream ciphers GEA-1 and GEA-2 were initially designed and utilized for this purpose. As analyzed in [1], a variety of current mobile phones still support GEA-1. As pointed out in [1] and [2], it is a serious security problem because the support of GEA-1 by the current mobile phones makes it possible to recover a previous session key. Once the previous session key is recovered, the attacker can decrypt the previous session until the key becomes invalid.

The stream cipher GEA-1 was designed by ETSI Security Algorithms Group of Experts (SAGE) in 1998 and not made public until 2021. Only a technical report on the design process can be found at ETSI [3]. The GEA-2 stream cipher [4] was designed in 1999 as an improved variant of GEA-1. Both of them take a 64-bit key, a 32-bit initialization vector (IV) which is commonly used as a counter incremented for each frame, and a public bit *dir* which indicates the transfer direction as input, and output a keystream of 1,600 bytes for each frame. Recently, an improved variant of GEA-2 named GEA-2a was designed in [2]. The designers of GEA-2a claimed that the new variant can resist against all existing attacks and provide the 64-bit security.

1.1. Related Works. The full description of GEA-1 was first given by Beierle et al. [1] at EUROCRYPT 2021, where they presented the first publicly available cryptanalytic attack on it. Their attack on GEA-1 is based on a unusual weakness that after the linear initialization process the joint initial state of two of the three linear feedback shift registers (LFSRs) has only 2^{40} possible values (out of 2^{64}). This weakness leads to a key recovery attack on GEA-1, which has an online/offline time complexity of $2^{40}/2^{37}$ GEA-1 evaluations and a memory space of 44.5 GiB, where the time complexity unit "GEA-1

Table 1: Comparisons of our cryptanalytic results with the previous attacks on GEA-1.

Attack	Offline time complexity	Online time complexity	Memory complexity	Number of related keys	Data complexity	References
Key recovery attack	2^{37} GEA-1 evaluations	2^{40} GEA-1 evaluations	$2^{38.5}$ bits (44.5 GiB)	—	65 bits	[1]
Key recovery attack	—	2^{40} GEA-1 evaluations	2^{25} bits (4 MiB)	—	65 bits	[7]
Key recovery attack	2^{32} GEA-1 evaluations	2^{26} GEA-1 evaluations	2^{26} bits (8 MiB)	—	2^{38} bits	[2]
Key recovery attack	—	$2^{15.372}$ GEA-1 encryptions	—	50	$2^{20.668}$ bits	[2]
Distinguishing attack	—	$\mathbf{2^{11}}$ GEA-1 encryptions	—	**1**	$\mathbf{2^{16}}$ bits	Sect. 4
Key recovery attack	—	$\mathbf{2^{21.02}}$ GEA-1 encryptions	—	**7**	$\mathbf{2^{19.81}}$ bits	Sect. 5

evaluation" indicates the time cost of generating a 128-bit keystream. The attack recovers the 64-bit key of GEA-1 and requires only 65 bits of known keystream. After then, they checked how frequently the weakness occurs for randomly chosen LFSRs experimentally, the experimental results showed that the weakness in GEA-1 is unlikely to occur by chance. It indicates that the 40-bit security is intentionally designed for GEA-1 due to export regulations. Later, Beierle et al. [5] and Beierle et al. [6] took a deep insight into the design of GEA-1 and analyzed how to construct such a weak GEA-like cipher effectively.

At EUROCRYPT 2022, Amzaleg and Dinur [7] improved the attack on GEA-1 by Beierle et al. [1]. In the improved attack on GEA-1, the required memory space is decreased from 44.5 GiB to about 4 MiB, but the time complexity remains 2^{40}. The attack can be implemented in an average of 2.5 hr on a modern laptop.

Recently, new attacks on GEA-1 were proposed by Ding et al. in [2]. A key recovery attack on GEA-1 in the chosen IV setting was presented, where none of the online and offline time complexities is larger than 2^{32}. It requires a memory space of 8 MiB and 64 keystream bits for each of 2^{32} chosen IVs. Furthermore, they analyzed the slide property of GEA-1 and used it to devise a practical key recovery attack in the related key setting. Their result shows that the 64-bit secret key of GEA-1 can be successfully recovered with a time complexity of $2^{15.372}$ GEA-1 encryptions, requiring a total of $2^{20.668}$ keystream bits. The main practical obstacle is that their attack requires 50 related keys, which are too many to be available to the attacker.

1.2. Our Contributions. A structural weakness of the GEA-1 stream cipher that has not been found in previous works is discovered and analyzed in this paper. As results, new practical distinguishing attack and key recovery attack on GEA-1 are presented. The comparisons of the previous attacks with our cryptanalytic results are summarized in Table 1. In Table 1, the complexities of the previous attacks on GEA-1 are described in detail to make clear comparisons with our new attacks. Our contributions are given as follows.

(1) In this paper, we find that the initialization of the GEA-1 stream cipher is noninjective, due to the fact that the input size of GEA-1 is much larger than the size of the nonlinear feedback shift register (NFSR) used in the initialization of GEA-1. Based on this observation, the differential collision characteristic of GEA-1, i.e., there are different inputs of GEA-1 which generate the identical keystream, is explored and analyzed. The result shows that the probability that two different inputs of GEA-1 generate the identical keystream can be up to $2^{-7.30}$, which is quite high compared to an ideal stream cipher that generates random sequences.

(2) Based on the differential collision characteristic of GEA-1, a practical distinguishing attack on GEA-1 in the related key setting is proposed. The attack has a time complexity of 2^{11} GEA-1 encryptions, requiring one related key, 2^{10} chosen IVs and 2^{16} keystream bits. Note that the required 2^{16} keystream bits are generated by two keys together with 2^{10} chosen IVs, and only 32 keystream bits are needed for each key-IV pair. The success probability of this attack is almost 1. The result shows that the keystreams generated by GEA-1 are far from random and can be easily distinguished with a practical time cost.

(3) Based on the differential collision characteristic of GEA-1, a practical key recovery attack on GEA-1 in the related key setting is presented. The key recovery attack on GEA-1 has a time complexity of $2^{21.02}$ GEA-1 encryptions, requiring seven related keys and $2^{13.81}$ chosen IVs. The attack requires $2^{19.81}$ keystream bits, which are generated by two keys together with $2^{13.81}$ chosen IVs, and only 32 keystream bits are needed for each key-IV pair. The success probability of this attack is almost 1. The attack is confirmed by the experimental results, which show that GEA-1 can be broken within about 41.75 s on a common PC in the related key setting. As shown in Table 1, our practical key recovery attack on GEA-1 has a significantly lower time cost, compared with the previous attacks [1, 7]. Meanwhile, our practical key recovery attack on GEA-1 requires only seven related keys, which is much less than the existing related key attack in [2]. In contrast, if only seven related keys are used in the

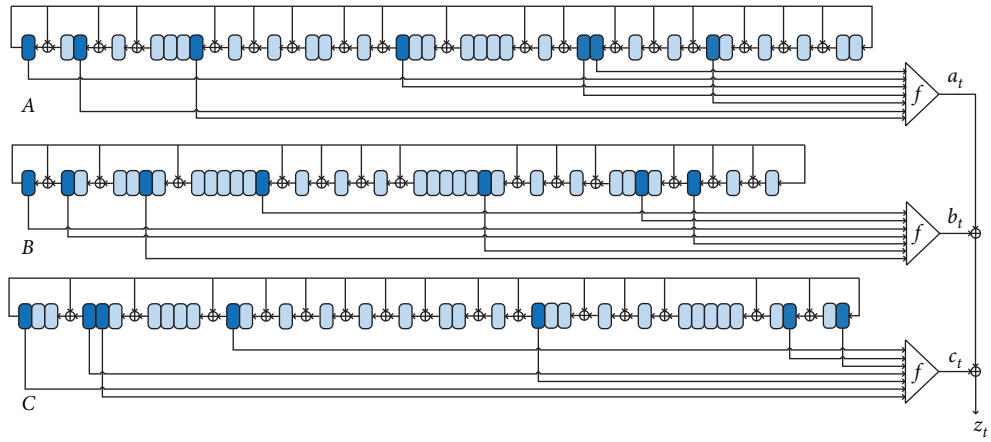

FIGURE 1: An overview of the structure of GEA-1.

attack [2], their time complexity should be about 2^{57} GEA-1 encryptions, which is much worse than our attack. Thus, our practical key recovery attack on GEA-1 is more practical than the attack [2].

1.3. Feasibility and Impact of Our Attacks. In this paper, based on the differential collision characteristic of GEA-1, new practical distinguishing attack and key recovery attack on GEA-1 are proposed. The feasibility and impact of these attacks are discussed as follows.

(1) To carry out the attacks on GEA-1 proposed in the paper, the attacker requires to collect about 32 keystream bits for one frame. As shown by Beierle et al. [1] in, collecting 65 keystream bits is feasible for an entirely passive attacker by exploiting predictable SNDCP (Subnetwork Dependent Convergence Protocol) and IP header patterns. Thus, the requirement of about 32 keystream bits for one frame in our attacks can be easily achieved.

(2) GEA-1 is a proprietary stream cipher and was initially designed and used to protect against eavesdropping GPRS between the base station and the phone. It is still supported by a variety of current mobile phones. As shown in [1], once the key is recovered, the attacker can decrypt all traffic for the complete GPRS session until the key gets invalid, which happens in the GPRS authentication and ciphering procedure triggered by the network. Thus, the practical key recovery attack on the GEA-1 stream cipher is probably a serious threat to the massive GPRS communication users.

The rest of this paper is structured as follows. A brief description of GEA-1 is given in Section 2. In Section 3, the differential collision characteristic of GEA-1 is introduced. Based on the differential collision characteristic of GEA-1, practical distinguishing and key recovery attack on GEA-1 are proposed in Sections 4 and 5, respectively. The paper is concluded in Section 6.

2. A Brief Description of GEA-1

This section gives a brief description of the GEA-1 stream cipher, for more details refer to [1, 2]. An overview of the structure of the cipher is depicted in Figure 1.

2.1. The Keystream Generator of GEA-1. The keystream generator of GEA-1 is mainly made up of three LFSRs over $\mathbb{F}_2$ denoted as A, B, and C, and a filter function denoted as f. The three LFSRs have the bit sizes of 31, 32, and 33, respectively. Let $A^{(t)} = (a_0^{(t)}, \cdots, a_{30}^{(t)})$, $B^{(t)} = (b_0^{(t)}, \cdots, b_{31}^{(t)})$, and $C^{(t)} = (c_0^{(t)}, \cdots, c_{32}^{(t)})$ denote the internal state of GEA-1 at time t, where $a_0^{(t)}$, $b_0^{(t)}$, $c_0^{(t)}$ represent the leftmost bits of three LFSRs, respectively. All of these three LFSRs work in Galois mode, and their update functions are given as follows.

LFSR A:

$$a_i^{(t+1)} = \begin{cases} a_{i+1}^{(t)} \oplus a_0^{(t)}, & \text{if } i \in T_A \\ a_{i+1}^{(t)}, & \text{if } i \in \{0, \cdots, 29\} - T_A, \\ a_0^{(t)}, & \text{if } i = 30 \end{cases} \tag{1}$$

where $T_A = \{0, 2, 3, 7, 8, 9, 11, 12, 15, 19, 20, 22, 23, 24, 26, 27, 28\}$.

LFSR B:

$$b_i^{(t+1)} = \begin{cases} b_{i+1}^{(t)} \oplus b_0^{(t)}, & \text{if } i \in T_B \\ b_{i+1}^{(t)}, & \text{if } i \in \{0, \cdots, 30\} - T_B, \\ b_0^{(t)}, & \text{if } i = 31 \end{cases} \tag{2}$$

where $T_B = \{0, 2, 6, 12, 13, 14, 15, 22, 23, 24, 28, 29, 30\}$.

LFSR C:

$$c_i^{(t+1)} = \begin{cases} c_{i+1}^{(t)} \oplus c_0^{(t)}, & \text{if } i \in T_C \\ c_{i+1}^{(t)}, & \text{if } i \in \{0, \cdots, 31\} - T_C, \\ c_0^{(t)}, & \text{if } i = 32 \end{cases} \tag{3}$$

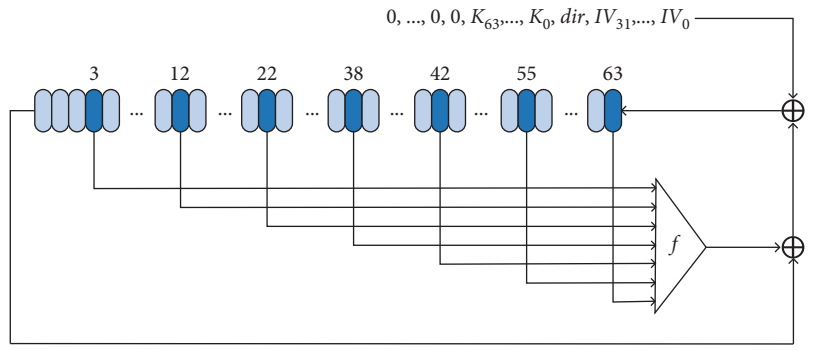

FIGURE 2: An overview of the initialization of the NFSR S.

where $T_C = \{2, 5, 9, 11, 12, 13, 14, 15, 17, 18, 21, 22, 23, 28, 30\}$.

The filter function f is a nonlinear Boolean function that takes seven bits as input and generates one bit as output. It has an algebraic degree of 4. The specification of f is given in algebraic normal form as:

$$\begin{aligned} f(x_0, \cdots, x_6) = {} & x_0x_1x_5x_6 \oplus x_0x_2x_3x_6 \oplus x_0x_2x_5x_6 \oplus x_0x_3x_5x_6 \\ & \oplus x_1x_2x_5x_6 \oplus x_1x_3x_4x_6 \oplus x_1x_3x_5x_6 \oplus x_0x_1x_3 \\ & \oplus x_0x_1x_4 \oplus x_0x_1x_6 \oplus x_0x_2x_3 \oplus x_0x_2x_4 \oplus x_0x_2x_6 \\ & \oplus x_0x_3x_5 \oplus x_1x_2x_5 \oplus x_1x_2x_6 \oplus x_1x_4x_6 \oplus x_2x_5x_6 \\ & \oplus x_0x_2 \oplus x_0x_3 \oplus x_0x_5 \oplus x_1x_3 \oplus x_1x_5 \oplus x_1x_6 \\ & \oplus x_2x_3 \oplus x_2x_5 \oplus x_2x_6 \oplus x_4x_5 \oplus x_5x_6 \oplus x_1 \oplus x_2 \\ & \oplus x_3 \oplus x_5 \end{aligned} \tag{4}$$

In the keystream generation process, one keystream bit z_t is generated per clock by:

$$z_t = a_t \oplus b_t \oplus c_t, t \geq 0, \tag{5}$$

where the three output bits a_t, b_t, c_t can be generated as follows:

$$a_t = f\left(a_{22}^{(t)}, a_0^{(t)}, a_{13}^{(t)}, a_{21}^{(t)}, a_{25}^{(t)}, a_2^{(t)}, a_7^{(t)}\right), \tag{6}$$

$$b_t = f\left(b_{12}^{(t)}, b_{27}^{(t)}, b_0^{(t)}, b_1^{(t)}, b_{29}^{(t)}, b_{21}^{(t)}, b_5^{(t)}\right), \tag{7}$$

$$c_t = f\left(c_{10}^{(t)}, c_{30}^{(t)}, c_{32}^{(t)}, c_3^{(t)}, c_{19}^{(t)}, c_0^{(t)}, c_4^{(t)}\right), \tag{8}$$

2.2. The Initialization of GEA-1. The GEA-1 stream cipher takes a 64-bit key, a 32-bit public IV, and a public bit *dir* which indicates the transfer direction as input. The initialization process of GEA-1 uses a NFSR in size of 64 (denoted as S). At the beginning of this process, the NFSR S is set to be the all-zero state. Then it is clocked 97 times, feeding in one input bit with each clock. The 97 input bits are introduced in the sequence $iv_0, \cdots, iv_{31}, dir, k_0, \cdots, k_{63}$. After loading all input bits, the NFSR S is clocked another 128 times with all zeros as input. An overview of the initialization of the NFSR S is depicted in Figure 2. As shown in Figure 2, the feedback bit of NFSR S is produced by the filter function f, XORed with one input bit and the bit that is shifted out.

After clocking the NFSR S 225 times, the content of NFSR S is denoted as $S^{(225)} = s = (s_0, \cdots, s_{63})$, which is named the *initial state* of GEA-1 in this paper. It is used as a seed for initializing the three LFSRs of GEA-1 as follows. First, three LFSRs are filled with all zeros. Then it clocks each of three LFSRs 64 times, where the feedback bit is produced by XORing one bit from the initial state and the bit that is shifted out. More specifically, the LFSRs A, B, and C insert the bits from the initial state, s starting from s_0, s_{16} and s_{32}, respectively. It should be noted that if any of the LFSRs A, B, and C end up in the all-zero state, the leftmost bit of the LFSR is forcibly set to one before producing the first keystream bit.

3. Differential Collision Characteristic of GEA-1

In this section, we will present a structural weakness in the GEA-1 stream cipher by discovering the differential collision characteristic of GEA-1. More specifically, since the input of GEA-1 (i.e., key, IV, and *dir*) has a size of 97 bits which is much larger than the size of NFSR S, then it is certain that there are different inputs of GEA-1 which generate the identical initial state after clocking the NFSR S 225 times. Once the same initial state is generated, the identical keystream will be generated. For convenience of description, a new definition is given as follows.

Definition 1. Two different inputs (K, IV, dir) and (K', IV', dir') are called an input collision pair of GEA-1, if they generate the identical keystream.

Denote by $\Delta K, \Delta IV$, and Δdir the differences of K, IV, and *dir*, respectively, and denote by $S^{(i)} = (s_0^{(i)}, \cdots, s_{63}^{(i)})$ the state of the NFSR S at clock i, $0 \leq i \leq 225$. It is easy to obtain $S^{(0)} = 0$, since the NFSR S is set to be all-zero state at the beginning of initialization. After then, the NFSR S is updated 225 times as follows.

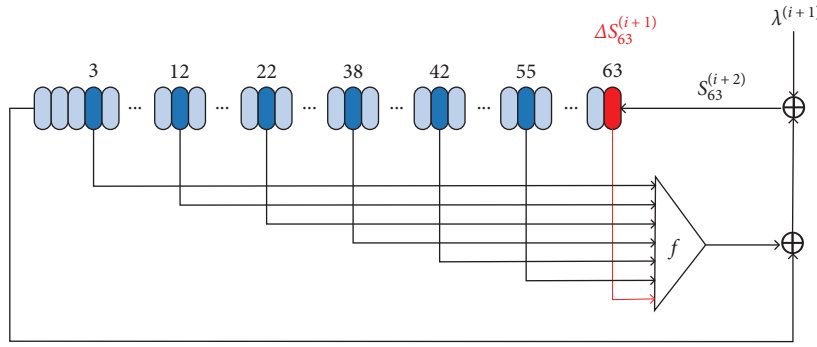

FIGURE 3: The differential path to generate the identical feedback bit $s_{63}^{(i+2)}$.

For $0 \le i \le 224$,

$$\begin{aligned} -s_j^{(i+1)} &= s_{j+1}^{(i)}, 0 \le j \le 62; \\ -s_{63}^{(i+1)} &= f\left(s_3^{(i)}, s_{12}^{(i)}, s_{22}^{(i)}, s_{38}^{(i)}, s_{42}^{(i)}, s_{55}^{(i)}, s_{63}^{(i)}\right) \oplus s_0^{(i)} \oplus \lambda^{(i)}. \end{aligned} \tag{9}$$

Where

$$\left(\lambda^{(0)}, \cdots, \lambda^{(224)}\right) = (iv_0, \cdots, iv_{31}, dir, k_0, \cdots, k_{63}, 0, \cdots, 0). \tag{10}$$

Now, we introduce an effective method of searching for input collision pairs of GEA-1. To achieve this goal, we have found a kind of differential paths for GEA-1. Firstly, we introduce the difference into the input bit $\lambda^{(i)}$, i.e., $\Delta\lambda^{(i)} = 1$, and then the difference will appear in the updated bit $s_{63}^{(i+1)}$ after one clock, i.e., $\Delta s_{63}^{(i+1)} = 1$. Here, the integral parameter i should satisfy $0 \le i \le 96$ and $i \ne 32$, since the input bit $\lambda^{(i)}$ is equal to the public bit dir for $i = 32$ and fixed to be zero for $97 \le i \le 224$, and cannot contain any difference. It is easy to see that there are nine input bits (i.e., $s_3^{(i+1)}, s_{12}^{(i+1)}, s_{22}^{(i+1)}, s_{38}^{(i+1)}, s_{42}^{(i+1)}, s_{55}^{(i+1)}, s_{63}^{(i+1)}, s_0^{(i+1)}, \lambda^{(i+1)}$) to generate the feedback bit $s_{63}^{(i+2)}$. As shown in Figure 3, if the difference only appears in the state bit $s_{63}^{(i+1)}$ and the other eight state bits do not contain any difference, it is possible that the difference $\Delta s_{63}^{(i+1)}$ disappears in the feedback bit $s_{63}^{(i+2)}$, i.e., $\Delta s_{63}^{(i+2)} = 0$. Thus, when the following condition denoted as $C1$ is satisfied, $\Delta s_{63}^{(i+2)} = 0$ holds.

$$\Delta f\left(\begin{array}{c} \Delta s_3^{(i+1)} = \Delta s_{12}^{(i+1)} = \Delta s_{22}^{(i+1)} = \Delta s_{38}^{(i+1)} = \\ \Delta s_{42}^{(i+1)} = \Delta s_{55}^{(i+1)} = 0, \Delta s_{63}^{(i+1)} = 1 \end{array}\right) = 0, \tag{11}$$

where $\Delta f(\cdot)$ denotes the output difference of the nonlinear function f.

After then, by shifting more, the bit $s_{63}^{(i+1)}$ is updated to be $s_{55}^{(i+9)}, s_{42}^{(i+22)}, s_{38}^{(i+26)}, s_{22}^{(i+42)}, s_{12}^{(i+52)}$, and $s_3^{(i+61)}$ after 8, 21, 25, 41, 51, and 60 clocks, respectively. By simply repeating the process above, we know that it is possible to generate the identical feedback bit if the difference only appears in one of all input bits of the nonlinear function f. That is, when the following six conditions denoted as $C2, \cdots, C7$ are satisfied, $\Delta s_{63}^{(i+10)} = \Delta s_{63}^{(i+23)} = \Delta s_{63}^{(i+27)} = \Delta s_{63}^{(i+43)} = \Delta s_{63}^{(i+53)} = \Delta s_{63}^{(i+62)} = 0$ holds.

$$\Delta f\left(\begin{array}{c} \Delta s_3^{(i+9)} = \Delta s_{12}^{(i+9)} = \Delta s_{22}^{(i+9)} = \Delta s_{38}^{(i+9)} = \\ \Delta s_{42}^{(i+9)} = 0, \Delta s_{55}^{(i+9)} = 1, \Delta s_{63}^{(i+9)} = 0 \end{array}\right) = 0, \tag{12}$$

$$\Delta f\left(\begin{array}{c} \Delta s_3^{(i+22)} = \Delta s_{12}^{(i+22)} = \Delta s_{22}^{(i+22)} = \Delta s_{38}^{(i+22)} = 0, \\ \Delta s_{42}^{(i+22)} = 1, \Delta s_{55}^{(i+22)} = \Delta s_{63}^{(i+22)} = 0 \end{array}\right) = 0, \tag{13}$$

$$\Delta f\left(\begin{array}{c} \Delta s_3^{(i+26)} = \Delta s_{12}^{(i+26)} = \Delta s_{22}^{(i+26)} = 0, \Delta s_{38}^{(i+26)} = 1, \\ \Delta s_{42}^{(i+26)} = \Delta s_{55}^{(i+26)} = \Delta s_{63}^{(i+26)} = 0 \end{array}\right) = 0, \tag{14}$$

$$\Delta f\left(\begin{array}{c} \Delta s_3^{(i+42)} = \Delta s_{12}^{(i+42)} = 0, \Delta s_{22}^{(i+42)} = 1, \Delta s_{38}^{(i+42)} = \\ \Delta s_{42}^{(i+42)} = \Delta s_{55}^{(i+42)} = \Delta s_{63}^{(i+42)} = 0 \end{array}\right) = 0, \tag{15}$$

$$\Delta f\left(\begin{array}{c} \Delta s_3^{(i+52)} = 0, \Delta s_{12}^{(i+52)} = 1, \Delta s_{22}^{(i+52)} = \Delta s_{38}^{(i+52)} = \\ \Delta s_{42}^{(i+52)} = \Delta s_{55}^{(i+52)} = \Delta s_{63}^{(i+52)} = 0 \end{array}\right) = 0, \tag{16}$$

$$\Delta f\left(\begin{array}{c} \Delta s_3^{(i+61)} = 1, \Delta s_{12}^{(i+61)} = \Delta s_{22}^{(i+61)} = \Delta s_{38}^{(i+61)} = \\ \Delta s_{42}^{(i+61)} = \Delta s_{55}^{(i+61)} = \Delta s_{63}^{(i+61)} = 0 \end{array}\right) = 0. \tag{17}$$

Finally, after 63 clocks, the bit $s_{63}^{(i+1)}$ is updated to be $s_0^{(i+64)}$ by shifting. As shown in Figure 4, since the seven input bits of the nonlinear function f do not contain any difference, we have to introduce another difference into the input bit $\lambda^{(i+64)}$ (i.e., $\Delta\lambda^{(i+64)} = 1$) to generate the identical feedback bit $s_{63}^{(i+65)}$. Since the input bit $\lambda^{(i)}$ is fixed to be zero for $97 \le i \le 224$ and cannot contain any difference, then it has $i + 64 \le 96$ (and thus $i \le 32$) to generate the identical feedback bit $s_{63}^{(i+65)}$. Therefore, when the input difference $\Delta\lambda^{(i+64)} = 1$ is satisfied, $\Delta s_{63}^{(i+65)} = 0$ holds directly.

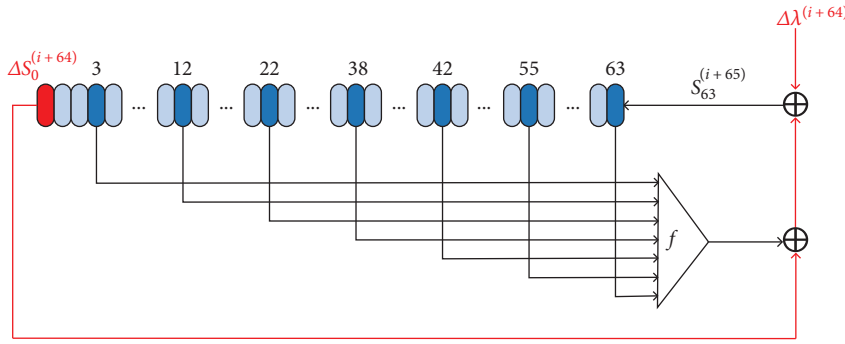

FIGURE 4: The differential path to generate identical $s_{63}^{(i+65)}$.

TABLE 2: Kind of differential paths for GEA-1.

t-th Clock	$\Delta s_0^{(t)}$	⋯	$\Delta s_3^{(t)}$	⋯	$\Delta s_{12}^{(t)}$	⋯	$\Delta s_{22}^{(t)}$	⋯	$\Delta s_{38}^{(t)}$	⋯	$\Delta s_{42}^{(t)}$	⋯	$\Delta s_{55}^{(t)}$	⋯	$\Delta s_{63}^{(t)}$
$t=0$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t=i$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$t=i+1$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1*
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t=i+9$	0	0	0	0	0	0	0	0	0	0	0	0	*1*	0	0
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t=i+22$	0	0	0	0	0	0	0	0	0	0	*1*	0	0	0	0
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t=i+26$	0	0	0	0	0	0	0	0	*1*	0	0	0	0	0	0
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t=i+42$	0	0	0	0	0	0	*1*	0	0	0	0	0	0	0	0
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t=i+52$	0	0	0	0	*1*	0	0	0	0	0	0	0	0	0	0
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t=i+61$	0	0	*1*	0	0	0	0	0	0	0	0	0	0	0	0
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t=i+64$	*1*	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$t=i+65$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Up to now, we have found a kind of differential paths for GEA-1, which is detailed in Table 2. As described in Table 2, the first difference introduced into the NFSR is at the $(i+1)$-th clock, due to the difference $\Delta\lambda^{(i)}$. From the $(i+1)$-th to the $(i+64)$-th clock, the weight of the difference of the NFSR is always to be one, i.e., $\#\{\Delta s_l^{(t)}=1|l=0,1,\cdots,63\}=1$ for $i+1\le t\le i+64$. The difference $\Delta s_0^{(i+64)}$ disappears at the $(i+65)$-th clock, due to the introduced difference $\Delta\lambda^{(i+64)}$.

Clearly, to construct the differential paths for GEA-1 in Table 2, the input difference should satisfy a condition, called **R1**, described as follows.

$$\Delta\lambda^{(j)}=\begin{cases}1, & if\ j\in\{i,i+64\}\\ 0, & if\ j\in\{0,\cdots,96\}-\{i,i+64\}\end{cases}, \tag{18}$$

where the parameter i is an integer satisfying $0\le i\le 31$.

When the condition **R1** is satisfied, the kind of differential paths for GEA-1 in Table 2 holds when the seven conditions $C1,\cdots,C7$ hold simultaneously. Once the seven conditions $C1,\cdots,C7$ hold simultaneously, two different input pairs will generate the identical state $S^{(i+65)}$, and then the identical initial state and keystream will be generated. Now, an observation for GEA-1 has been obtained as follows.

Observation 1. *When the input difference is chosen to satisfy the condition **R1** and the seven conditions $C1,\cdots,C7$ hold simultaneously, two different inputs of GEA-1 are an input collision pair.*

Now, we calculate the probability that the seven conditions $C1,\cdots,C7$ hold simultaneously. First, for each condition, we make an assumption that the seven input bits of the nonlinear function f are independent and identically distributed. Under this assumption, we can easily calculate the probabilities as follows.

Table 3: The experimental probabilities that the seven conditions $C1, \cdots, C7$ hold simultaneously.

i	The total number of input collision pairs	The average number of input collision pairs per key	The experimental probability	i	The total number of input collision pairs	The average number of input collision pairs per key	The experimental probability
16	50,792	507.92	$2^{-7.01}$	24	52,519	525.19	$2^{-6.96}$
17	48,962	489.62	$2^{-7.06}$	25	48,948	489.48	$2^{-7.06}$
18	50,882	508.82	$2^{-7.01}$	26	50,986	509.86	$2^{-7.01}$
19	50,930	509.30	$2^{-7.01}$	27	50,077	500.77	$2^{-7.03}$
20	49,386	493.86	$2^{-7.05}$	28	52,360	523.60	$2^{-6.97}$
21	50,594	505.94	$2^{-7.02}$	29	51,627	516.27	$2^{-6.99}$
22	50,881	508.81	$2^{-7.01}$	30	50,807	508.07	$2^{-7.01}$
23	53,792	537.92	$2^{-6.93}$	31	50,915	509.15	$2^{-7.01}$

$$Pr(C1) = 2^{-1.30}, Pr(C2) = 2^{-1}, Pr(C3) = 2^{-1}, \; Pr(C4) = 2^{-1}, Pr(C5) = 2^{-1}, Pr(C6) = 2^{-1}, Pr(C7) = 2^{-1}. \tag{19}$$

After then, we make another assumption that the seven conditions $C1, \cdots, C7$ are independent, and then we can calculate the probability that the seven conditions $C1, \cdots, C7$ hold simultaneously as follows.

$$Pr(C1, C2, C3, C4, C5, C6, C7) = \prod_{j=1}^{7} Pr(Cj) = 2^{-7.30}. \tag{20}$$

To verify the theoretical probability calculated above, we have made an experiment. In this experiment, we first randomly select 100 keys and 2^{16} IVs to form 100×2^{16} different inputs. Under the condition **R1**, we can obtain another 100×2^{16} different inputs. After then, for each of these 100 keys, we count the number of IVs such that (K, IV, dir) and (K', IV', dir') are an input collision pair of GEA-1. Then we obtain the following results:

(1) When the integer i satisfies $0 \le i \le 15$, there does not exist any IV such that (K, IV, dir) and (K', IV', dir') are an input collision pair of GEA-1.

(2) When the integer i satisfies $16 \le i \le 31$, for each of these 100 keys, there are some IVs such that (K, IV, dir) and (K', IV', dir') are an input collision pair of GEA-1. The experimental results are listed in Table 3. Some input collision pairs found in this experiment are listed in Table 4. Take $i = 16$ for example. The total number of input collision pairs we have found is 50,792, and thus the average number of input collision pairs per key is $50792/100 = 507.92$. This means that the experimental probability that the seven conditions $C1, \cdots, C7$ hold simultaneously is about $507.92/2^{16} = 2^{-7.01}$, which is very close to the theoretical result $2^{-7.30}$.

As shown in Table 3, when $16 \le i \le 31$, the experimental probabilities are all quite close to the theoretical result $2^{-7.30}$, and thus the kind of differential paths in Table 2 exists and input collision pairs of GEA-1 can be found effectively. However, when $0 \le i \le 15$, none of input collision pairs is found among 100 keys and 2^{16} IVs. In order to validate this, we make a supplementary experiment by increasing the number of IVs to 2^{24}, but the result remains the same and none of input collision pairs of GEA-1 is found. This is probably due to that there are contradictions in the state of the NFSR S such that the seven conditions $C1, \cdots, C7$ cannot hold simultaneously when $0 \le i \le 15$.

According to the experimental results, we update the condition **R1** to **R2**. The new condition **R2** is the same with **R1**, except that the parameter i in **R2** is limited to satisfy $16 \le i \le 31$, instead of $0 \le i \le 31$ in **R1**. Now, a new observation for GEA-1 can be given as follows.

Observation 2. *When the input difference is chosen to satisfy the condition **R2**, two different inputs of GEA-1 are an input collision pair with a probability of about* $2^{-7.30}$.

Therefore, the probability that two different inputs of GEA-1 satisfying the condition **R2** generate the identical keystream is about $2^{-7.30}$, which is quite high compared to an ideal stream cipher that generates random sequences. Using this weakness, the attacker can easily distinguish the keystream generated by GEA-1 from the random sequence.

4. Practical Distinguishing Attacks on GEA-1

Based on Observation 2, this section aims at proposing practical distinguishing attacks on GEA-1 in the related key chosen IV setting, whose goal is to distinguish the keystream from a truly random sequence. It should be noted that the related key chosen IV setting is a common attack setting in cryptanalysis of stream ciphers and has been utilized in many cryptanalytic works, e.g., Grain-like [8–10], WG-8 [11], SNOW 3G [12], GEA-1 [2], and GEA-2 [2]. In this attack setting, the chosen IV setting can be directly satisfied. Though the related key setting is considered to be a more unrealistic scenario than the chosen IV setting, it may still be utilized to break some cryptosystems. For instance, related key weaknesses of the RC4 stream cipher led to a practical attack on the WEP protocol [13]. In the related key setting, the attacker is allowed to use two different keys that have a

Table 4: Some input collision pairs of GEA-1.

i	The input collision pair
16	K:01111000000010001100001111100001101010011100000*1*01111110100001000
	K':01111000000010001100001111100001101010011100000*0*01111110100001000
	IV:11111110000011001111110001110010
	IV':11111110000011000111110001110010
	dir:0
	dir':0
	Keystream:10010101111010110000111001000111000010010100110011110000101100000⋯
24	K:01111000000010001100001111100001101010011100000101111101*0*0001000
	K':01111000000010001100001111100001101010011100000101111100*0*00001000
	IV:00110101110111110100101100100111
	IV':00110101110111110100101110100111
	dir:1
	dir':1
	Keystream:01100010111010111010011110001100001000110001111010100011101101 0⋯
31	K:01111000000010001100001111100001101010011100000101111101000010*0*0
	K':01111000000010001100001111100001101010011100000101111101000010*1*0
	IV:01000110010110000100100010100000
	IV':01000110010110000100100010100001
	dir:0
	dir':0
	Keystream:01111001110010100010110110101000011101100101001010101101010111 10⋯

known relationship between them, but he does not know the values of these two keys [14–16]. Thus, the condition **R2** can be satisfied directly in the related key chosen IV setting.

4.1. A Practical Distinguishing Attack on GEA-1. Based on Observation 2, a practical distinguishing attack on GEA-1 will be proposed in this subsection. The attack can be described as an algorithm as follows.

Clearly, there are two types of errors when Algorithm 1 makes such a judgment. The first is that the algorithm judges the output sequences are keystreams generated by GEA-1 but they are in fact random sequences. The probability of this error can be calculated as $1-(1-2^{-L})^m$. The second is that the algorithm judges the output sequences are random but they are in fact keystreams generated by GEA-1. The probability of this error can be calculated as $(1-2^{-7.30})^m$. Thus, the probability that Algorithm 1 succeeds can be calculated as:

$$\begin{aligned} p &= 1-(1-(1-2^{-L})^m)-(1-2^{-7.30})^m \\ &= (1-2^{-L})^m-(1-2^{-7.30})^m. \end{aligned} \tag{21}$$

It is easy to see that there is tradeoff between the number of chosen IVs used in Algorithm 1 and the success probability of this algorithm. The values of m and L can be chosen by the attacker to make a reasonable tradeoff between them. Here, to achieve a high success probability, we set $m=2^{12}$ and $L=32$ for GEA-1, and then we have $p=(1-2^{-32})^{2^{12}}-(1-2^{-7.30})^{2^{12}}\approx 1$. To validate this, we make an experiment by randomly choosing 1,000 keys, and execute Algorithm 1 once for each key. The result shows that Algorithm 1 always succeeds giving the right output. Thus, a distinguishing attack on GEA-1 with a time complexity of $2\cdot m=2^{13}$ GEA-1 encryptions has been proposed, requiring one related key and $m=2^{12}$ chosen IVs. Each key and IV pair only needs to generate $L=32$ keystream bits, which leads to a total data complexity of $2\cdot 2^{12}\cdot 32=2^{18}$ keystream bits. The attack has a success probability of almost 1.

4.2. Improved Distinguishing Attacks on GEA-1. In fact, the practical distinguishing attack above can be further improved, if we take a deeper look at the seven conditions $C1,\cdots,C7$. This subsection aims at presenting improved distinguishing attacks on GEA-1.

At the beginning of initialization of GEA-1, the NFSR S is filled with all zeros, and then it is clocked 97 times, feeding in one input bit for each time. The 97 input bits are loaded in the sequence $iv_0,\cdots,iv_{31},dir,k_0,\cdots,k_{63}$. Since both of IV and dir are public to the attacker, the state of $S^{(t)}$ for $0\le t\le 33$ can be naturally known to the attacker. Take $i=16$ for example. The first two conditions $C1$ and $C2$ directly hold in the chosen IV setting, as no key bit is involved in these two conditions. This means that when the remaining five conditions $C3,\cdots,C7$ hold simultaneously, the two different input pairs are an input collision pair. The probability that two different inputs (K,IV,dir) and (K',IV',dir') are an input collision pair becomes 2^{-5}, which is larger than $2^{-7.30}$ by a factor of $2^{2.30}$. This enables us to propose a better distinguishing attack on GEA-1 that has a time complexity of $2\cdot m=2\cdot 2^{10}=2^{11}$ GEA-1 encryptions, by choosing $m=2^{10}$ which is large enough. This improved attack requires only $m=2^{10}$ chosen IVs and has a total data complexity of $2\cdot 2^{10}\cdot 32=2^{16}$

1. Randomly choose m IVs, i.e., $IV_1, \cdots, IV_m$.

2. For h from 1 to m, do the followings:

2.1 Generate an output sequence with length L using the input (K, IV_h, dir).

2.2 Generate another output sequence with length L using the input (K', IV_h', dir'), where the input difference between (K, IV_h, dir) and (K', IV_h', dir') satisfies the condition **R2**.

2.3 If the two output sequences are identical, judge that they are keystreams generated by GEA-1. Otherwise, return to Step 2 and try the next IV.

3. If no identical output sequence is found after checking all m IVs, judge that they are random sequences.

Algorithm 1: Distinguishing attack on GEA-1.

Table 5: The key bits involved in the conditions $C1, \cdots, C7$ for $16 \le i \le 31$.

i	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$
16	None	None	$k_0, \cdots, k_4$	$k_0, \cdots, k_8$	$k_0, \cdots, k_{24}$	$k_0, \cdots, k_{34}$	$k_0, \cdots, k_{43}$
17	None	None	$k_0, \cdots, k_5$	$k_0, \cdots, k_9$	$k_0, \cdots, k_{25}$	$k_0, \cdots, k_{35}$	$k_0, \cdots, k_{44}$
18	None	None	$k_0, \cdots, k_6$	$k_0, \cdots, k_{10}$	$k_0, \cdots, k_{26}$	$k_0, \cdots, k_{36}$	$k_0, \cdots, k_{45}$
19	None	None	$k_0, \cdots, k_7$	$k_0, \cdots, k_{11}$	$k_0, \cdots, k_{27}$	$k_0, \cdots, k_{37}$	$k_0, \cdots, k_{46}$
20	None	None	$k_0, \cdots, k_8$	$k_0, \cdots, k_{12}$	$k_0, \cdots, k_{28}$	$k_0, \cdots, k_{38}$	$k_0, \cdots, k_{47}$
21	None	None	$k_0, \cdots, k_9$	$k_0, \cdots, k_{13}$	$k_0, \cdots, k_{29}$	$k_0, \cdots, k_{39}$	$k_0, \cdots, k_{48}$
22	None	None	$k_0, \cdots, k_{10}$	$k_0, \cdots, k_{14}$	$k_0, \cdots, k_{30}$	$k_0, \cdots, k_{40}$	$k_0, \cdots, k_{49}$
23	None	None	$k_0, \cdots, k_{11}$	$k_0, \cdots, k_{15}$	$k_0, \cdots, k_{31}$	$k_0, \cdots, k_{41}$	$k_0, \cdots, k_{50}$
24	None	None	$k_0, \cdots, k_{12}$	$k_0, \cdots, k_{16}$	$k_0, \cdots, k_{32}$	$k_0, \cdots, k_{42}$	$k_0, \cdots, k_{51}$
25	None	k_0	$k_0, \cdots, k_{13}$	$k_0, \cdots, k_{17}$	$k_0, \cdots, k_{33}$	$k_0, \cdots, k_{43}$	$k_0, \cdots, k_{52}$
26	None	k_0, k_1	$k_0, \cdots, k_{14}$	$k_0, \cdots, k_{18}$	$k_0, \cdots, k_{34}$	$k_0, \cdots, k_{44}$	$k_0, \cdots, k_{53}$
27	None	k_0, k_1, k_2	$k_0, \cdots, k_{15}$	$k_0, \cdots, k_{19}$	$k_0, \cdots, k_{35}$	$k_0, \cdots, k_{45}$	$k_0, \cdots, k_{54}$
28	None	$k_0, \cdots, k_3$	$k_0, \cdots, k_{16}$	$k_0, \cdots, k_{20}$	$k_0, \cdots, k_{36}$	$k_0, \cdots, k_{46}$	$k_0, \cdots, k_{55}$
29	None	$k_0, \cdots, k_4$	$k_0, \cdots, k_{17}$	$k_0, \cdots, k_{21}$	$k_0, \cdots, k_{37}$	$k_0, \cdots, k_{47}$	$k_0, \cdots, k_{56}$
30	None	$k_0, \cdots, k_5$	$k_0, \cdots, k_{18}$	$k_0, \cdots, k_{22}$	$k_0, \cdots, k_{38}$	$k_0, \cdots, k_{48}$	$k_0, \cdots, k_{57}$
31	None	$k_0, \cdots, k_6$	$k_0, \cdots, k_{19}$	$k_0, \cdots, k_{23}$	$k_0, \cdots, k_{39}$	$k_0, \cdots, k_{49}$	$k_0, \cdots, k_{58}$

keystream bits, while its success probability is still almost 1. Clearly, this improved attack always holds for $16 \le i \le 24$. However, when $25 \le i \le 31$, only the first condition $C1$ can be directly satisfied in the chosen IV setting. Similarly, the probability that two different inputs (K, IV, dir) and (K', IV', dir') are an input collision pair becomes 2^{-6}, and thus we can obtain another distinguishing attack on GEA-1 that has a time complexity of $2 \cdot m = 2 \cdot 2^{11} = 2^{12}$ GEA-1 encryptions, requiring $m = 2^{11}$ chosen IVs and 2^{17} keystream bits. The success probability is still almost 1.

5. Practical Key Recovery Attacks on GEA-1

This section focuses on how to recover the secret key of GEA-1 based on the differential collision distinguishers constructed above. To effectively recover the secret key of GEA-1, we have analyzed which key bits are involved in the conditions $C1, \cdots, C7$ for $16 \le i \le 31$. The obtained results are listed in Table 5.

5.1. Key Recovery Attacks on GEA-1 Using One Related Key. In this subsection, some key recovery attacks on GEA-1 using one related key will be proposed. For convenience of description, a key recovery attack on GEA-1 when choosing $i = 16$ is presented as follows.

By Algorithm 1, we can find an input collision pair such that the seven conditions $C1, \cdots, C7$ are simultaneously satisfied. However, no key bit can be recovered by using the first two conditions, i.e., $C1$ and $C2$, since no key bit is involved in these two conditions. As for the remaining five conditions $C3, \cdots, C7$, as listed in Table 5, there are a total of 44 key bits (i.e., $k_0, \cdots, k_{43}$) are involved in these five conditions. About five key bits can be recovered theoretically by using these five conditions, as the probability that these five conditions hold simultaneously is about 2^{-5}. Thus, the attacker can make an exhaustive search of these 44 key bits, and then check whether these five conditions hold simultaneously. This enables to reduce the number of possible guesses from 2^{44} to about$2^{44} \cdot 2^{-5} = 2^{39}$. Then, the attacker can make an exhaustive search of the obtained 2^{39} possible guesses together with the remaining $64 - 44 = 20$ key bits (i.e., $k_{44}, \cdots, k_{63}$) to recover the 64-bit key. Since the maximum number of possible guesses is no more than $2^{39} \cdot 2^{20} = 2^{59}$ in this whole key recovery process, the time complexity of this key recovery process is at most 2^{59}. Considering the cost of finding an input collision pair, the key recovery attack on GEA-1 has a total time complexity of $2^{11} + 2^{59} \approx 2^{59}$ GEA-1 encryptions, requiring one related key, 2^{10} chosen IVs and 2^{16} keystream bits. The success probability of the key recovery attack is

Table 6: The detailed process of recovering the key of GEA-1 using seven related keys.

j-th Step	The key bits to be guessed	The number of possible guesses	The condition(s) to be checked	The number of possible guesses after checking the condition(s)
1	k_0	2	$C2$ with $i=25$	1
2	k_1	2	$C2$ with $i=26$	1
3	k_2	2	$C2$ with $i=27$	1
4	k_3	2	$C2$ with $i=28$	1
5	k_4	2	$C2$ with $i=29$	1
6	k_5	2	$C2$ with $i=30$	1
7	k_6	2	$C2$ with $i=31$	1
8	$k_7, \cdots, k_{13}$	2^7	$C3$ with $i=25$	2^6
9	k_{14}	2^7	$C3$ with $i=26$	2^6
10	k_{15}	2^7	$C3$ with $i=27$	2^6
11	k_{16}	2^7	$C3$ with $i=28$	2^6
12	k_{17}	2^7	$C3$ with $i=29$, $C4$ with $i=25$	2^5
13	k_{18}	2^6	$C3$ with $i=30$, $C4$ with $i=26$	2^4
14	k_{19}	2^5	$C3$ with $i=31$, $C4$ with $i=27$	2^3
15	k_{20}	2^4	$C4$ with $i=28$	2^3
16	k_{21}	2^4	$C4$ with $i=29$	2^3
17	k_{22}	2^4	$C4$ with $i=30$	2^3
18	k_{23}	2^4	$C4$ with $i=31$	2^3
19	$k_{24}, \cdots, k_{33}$	2^{13}	$C5$ with $i=25$	2^{12}
20	k_{34}	2^{13}	$C5$ with $i=26$	2^{12}
21	k_{35}	2^{13}	$C5$ with $i=27$	2^{12}
22	k_{36}	2^{13}	$C5$ with $i=28$	2^{12}
23	k_{37}	2^{13}	$C5$ with $i=29$	2^{12}
24	k_{38}	2^{13}	$C5$ with $i=30$	2^{12}
25	k_{39}	2^{13}	$C5$ with $i=31$	2^{12}
26	$k_{40}, \cdots, k_{43}$	2^{16}	$C6$ with $i=25$	2^{15}
27	k_{44}	2^{16}	$C6$ with $i=26$	2^{15}
28	k_{45}	2^{16}	$C6$ with $i=27$	2^{15}
29	k_{46}	2^{16}	$C6$ with $i=28$	2^{15}
30	k_{47}	2^{16}	$C6$ with $i=29$	2^{15}
31	k_{48}	2^{16}	$C6$ with $i=30$	2^{15}
32	k_{49}	2^{16}	$C6$ with $i=31$	2^{15}
33	k_{50}, k_{51}, k_{52}	2^{18}	$C7$ with $i=25$	2^{17}
34	k_{53}	2^{17}	$C7$ with $i=26$	2^{16}
35	k_{54}	2^{17}	$C7$ with $i=27$	2^{16}
36	k_{55}	2^{17}	$C7$ with $i=28$	2^{16}
37	k_{56}	2^{17}	$C7$ with $i=29$	2^{16}
38	k_{57}	2^{17}	$C7$ with $i=30$	2^{16}
39	k_{58}	2^{17}	$C7$ with $i=31$	2^{16}
40	$k_{59}, \cdots, k_{63}$	2^{21}	—	—

completely dominated by the distinguishing attack's success probability, and thus is also almost 1.

Clearly, we can obtain similar key recovery attacks for different values of i with $17 \le i \le 31$. The best result we have found is obtained when i equals to 25. When $i=25$, the attacker first guesses the values of 53 key bits $k_0, \cdots, k_{52}$, and then reduces the number of possible guesses from 2^{53} to about $2^{53} \cdot 2^{-6} = 2^{47}$ by checking whether the six conditions $C2, \cdots, C7$ hold simultaneously. Then the attacker guesses the values of the remaining 11 key bits $k_{53}, \cdots, k_{63}$ to recover the 64-bit key, which increases the number of possible guesses from 2^{47} to about $2^{47} \cdot 2^{11} = 2^{58}$. Since the maximum number of possible guesses is no more than 2^{58} in this whole key recovery process, the time complexity of this key recovery process is at most 2^{58}. Considering the cost of finding an input collision pair, the key recovery attack on GEA-1 has a total time complexity of $2^{12} + 2^{58} \approx 2^{58}$ GEA-1 encryptions, requiring one related key, 2^{11} chosen IVs and 2^{17} keystream bits. The success probability of this attack is almost 1.

5.2. Practical Key Recovery Attack on GEA-1 Using More Related Keys. In fact, the key recovery attacks on GEA-1 above are all "minimal" in the sense that each of them requires only one related key. If more related keys are available for cryptanalysis, the time complexity can be significantly reduced. Now, we attempt to propose a practical related key attack on GEA-1 using more related keys. It should be noted that there is a tradeoff between the number of related keys used in the attack and the time complexity of the attack. To achieve a practical time complexity, we assume that seven related keys are available to the attacker, i.e., for $i = 25, \cdots, 31$. Similar to the attacks using one related key above, for each of these seven related keys, the attacker should find an input collision pair such that the seven conditions $C1, \cdots, C7$ are simultaneously satisfied. This leads to a time complexity of $7 \times 2^{12} \approx 2^{14.81}$ GEA-1 encryptions and requires about $7 \times 2^{11} \approx 2^{13.81}$ chosen IVs. After then, the detailed process of recovering the key of GEA-1 is described in Table 6.

As shown in Table 6, in the first step, the attacker guesses the values of one key bit k_0, which leads to two possible guesses. Then, the attacker can check whether the condition $C2$ with $i = 25$ is satisfied, reducing the number of possible guesses from two to about $2 \cdot 2^{-1} = 1$, since $Pr(C2) = 0.5$ holds. The following 38 steps are all similar to the first step. After the first 39 steps in Table 6, about 2^{16} possible guesses are obtained. Then the attacker could make an exhaustive search of the remaining five key bits $k_{59}, \cdots, k_{63}$ to recover the 64-bit key of GEA-1, which increases the number of possible guesses from 2^{16} to $2^{16} \cdot 2^5 = 2^{21}$.

It is easy to see that the maximum number of possible guesses is no more than 2^{21}, as shown in Table 6. Thus, the time complexity of the key recovery process is at most 2^{21}. Considering the cost of finding input collision pairs, the total time complexity of the attack on GEA-1 is $2^{12} \times 7 + 2^{21} \approx 2^{21.02}$ GEA-1 encryptions, requiring seven related keys, $2^{13.81}$ chosen IVs and $7 \times 2^{17} \approx 2^{19.81}$ keystream bits. The success probability of this attack can be calculated as $p^7 \approx 1$, as seven related keys are used in this attack. To validate this cryptanalytic result, we have randomly chosen 100 keys and simulated the whole attack process once for each key on a common PC with 2.5 GHz Intel Pentium 4 processor. The experimental result shows that the attack above always succeeds recovering the 64-bit secret key of GEA-1, and the average time to recover the 64-bit key is approximately 41.75 s.

6. Conclusions

In this paper, we find that the initialization of GEA-1 is noninjective, due to that the input size of GEA-1 is much larger than the size of the NFSR used in the initialization of GEA-1. Based on this observation, a structural weakness of the GEA-1 stream cipher that has not been found by the previous works is discovered and analyzed. As results, new practical attacks on GEA-1 are proposed, and the cryptanalytic attacks show that GEA-1 cannot provide enough security and should be immediately prohibited from being supported in the massive GPRS devices. It should be noted that this newfound weakness of GEA-1 is not present in its successors GEA-2 and GEA-2a. That is because the input size of GEA-2 (or GEA-2a) is no smaller than the size of the NFSR. Hopefully, this can provide some new insights on how to design a secure GEA-like stream cipher.

Data Availability

No new data were generated or analysed in support of this research.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant numbers 61602514, 62202493, 61802437, and 61902428.

References

[1] C. Beierle, P. Derbez, G. Leander et al., "Cryptanalysis of the GPRS encryption algorithms GEA-1 and GEA-2." in *Advances in Cryptology —EUROCRYPT 2021*, A. Canteaut and F. X. Standaert, Eds., vol. 12697 of *Lecture Notes in Computer Science*, pp. 155–183, Springer, Cham, 2021.

[2] L. Ding, Z. Wu, X. Wang, Z. Guan, and M. Li, "New attacks on the GPRS encryption algorithms GEA-1 and GEA-2," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2878–2889, 2022.

[3] ETSI: security algorithms group of experts (sage), "Report on the specification, evaluation and usage of the gsm gprs encryption algorithm (gea)," Technical Report https://www.etsi.org/deliver/etsi_tr/101300_101399/101375/01.01.01, 1998.

[4] C. Brookson, "GPRS security," 2001, https://web.archive.org/web/20120914110208/.

[5] C. Beierle, T. Beyne, P. Felke, and G. Leander, "Constructing and deconstructing intentional weaknesses in symmetric ciphers," 2021, Cryptology ePrint Archive https://eprint.iacr.org/2021/829.

[6] C. Beierle, T. Beyne, P. Felke, and G. Leander, "Constructing and deconstructing intentional weaknesses in symmetric ciphers," in *Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022*, pp. 748–778, Association for Computing Machinery, Santa Barbara, CA, USA, August 2022.

[7] D. Amzaleg and I. Dinur, "Refined cryptanalysis of the GPRS ciphers GEA-1 and GEA-2," in *Advances in Cryptology–EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 57–85, Association for Computing Machinery, Trondheim, Norway, May 2022.

[8] L. Yuseop, J. Kitae, S. Jaechul, and H. Seokhie, "Related-key chosen IV attacks on Grain-v1 and Grain-128," in *Information Security and Privacy. ACISP 2008*, Y. Mu, W. Susilo, and J. Seberry, Eds., vol. 5107 of *Lecture Notes in Computer Science*, pp. 321–335, Springer, Berlin, 2008.

[9] L. Ding and J. Guan, "Related key chosen IV attack on Grain-128a stream cipher," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 5, pp. 803–809, 2013.

[10] S. Banik, S. Maitra, S. Sarkar, and T. M. Sönmez, "A chosen IV related key attack on Grain-128a," in *Information Security and Privacy. ACISP 2013*, C. Boyd and L. Simpson, Eds., vol. 7959 of *Lecture Notes in Computer Science*, pp. 13–26, Springer, Berlin, 2013.

[11] L. Ding, C. Jin, J. Guan, and Q. Wang, "Cryptanalysis of lightweight WG-8 stream cipher," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 645–652, 2014.

[12] A. Kircanski and A. M. Youssef, "On the sliding property of SNOW 3G and SNOW 2.0," *IET Information Security*, vol. 5, no. 4, pp. 199–206, 2011.

[13] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," in *Selected Areas in Cryptography. SAC 2001*, S. Vaudenay and A. M. Youssef, Eds., vol. 2259 of *Lecture Notes in Computer Science*, pp. 1–24, Springer, Berlin, Heidelberg, 2001.

[14] E. Biham, "New types of cryptanalytic attacks using related keys," *Journal of Cryptology*, vol. 7, no. 4, pp. 229–246, 1994.

[15] L. R. Knudsen, "Cryptanalysis of LOKI91," in *Advances in Cryptology—AUSCRYPT '92. AUSCRYPT 1992*, J. Seberry and Y. Zheng, Eds., vol. 718 of *Lecture Notes in Computer Science*, pp. 196–208, Springer, Berlin, Heidelberg, 1992.

[16] M. Ciet, G. Piret, and J.-J. Quisquater, "Related-key and slide attacks: analysis, connections, and improvements," in *Proc. ISIT 2002*, pp. 315–325, Lausanne, Switzerland, 2002.