Heliyon 10 (2024) e24197

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

A practical key recovery attack on the lightweight WG-5 stream cipher

Lin Ding *, Zhiyi Liao, Zhengting Li *, Zheng Wu, Xinhai Wang, Ziyu Guan

PLA SSF Information Engineering University, Zhengzhou 450001, China

ARTICLE INFO

Keywords:
Cryptanalysis
Slide attack
WG-5
Stream cipher
Lightweight cryptography

ABSTRACT

WG-5 is a lightweight stream cipher proposed for usage in the resource-constrained devices, e.g., passive RFID tags, industrial controllers, contactless smart cards and sensors. In this paper, a weakness called *slide property* of WG-5 which has not been discovered in previous works is for the first time explored and analyzed. The result shows that the probability that two related key-IV pairs of WG-5 generate the shifted keystreams can be up to 2^{-20}, which is significantly high compared with an ideal stream cipher that generates the random keystreams. The correctness and accuracy of this theoretical probability is confirmed experimentally. Based on the slide property of WG-5, some key recovery attacks on WG-5 in the related key setting are proposed. The cryptanalytic result shows that the 80-bit secret key of WG-5 can be recovered with a time complexity of $2^{25.615}$, requiring 6 related keys and 80 keystream bits for each of $2^{24.585}$ chosen IVs. The experimental result validates our attack and shows that WG-5 can be broken within about 92.054 seconds on a common PC in the related key setting. These results imply that the design of WG-5 is far from optimal and needs to be strengthened to provide enough security for the lightweight constrained applications.

1. Introduction

In nowadays, the deployment of the resource-constrained devices, e.g., contactless smart cards, sensors, RFID tags, industrial controllers and health-care devices, is becoming more and more popular. However, the majority of the conventional cryptographic algorithms were designed for desktop and server environments with sufficient resources, which makes them difficult or impossible to be implemented in the resource-constrained devices. Lightweight cryptography, a subfield of cryptography, is a form of encryption algorithms designed for the resource-constrained devices. Unlike the conventional cryptographic algorithms, lightweight encryption algorithms aim at using less memory, fewer computing resources, and a smaller amount of power to provide secure solutions for the resource-constrained devices. Now, lightweight cryptographic techniques are widely recognized to be the most appropriate solution to provide security and privacy for resource-constrained IoT networks. For the most current state-of-the-art researches in lightweight cryptography, we refer the reader to some literature reviews [1,2] in this field.

WG-5 is a lightweight stream cipher proposed in 2013 by Aagaard, Gong and Mota [3], and is expected to provide a target security level of 80 bits. It is a lightweight version of the well-known Welch-Gong (WG) family of stream ciphers. The original WG is a synchronous stream cipher developed by Nawaz and Gong [4], and was submitted to the eSTREAM project [5] in 2005. In

* Corresponding author.
E-mail addresses: dinglin_cipher@163.com (L. Ding), lizhengting0225@163.com (Z. Li).

https://doi.org/10.1016/j.heliyon.2024.e24197
Received 23 July 2023; Received in revised form 11 December 2023; Accepted 4 January 2024

Available online 11 January 2024

Table 1
The comparisons of our cryptanalytic results with the previous attacks on WG-5.

Time complexity	Keystream required for one key-IV pair	Total data complexity	Number of required related keys	Ref.
2^{33}	2^{15}	2^{15}	-	[20]
$2^{76.81}$	80	$2^{12.644}$	-	[21]
2^{70}	80	$2^{26.322}$	1	**Sect. 4.1**
$2^{25.615}$	80	$2^{30.907}$	6	**Sect. 4.2**

the following years, the designers had tried to enrich their design and proposed several new variants of WG, e.g., WG-5 [3], WG-7 [6], WG-8 [7], WG-16 [8] and WG-29 [9]. The WG family of stream ciphers uses the same structure, which is mainly made up of a linear feedback shift registers (LFSR) and a Welch-Gong filtering transformation. As shown in [10], the keystream generated by the filtering transformation is theoretically proven to provide random properties. Besides, based on this filtering transformation, a new lightweight sponge-based authenticated cipher called WAGE was proposed in [11]. It was submitted to the NIST lightweight cryptography standardization competition [12] and became one of 32 Round 2 candidates of this competition. Due to the novelty in design, the WG family of stream ciphers has attracted a lot of attention in recent years, and several attacks on them have been proposed in [13–19].

Related works. To the best of our knowledge, there have been two published attacks on WG-5 up to now. In [20], Rønjom presented an algebraic attack on WG-5, which recovers the 80-bit secret key of WG-5 with a time complexity of 2^{33} and requires about 2^{15} keystream bits. However, in lightweight constrained applications, the available online data for a given key that may be queried by an adversary is usually limited by the running protocol, thus large amount of data is generally hard to collect. In [21], Rohit, AlTawy, and Gong proposed a MILP-based cube attack on the reduced-round WG-5. Their result shows that the secret key of WG-5 after 24 (out of 64) rounds of initialization can be recovered with a time complexity of $2^{76.81}$. Since there are five cubes are used in this attack and each cube has a size of 4, thus this attack requires a total of $5 \times 2^4 \times 80 = 2^{12.644}$ keystream bits. Note that only 80 keystream bits are required for each key-IV pair in this cube attack. They claimed that their cube attack is a more realistic attack, since it requires significantly less keystream bits for one key-IV pair than the algebraic attack presented in [20]. However, the time complexity is extremely high, which makes their cube attack unpractical. Since the full initialization process of WG-5 consists of 64 rounds, WG-5 has sufficient resistance against cube attacks, according to the cryptanalytic result in [21].

Our contributions. A weakness called *slide property* of the lightweight WG-5 stream cipher that has not been discovered in previous works is explored and analyzed in this paper. As results, some key recovery attacks on WG-5 in the related key setting are proposed. The comparisons of our cryptanalytic results with the previous attacks are given in Table 1. The contributions of this paper can be summarized as follows.

- This paper for the first time discovers the slide property of the lightweight WG-5 stream cipher. The result shows that the probability that two related key-IV pairs of WG-5 generate the shifted keystreams can be up to 2^{-20}, which is significantly high compared with an ideal stream cipher that generates the random keystreams. The correctness and accuracy of this theoretical probability is confirmed experimentally. The weakness indicates that the keystreams generated by WG-5 are far from random.
- Based on the slide property of WG-5, a simple key recovery attack on WG-5 using one related key is proposed. The attack recovers the 80-bit key of WG-5 with a time complexity of 2^{70} and a success probability of 0.632. It requires a total of $2^{20} \times 80 = 2^{26.322}$ keystream bits, as 80 keystream bits are needed for each of 2^{20} chosen IVs.
- To reduce the time complexity of the simple attack on WG-5 above, a more practical key recovery attack on WG-5 using more related keys is proposed. In this attack, the 80-bit key of WG-5 can be recovered with a time complexity of $2^{25.615}$, requiring 6 related keys and 80 keystream bits for each of $2^{24.585}$ chosen IVs. Thus, the attack requires a total of $2^{24.585} \times 80 = 2^{30.907}$ keystream bits, and has a success probability of 0.897. We have validated this cryptanalytic result by simulating the whole process of the attack. The experimental result shows that WG-5 can be broken within about 92.054 seconds on a common PC in the related key setting. These results imply that the design of WG-5 is far from optimal and needs to be strengthened to provide enough security for the lightweight constrained applications.
- Compared with the existing attacks on WG-5, there are some obvious advantages in our attack. On the one hand, our attack requires only 80 keystream bits for one key-IV pair, which is much less than the algebraic attack on WG-5 in [20]. On the other hand, our attack can recover the 80-bit key of full WG-5 with a time practical complexity of $2^{25.615}$, while the cube attack on WG-5 proposed in [21] can only threaten the security of reduced-round WG-5 (i.e., 24 out of 64 rounds of initialization) and has a time complexity of $2^{76.81}$ which is obviously unrealistic to be performed on a common PC.

Outline. The rest of this paper is structured as follows. A brief description of WG-5 is given in Section 2. In Section 3, the slide property of WG-5 is discovered and analyzed. Based on the slide property of WG-5, some key recovery attacks on WG-5 are proposed in Section 4. The paper is concluded in Section 5.

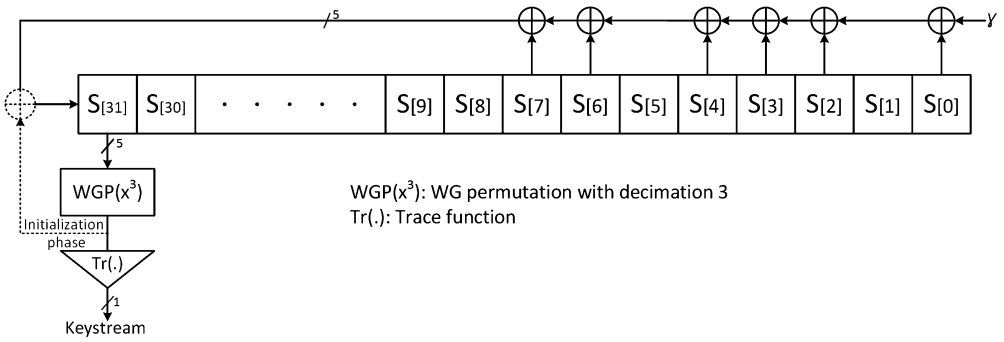

Fig. 1. An overview of the WG-5 stream cipher.

2. A brief description of WG-5

WG-5 is a lightweight variant of the well-known WG stream cipher. It supports an 80-bit secret key and an 80-bit initialization vector (IV). As depicted in Fig. 1 [3], the WG-5 stream cipher consists of a 32-stage linear feedback shift register (LFSR) and Welch-Gong filtering transformation. The LFSR is defined over the extension field $\mathbb{F}_{2^5}$ with the primitive feedback polynomial $x^{32}+x^7+x^6+x^4+x^3+x^2+\gamma$, where $\gamma=\alpha^4+\alpha^3+\alpha^2+\alpha+1$ and α is a root of the polynomial $x^5+x^4+x^2+x+1$. It operates in two processes, i.e., an initialization process and a keystream generation process.

2.1. Initialization process of WG-5

The WG-5 stream cipher takes an 80-bit key and an 80-bit IV as input. Denote by $K=(K[0],\cdots,K[15])=\left(k_0,\cdots,k_{79}\right)$ and $IV=(IV[0],\cdots,IV[15])=\left(iv_0,\cdots,iv_{79}\right)$ the 80-bit key and 80-bit IV of WG-5, respectively, where $K[i]=\left(k_{5i},k_{5i+1},k_{5i+2},k_{5i+3},k_{5i+4}\right)$ and $IV[i]=\left(iv_{5i},iv_{5i+1},iv_{5i+2},iv_{5i+3},iv_{5i+4}\right)$ for $0\le i\le 15$. Let $S^{(t)}=\left(S_0^{(t)},\cdots,S_{31}^{(t)}\right)$ denote the 160-bit state of WG-5 at time t, where $S_i^{(t)}=\left(s_{5i}^{(t)},s_{5i+1}^{(t)},s_{5i+2}^{(t)},s_{5i+3}^{(t)},s_{5i+4}^{(t)}\right)$ for $0\le i\le 31$. At the beginning of initialization of WG-5, the key and IV are loaded into the LFSR as follows.

For $0\le j\le 15$,

$$S_{2j}^{(0)}=K[j]$$
$$S_{2j+1}^{(0)}=IV[j]$$

After loading the Key and IV, the initialization process runs for 64 rounds with the output of WG-permutation feedback into the LFSR. The internal state update function of WG-5 is given as follows.

For $0\le t\le 63$,

$$S_i^{(t+1)}=S_{i+1}^{(t)},0\le i\le 30$$
$$S_{31}^{(t+1)}=\gamma S_0^{(t)}\oplus S_2^{(t)}\oplus S_3^{(t)}\oplus S_4^{(t)}\oplus S_6^{(t)}\oplus S_7^{(t)}\oplus WGP\left(\left(S_{31}^{(t)}\right)^3\right)$$

where $WGP\left(x^3\right)$ denotes the nonlinear WG-permutation with decimation $d=3$ and can be simply considered as a 5-bit S-box.

2.2. Keystream generation process of WG-5

After the 64 initialization rounds, the WG-5 stream cipher is ready to generate keystream bits. During the keystream generation process, one keystream bit per clock is generated.

For $t\ge 64$,

$$z_{t-64}=Tr\left(WGP\left(\left(S_{31}^{(t)}\right)^3\right)\right)$$

where $Tr(\cdot)$ denotes the trace function. To remove any possible ambiguity, the Algebraic Normal Form (ANF) of this keystream generation function is given as follows.

$$\begin{aligned}z_{t-64}=&s_{155}^{(t)}s_{156}^{(t)}s_{157}^{(t)}\oplus s_{155}^{(t)}s_{157}^{(t)}s_{158}^{(t)}\oplus s_{155}^{(t)}s_{157}^{(t)}s_{159}^{(t)}\oplus s_{155}^{(t)}s_{158}^{(t)}s_{159}^{(t)}\\&\oplus s_{156}^{(t)}s_{157}^{(t)}s_{158}^{(t)}\oplus s_{156}^{(t)}s_{158}^{(t)}s_{159}^{(t)}\oplus s_{155}^{(t)}s_{156}^{(t)}\oplus s_{155}^{(t)}s_{157}^{(t)}\oplus s_{155}^{(t)}s_{159}^{(t)}\\&\oplus s_{156}^{(t)}s_{158}^{(t)}\oplus s_{156}^{(t)}s_{159}^{(t)}\oplus s_{155}^{(t)}\oplus s_{156}^{(t)}\oplus s_{157}^{(t)}\oplus s_{158}^{(t)}\oplus s_{159}^{(t)}\end{aligned}$$

Table 2
The concrete relation of $S_i^{(2)}$ and $S'^{(0)}_i$ for $0 \le i \le 31$ to form a 2-slide pair.

Key-IV pair	Internal state	$i=0$	$i=1$	$i=2$	$i=3$	$\cdots\cdots$	$i=29$	$i=30$	$i=31$
(K, IV)	$S^{(0)}$	$K[0]$	$IV[0]$	$K[1]$	$IV[1]$	$\cdots\cdots$	$IV[14]$	$K[15]$	$IV[15]$
	$S^{(1)}$	$IV[0]$	$K[1]$	$IV[1]$	$K[2]$	$\cdots\cdots$	$K[15]$	$IV[15]$	$S_{31}^{(1)}$
	$S^{(2)}$	$K[1]$	$IV[1]$	$K[2]$	$IV[2]$	$\cdots\cdots$	$IV[15]$	$S_{31}^{(1)}$	$S_{31}^{(2)}$
(K', IV')	$S'^{(0)}$	$K'[0]$	$IV'[0]$	$K'[1]$	$IV'[1]$	$\cdots\cdots$	$IV'[14]$	$K'[15]$	$IV'[15]$

In the keystream generation process, the internal state update function is given as follows. It should be noted that unlike the initialization process, the non-linear feedback is not used to update the leftmost register of the LFSR during the keystream generation process.

For $t \ge 64$,

$$S_i^{(t+1)} = S_{i+1}^{(t)}, 0 \le i \le 30$$
$$S_{31}^{(t+1)} = \gamma S_0^{(t)} \oplus S_2^{(t)} \oplus S_3^{(t)} \oplus S_4^{(t)} \oplus S_6^{(t)} \oplus S_7^{(t)}$$

3. Slide property of WG-5

In this section, the slide property of the WG-5 stream cipher will be explored and analyzed. The slide property focuses on describing the structural self-similarity of a symmetric cipher, and has been successfully applied in cryptanalysis of many well-known stream ciphers, e.g., Trivium [22], Salsa20 [22], Grain family of stream ciphers [23–25], Decim v2 [26], SNOW 2.0 [27], SNOW 3G [27], GEA-1 [28] and GEA-2 [28].

Let (K, IV) and (K', IV') be two different key-IV pairs of WG-5, and denote by $S^{(t)} = \left(S_0^{(t)}, \cdots, S_{31}^{(t)}\right)$ and $S'^{(t)} = \left(S'^{(t)}_0, \cdots, S'^{(t)}_{31}\right)$ the internal states of WG-5 at time t generated by (K, IV) and (K', IV'), respectively. For convenience, a definition is given as follows.

Definition 1. Two different key-IV pairs (K, IV) and (K', IV') are called a **2-slide pair** for WG-5, if $S^{(2)} = S'^{(0)}$ holds.

Clearly, there are 32 equations that have to be satisfied to form a 2-slide pair, i.e., $S_i^{(2)} = S'^{(0)}_i, 0 \le i \le 31$. The concrete relation of $S_i^{(2)}$ and $S'^{(0)}_i$ for $0 \le i \le 31$ to form a 2-slide pair is shown in Table 2.

As depicted in Table 2, to form a 2-slide pair, the following 32 equations should be satisfied.

- $K[i+1] = K'[i] (0 \le i \le 14)$
- $IV[i+1] = IV'[i] (0 \le i \le 14)$
- $S_{31}^{(1)} = K'[15]$
- $S_{31}^{(2)} = IV'[15]$

where

$$\begin{aligned} S_{31}^{(1)} &= \gamma S_0^{(0)} \oplus S_2^{(0)} \oplus S_3^{(0)} \oplus S_4^{(0)} \oplus S_6^{(0)} \oplus S_7^{(0)} \oplus WGP\left(\left(S_{31}^{(0)}\right)^3\right) \\ &= \gamma K[0] \oplus K[1] \oplus IV[1] \oplus K[2] \oplus K[3] \oplus IV[3] \oplus WGP\left((IV[15])^3\right) \\ S_{31}^{(2)} &= \gamma S_0^{(1)} \oplus S_2^{(1)} \oplus S_3^{(1)} \oplus S_4^{(1)} \oplus S_6^{(1)} \oplus S_7^{(1)} \oplus WGP\left(\left(S_{31}^{(1)}\right)^3\right) \\ &= \gamma S_1^{(0)} \oplus S_3^{(0)} \oplus S_4^{(0)} \oplus S_5^{(0)} \oplus S_7^{(0)} \oplus S_8^{(0)} \oplus WGP\left(\left(S_{31}^{(1)}\right)^3\right) \\ &= \gamma IV[0] \oplus IV[1] \oplus K[2] \oplus IV[2] \oplus IV[3] \oplus K[4] \oplus WGP\left(\left(S_{31}^{(1)}\right)^3\right) \end{aligned}$$

It is easy to see that the first 15 equations, i.e., $K[i+1] = K'[i] (0 \le i \le 14)$, can be directly satisfied in the related key setting. Similarly, the next 15 equations, i.e., $IV[i+1] = IV'[i] (0 \le i \le 14)$, can be directly satisfied in the chosen IV setting. Now, we can define a relationship called **R1** between the key-IV pair (K, IV) and its related key-IV pair (K', IV') as follows.

$$K' = K <<< 5$$
$$IV' = IV <<< 5$$

where $<<< 5$ denotes the rotation operation by 5 bits to the left.

It is easy to see that the first 30 equations above are directly satisfied under the relationship **R1**. Now, we consider the last two equations, i.e., $S_{31}^{(1)} = K'[15]$ and $S_{31}^{(2)} = IV'[15]$. As depicted above, it is clear to see that there are 30 IV bits totally (i.e., $IV[0], IV[1], IV[2], IV[3], IV[15], IV'[15]$) involved in the last two equations. More specifically, the 5-bit vector $IV[2]$ only appears in the equation $S_{31}^{(2)} = IV'[15]$, and the 5-bit vector $IV[3]$ appears in both of these two equations. Thus, the probability that the last two equations hold simultaneously is $2^{-5} \times 2^{-5} = 2^{-10}$, and if all of 2^{10} possible values of $(IV[2], IV[3])$ are exhausted, there must be one value of $(IV[2], IV[3])$ such that the last two equations hold simultaneously. Thus, an observation can be obtained as follows.

Observation 1. Under the relationship **R1**, two different key-IV pairs (K, IV) and (K', IV') of WG-5 form a 2-slide pair with a probability of 2^{-10}.

When $S^{(2)} = S'^{(0)}$ holds, then $S^{(t+2)} = S'^{(t)}$ holds directly for $1 \le t \le 62$. However, it is not necessarily that $S^{(65)} = S'^{(63)}$ holds, since $S^{(65)}$ is obtained from $S^{(64)}$ in the keystream generation process, while $S'^{(63)}$ is obtained from $S'^{(62)}$ in the initialization process. The only difference between the keystream generation process and the initialization process of WG-5 is the usage of the nonlinear WG-permutation's output. Thus, $S^{(65)} = S'^{(63)}$ holds if and only if $WGP\left(\left(S_{31}^{(64)}\right)^3\right) = 0$ holds. Similarly, it has that $S^{(66)} = S'^{(64)}$ holds if and only if $WGP\left(\left(S_{31}^{(65)}\right)^3\right) = 0$ holds. When $S^{(66)} = S'^{(64)}$ holds, then $S^{(t+2)} = S'^{(t)}$ always holds for $t \ge 65$, which implies that $z_{t+2} = z'_t$ always holds for $t \ge 0$. In other words, the related pair (K', IV') generates a 2-bit shifted keystream with respect to (K, IV). Thus, based on the above analysis, a new observation can be obtained as follows.

Observation 2. Under the relationship **R1**, the related pair (K', IV') generates a 2-bit shifted keystream with respect to (K, IV), when the following conditions are satisfied simultaneously.

- $S_{31}^{(1)} = K'[15] = K[0]$
- $S_{31}^{(2)} = IV'[15] = IV[0]$
- $WGP\left(\left(S_{31}^{(64)}\right)^3\right) = 0$
- $WGP\left(\left(S_{31}^{(65)}\right)^3\right) = 0$

According to **Observation 2**, it is easy to see that the related pair (K', IV') generates a 2-bit shifted keystream with respect to (K, IV) with a probability of $2^{-5} \times 2^{-5} \times 2^{-5} \times 2^{-5} = 2^{-20}$. This is a high probability compared with an ideal stream cipher that generates random keystreams. To validate this theoretical probability, we have made an experiment. In this experiment, we randomly choose 100 keys and 2^{23} different IVs for each key. The experimental result shows that, there are 7.93 2-slide pairs on average found per key, which leads to an experimental probability of $7.93/2^{23} \approx 2^{-20.013}$. Clearly, the experimental probability is quite close to the theoretical probability 2^{-20}, which confirms the correctness of the calculated theoretical probability.

It should be noted that there are other slide pairs for longer shifts, e.g., 4-slide pair and 6-slide pair, but the probability that two different key-IV pairs generate the shifted keystreams will become much smaller than 2^{-20}. Therefore, we no longer consider the slide properties of WG-5 with longer shifts.

4. Key recovery attacks on WG-5 based on the slide property of WG-5

Based on the slide property of WG-5 discovered in the above section, this section aims at presenting some key recovery attacks on the lightweight WG-5 stream cipher in the related key setting. As pointed by Biham and Dunkelman in [29], related key setting is a standard attack scenario in cryptanalysis of symmetric cryptosystems and has been applied in breaking some well-known stream ciphers. For instance, related-key weaknesses of the stream cipher RC4 led to a practical attack on the WEP protocol [30]. Generally speaking, in the related key setting [31], the attacker is assumed to obtain enough keystream bits generated by two different keys that have a known relationship, while their values are unknown to the attacker. Thus, the relationship **R1** defined above is reasonable and acceptable in the related key chosen IV setting.

4.1. A simple key recovery attack on WG-5 using one related key

After discovering the slide property of WG-5, we are ready to present our key recovery attacks on the cipher. First, a simple key recovery attack on WG-5 using one related key is proposed in this subsection.

To recover the 80-bit key of WG-5, the attacker has to find at least one 2-slide pair, which can be done by the following algorithm called **Algorithm 1**. In this algorithm, a total of $N\left(\ge 2^{20}\right)$ chosen IVs are required, since the related pair (K', IV') generates a 2-bit shifted keystream with respect to (K, IV) with a probability of 2^{-20}. In these N chosen IVs, the value of $(IV[2], IV[3])$ should be exhausted and the remaining 70 bits of IV can be randomly chosen.

Algorithm 1 Finding 2-slide pairs of WG-5.

Input: the fixed and unknown 80-bit key K, and N chosen IVs $IV_1, \cdots, IV_N$
Output: 2-slide pairs of WG-5

1. For h from 1 to N, do the following:

- Generate a keystream Z by the key-IV pair (K, IV_h);
- Generate a keystream Z' by the related key-IV pair (K', IV'_h);
- Check if Z and Z' are 2-bit shifted keystreams, if yes, a 2-slide pair is found and go to Step 2; otherwise, return to Step 1 and try the next chosen IV.

2. Output the found 2-slide pairs of WG-5.

As shown in **Algorithm 1**, for each of N chosen IVs, **Algorithm 1** should generate two keystreams. Thus, the algorithm has a time complexity of $2N$. Since the related pair (K', IV') generates a 2-bit shifted keystream with respect to (K, IV) with a probability of 2^{-20}, it is easy to know that **Algorithm 1** succeeds finding at least one 2-slide pair with a probability of $p = 1 - \left(1 - 2^{-20}\right)^N$. Clearly, $N = 2^{20}$ is a reasonable choice, and then the success probability $p \approx 0.632$ holds. As shown in **Algorithm 1**, since we only utilize the generated keystreams to make a check, the required length of the keystream generated by one key-IV pair is quite small. Since WG-5 has a key size of 80 bits, thus the required length of the keystream generated by one key-IV pair is chosen to be 80, which is large enough to uniquely determine the secret key.

Once the 2-slide pair of WG-5 is found, the attacker can recover some key bits of WG-5. More specifically, for the found 2-slide pair, the first two conditions in **Observation 2**, i.e., $S_{31}^{(1)} = K[0]$ and $S_{31}^{(2)} = IV[0]$, are satisfied simultaneously. In these two equations, there are 25 key bits involved, i.e., $K[0], K[1], K[2], K[3], K[4]$. When the attacker makes an exhaustive search of $(K[0], K[1], K[2])$, the remaining 10 key bits $(K[3], K[4])$ can be determined directly as follows.

$$K[3] = K[0] \oplus \gamma K[0] \oplus K[1] \oplus IV[1] \oplus K[2] \oplus IV[3] \oplus WGP\left((IV[15])^3\right)$$

$$K[4] = IV[0] \oplus \gamma IV[0] \oplus IV[1] \oplus K[2] \oplus IV[2] \oplus IV[3] \oplus WGP\left((K[0])^3\right)$$

After recovering the 10 key bits $(K[3], K[4])$, the attacker can make an exhaustive search of the remaining 55 key bits, i.e., $K[5], \cdots, K[15]$, to recover the 80-bit key of WG-5. Thus, the key recovery process has a time complexity of $2^{15} \times 2^{55} = 2^{70}$, since the attacker has to make an exhaustive search of 70 key bits totally. Considering the time cost of finding a 2-slide pair, the key recovery attack on WG-5 has a total time complexity of $2^{21} + 2^{70} \approx 2^{70}$, requiring one related key and 80 keystream bits for each of 2^{20} chosen IVs. The success probability of this attack is about 0.632, which mainly depends on the success probability of **Algorithm 1**. It should be noted that the success probability of the attack can be easily improved when more than 2^{20} chosen IVs are used in **Algorithm 1**. Assume that 2^{22} chosen IVs are used in **Algorithm 1**, the success probability of the attack will become $p = 1 - \left(1 - 2^{-20}\right)^{2^{22}} \approx 0.982$. Now, the attack has a time complexity of $2^{22} \times 2 + 2^{70} \approx 2^{70}$, requiring one related key and 80 keystream bits for each of 2^{22} chosen IVs.

4.2. A practical key recovery attack on WG-5 using more related keys

In the simple key recovery attack on WG-5 above, the attacker can recover the 80-bit key of WG-5 with a time complexity of 2^{70}, which is too high that makes the attack unpractical on a common PC. In fact, if more related keys are available to the attacker, the time complexity can be reduced significantly. In this subsection, a practical key recovery attack on WG-5 using more related keys is proposed.

There are 7 keys $K_0, \cdots, K_6$ used in this practical key recovery attack, and the relation between these keys is defined as follows.

$$K_{i+1} = K_i <<< 10$$

where $K_0 = K$. The process of recovering the 80-bit key K of WG-5 consists of 7 steps, which is described as follows in details.

- In the first step, the attacker utilizes the keys K_0 and K_1 and implements **Algorithm 1** once to find a 2-slide pair. After that, the attacker guesses the value of 15 key bits $K[0], K[1], K[2]$, and determines the value of 10 key bits $K[3], K[4]$ using the guessed key bits.
- In the second step, the attacker utilizes the keys K_1 and K_2 and implements **Algorithm 1** once to find a 2-slide pair. Since the 15 key bits $K[2], K[3], K[4]$ have been determined in the first step, the attacker does not need to guess more key bits and can directly determine the 10 key bits $K[5], K[6]$.
- In the third step, the attacker utilizes the keys K_2 and K_3 and implements **Algorithm 1** once to find a 2-slide pair. Since the 15 key bits $K[4], K[5], K[6]$ have been determined in the front steps, the attacker does not need to guess more key bits and can directly determine the 10 key bits $K[7], K[8]$.
- In the fourth step, the attacker utilizes the keys K_3 and K_4 and implements **Algorithm 1** once to find a 2-slide pair. Since the 15 key bits $K[6], K[7], K[8]$ have been determined in the front steps, the attacker does not need to guess more key bits and can directly determine the 10 key bits $K[9], K[10]$.

- In the fifth step, the attacker utilizes the keys K_4 and K_5 and implements **Algorithm 1** once to find a 2-slide pair. Since the 15 key bits $K[8], K[9], K[10]$ have been determined in the front steps, the attacker does not need to guess more key bits and can directly determine the 10 key bits $K[11], K[12]$.
- In the sixth step, the attacker utilizes the keys K_5 and K_6 and implements **Algorithm 1** once to find a 2-slide pair. Since the 15 key bits $K[10], K[11], K[12]$ have been determined in the front steps, the attacker does not need to guess more key bits and can directly determine the 10 key bits $K[13], K[14]$.
- In the last step, the remaining 5 key bits $K[15]$ can be recovered by making an exhaustive search.

In the seven steps above, the attacker has to implement **Algorithm 1** six times, which leads to a time complexity of $2^{23} \times 6 \approx 2^{25.585}$. It should be noted that to achieve a high success probability, 2^{22} chosen IVs are used in **Algorithm 1** in this attack, which implies that the time complexity of **Algorithm 1** is 2^{23} here. In the whole process, only 20 key bits have to be guessed, i.e., $K[0], K[1], K[2], K[15]$, which leads to a time complexity of 2^{20}. Thus, the attack has a total time complexity of $2^{25.585} + 2^{20} \approx 2^{25.615}$, requiring 6 related keys and $2^{22} \times 6 \approx 2^{24.585}$ chosen IVs. Note that only 80 keystream bits are required for each key-IV pair in this attack. Since **Algorithm 1** is implemented six times, the success probability of the attack can be calculated as $p^6 = 0.982^6 = 0.897$. To validate this cryptanalytic result, we have simulated the whole process of the attack above. In this simulation, we randomly choose 10 keys and execute the whole attack process for each key. The simulation results show that it takes about 90.175 seconds on average to implement **Algorithm 1** six times, and the key recovery process can be done within 1.879 seconds on average. The simulation was implemented on a common PC with 2.5 GHz Intel Pentium 4 processor. The experimental results corroborate our assertion that WG-5 can be broken within about 92.054 seconds on a common PC in the related key setting.

5. Conclusions

In this paper, a weakness called *slide property* of WG-5 is for the first time discovered. That is, the probability that two related key-IV pairs of WG-5 generate the shifted keystreams can be up to 2^{-20}, which is a significantly high probability compared to an ideal stream cipher that generates random keystreams. The correctness of this probability is confirmed experimentally. Based on this weakness, a practical key recovery attack on WG-5 is proposed. In this attack, the 80-bit key of WG-5 can be recovered with a time complexity of $2^{25.615}$, requiring 6 related keys and 80 keystream bits for each of $2^{24.585}$ chosen IVs. The attack has a success probability of 0.897. We have validated this cryptanalytic result by simulating the whole process of the attack. The experimental results show that WG-5 can be broken within about 92.054 seconds on a PC in the related key setting. These results imply that the design of WG-5 is far from optimal and needs to be strengthened to provide enough security for the lightweight constrained applications. It should be noted that the attack on WG-5 proposed in this paper is in the related key setting, which is more stringent than the single key setting. It is an interesting task to propose an efficient key recovery attack on full WG-5 in the single key setting. We leave it as a future work.

Funding statement

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

CRediT authorship contribution statement

Lin Ding: Writing – original draft, Validation, Supervision, Resources, Methodology, Investigation, Funding acquisition, Conceptualization. **Zhiyi Liao:** Visualization, Resources, Formal analysis, Data curation. **Zhengting Li:** Writing – original draft, Visualization, Validation, Methodology, Conceptualization. **Zheng Wu:** Visualization, Software, Resources, Formal analysis, Data curation. **Xinhai Wang:** Writing – review & editing, Resources, Investigation. **Ziyu Guan:** Writing – review & editing, Visualization, Validation, Methodology, Investigation, Data curation.

Declaration of competing interest

The authors declare no competing interests.

Data availability

No data was used for the research described in the article.

References

[1] R. Muhammad, M. Quazi, I. Rafiqul, Lightweight cryptography in IoT networks: a survey, Future Gener. Comput. Syst. 129 (2022) 77–89.
[2] A. Khattab, Z. Jeddi, E. Amini, M. Bayoumi, Cryptography in RFID systems, in: RFID Security, Springer, Cham, 2017, pp. 43–72.
[3] M.D. Aagaard, G. Gong, R.K. Mota, Hardware implementations of the WG-5 cipher for passive rfid tags, in: 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2013, pp. 29–34.
[4] G. Gong, Y. Nawaz, The WG stream cipher, Submitted to eSTREAM, https://www.ecrypt.eu.org/stream/ciphers/wg/wg.pdf, April 29, 2005.

[5] eSTREAM: the ECRYPT stream cipher project, https://www.ecrypt.eu.org/stream/index.html, 2008.
[6] Y. Luo, Q. Chai, G. Gong, X. Lai, A lightweight stream cipher WG-7 for RFID encryption and authentication, in: IEEE Global Telecommunications Conference (GLOBECOM), Dec 2010, 2010, pp. 1–6.
[7] X. Fan, K. Mandal, G. Gong, WG-8: a lightweight stream cipher for resource-constrained smart devices, in: K. Singh, A. Awasthi (Eds.), Quality, Reliability, Security and Robustness in Heterogeneous Networks, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 115, Springer, Berlin, 2013, pp. 617–632.
[8] H. El-Razouk, A. Reyhani-Masoleh, G. Gong, New hardware implementationsof WG(29,11) and WG-16 streamciphers using polynomial basis, IEEE Trans. Comput. 64 (7) (1 July 2015) 2020–2035, https://doi.org/10.1109/TC.2014.2346207.
[9] Y. Nawaz, G. Gong, WG: a family of stream ciphers with designed randomness properties, Inf. Sci. 178 (7) (2008) 1903–1916.
[10] G. Gong, A.M. Youssef, Cryptographic properties of the Welch-Gong transformation sequence generators, IEEE Trans. Inf. Theory 48 (11) (2002) 2837–2846.
[11] R. AlTawy, G. Gong, K. Mandal, R. Rohit, WAGE: an authenticated encryption with a twist, IACR Trans. Symmetric Cryptol. 2020 (S1) (2020) 132–159.
[12] Lightweight Cryptography, IST lightweight crypto standardization, https://csrc.nist.gov/Projects/Lightweight-Cryptography, March 01, 2023.
[13] M.A. Orumiehchiha, J. Pieprzyk, R. Steinfeld, Cryptanalysis of WG-7: a lightweight stream cipher, Cryptogr. Commun. 4 (3–4) (2012) 277–285.
[14] L. Ding, C. Jin, J. Guan, Q. Wang, Cryptanalysis of lightweight WG-8 stream cipher, IEEE Trans. Inf. Forensics Secur. 9 (2014) 645–652.
[15] S. Rostami, E. Shakour, M.A. Orumiehchiha, J. Pieprzyk, Cryptanalysis of WG-8 and WG-16 stream ciphers, Cryptogr. Commun. 11 (2019) 351–362.
[16] M. Joseph, G. Sekar, R. Balasubramanian, Distinguishing attacks on (ultra-) lightweight WG ciphers, in: Lightweight Cryptography for Security and Privacy, 5th International Workshop, LightSec 2016, Aksaray, Turkey, September 21–22, 2016, Springer International Publishing, Cham, pp. 45-59, Revised Selected Papers.
[17] L. Ding, C. Jin, J. Guan, S. Zhang, T. Cui, D. Han, W. Zhao, Cryptanalysis of WG family of stream ciphers, Comput. J. 58 (10) (2015) 2677–2685.
[18] L. Ding, D. Gu, L. Wang, C. Jin, J. Guan, A real-time related key attack on the WG-16 stream cipher for securing 4G-LTE networks, J. Inf. Sec. Appl. 63 (2021) 103015.
[19] M.A. Orumiehchiha, S. Rostami, E. Shakour, J. Pieprzyk, A differential fault attack on the WG family of stream ciphers, J. Cryptogr. Eng. 10 (2) (2020) 189–195.
[20] S. Rønjom, Improving algebraic attacks on stream ciphers based on linear feedback shift register over $\mathbb{F}_{2^k}$, Des. Codes Cryptogr. 82 (2017) 27–41.
[21] R. Rohit, R. AlTawy, G. Gong, Milp-based cube attack on the reduced-round WG-5 lightweight stream cipher, in: Cryptography and Coding: 16th IMA International Conference, IMACC 2017, Oxford, UK, December 12–14, 2017, Proceedings, Springer International Publishing, Cham, pp. 333-351.
[22] D. Priemuth-Schmid, A. Biryukov, Slid pairs in Salsa20 and trivium, in: Proceedings of INDOCRYPT 2008, India, 2008, pp. 1–14.
[23] L. Yuseop, J. Kitae, S. Jaechul, H. Seokhie, Related-key chosen IV attacks on Grain-v1 and Grain-128, in: Proc. ACISP 2008, Wollongong, Australia, Jul. 2008, pp. 321–335.
[24] L. Ding, J. Guan, Related key chosen IV attack on Grain-128a stream cipher, IEEE Trans. Inf. Forensics Secur. 8 (5) (May. 2013) 803–809.
[25] S. Banik, S. Maitra, S. Sarkar, S.T. Meltem, A chosen IV related key attack on Grain-128a, in: Proc. ACISP 2013, Brisbane, QLD, Australia, Jul. 2013, pp. 13–26.
[26] L. Ding, J. Guan, Related-key chosen IV attack on Decim v2 and Decim-128, Math. Comput. Model. 55 (1–2) (Jan. 2012) 123–133.
[27] A. Kircanski, A. Youssef, On the sliding property of SNOW 3G and SNOW 2.0, IET Inf. Secur. 5 (4) (Dec. 2011) 199–206.
[28] L. Ding, Z. Wu, X. Wang, Z. Guan, M. Li, New attacks on the GPRS encryption algorithms GEA-1 and GEA-2, IEEE Trans. Inf. Forensics Secur. 17 (Aug. 2022) 2878–2889.
[29] E. Biham, O. Dunkelman, Differential cryptanalysis in stream ciphers, Cryptology ePrint Archive, Report 2007/218, 2007 [Online], http://eprint.iacr.org/.
[30] S. Fluhrer, I. Mantin, A. Shamir, Weaknesses in the Key Scheduling Algorithm of RC4, in: Proc. SAC 2001, Toronto, Ontario, Canada, Aug. 16–17, 2001, pp. 1–24.
[31] M. Ciet, G. Piret, J.J. Quisquater, Related-key and slide attacks: analysis, connections, and improvements, in: Proceedings of the 23rd Symposium on Information Theory, June 2002, pp. 315–325.