Collision Attack on 4-branch, Type-2 GFN based Hash Functions using Sliced Biclique Cryptanalysis Technique

Megha Agrawal, Donghoon Chang, Mohona Ghosh, and Somitra Kumar Sanadhya

Indraprastha Institute of Information Technology, Delhi (IIIT-D), India
{meghaa,donghoon,mohonag,somitra}@iiitd.ac.in

Abstract. In this work, we apply the sliced biclique cryptanalysis technique to show 8-round collision attack on a hash function H based on 4-branch, Type-2 Generalized Feistel Network (Type-2 GFN). This attack is generic and works on 4-branch, Type-2 GFN with any parameters including the block size, type of round function, the number of S-boxes in each round and the number of SP layers inside the round function. We first construct a 8-round distinguisher on 4-branch, Type-2 GFN and then use this distinguisher to launch 8-round collision attack on *compression functions* based on Matyas-Meyer-Oseas (MMO) and Miyaguchi-Preneel (MP) modes. The complexity of the attack on 128-bit compression function is 2^{56}. The attack can be directly translated to collision attack on MP and MMO based *hash functions* and pseudo-collision attack on Davies-Meyer (DM) based *hash functions*. When the round function F is instantiated with double SP layer, we show the first 8 round collision attack on 4-branch, Type-2 GFN with double SP layer based compression function. The previous best attack on this structure was a 6-round near collision attack shown by Sasaki at Indocrypt'12. His attack cannot be used to generate full collisions on 6-rounds and hence our result can be regarded the best so far in literature on this structure.

Keywords: Sliced Biclique cryptanalysis, hash functions, collision attack, generalized Feistel network, double SP layer

1 Introduction

Feistel structure is one of the basic building blocks of block ciphers and block ciphers based constructions. A Feistel network divides the input message into two sub-blocks (or two branches). Generalized Feistel Networks (GFN) are variants of Feistel networks with more than two branches, i.e., a k-branch GFN partitions the input message into k sub-blocks. They are sometimes favored over traditional Feistel scheme due to their high parallelism, simple design and suitability for low cost implementations. Many types of generalized Feistel schemes have been proposed and studied by researchers, e.g., unbalanced Feistel network [26], alternating Feistel Network [2], type-1, type-2 and type-3 Feistel network [34] etc. Type-2, GFN in particular has seen wide adoption in well known block ciphers such as RC6 [23], SHAvite3 [3], CLEFIA [28], HIGHT [13] etc. Security analysis of generalized Feistel network [4,30,32,11,27] has been an active area of research for past many years. In fact, a comprehensive study done by Bogdanov et al. in [6] suggests that Type-2 GFN and its variants are more robust and secure against differential and linear cryptanalysis as compared to Type-1 GFN. Hence, we choose Type-2 GFN (shown in Fig. 1) as the basis for our study.

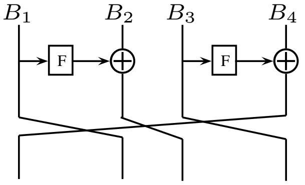

Fig. 1. 4 branch, Type-2 Generalized Feistel Structure with right cyclic shift.

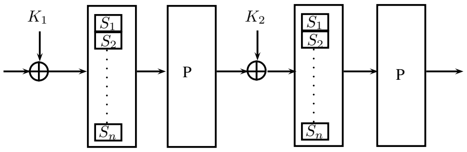

Fig. 2. Double SP Function.

Biclique cryptanalysis technique has garnered considerable interest amongst cryptographic community in the past couple of years. This approach, which is a variant of meet-in-the-middle attack, was first introduced by Khovratovich et al. in [16] for preimage attack on hash functions Skein and SHA-2. The concept was taken over by Bogdanov et al. [5] to successfully cryptanalyze full round AES and has been subsequently adopted to break many other block ciphers such as ARIA [33], SQUARE [19], TWINE [7], HIGHT [12], PRESENT [1] etc. All these biclique related attacks are carried out under the "unknown key settings" where the key used is unknown to the attacker and the main motive is to recover the secret key. However, this may not always be the case. Particularly, in the case of block cipher based hash modes such as Matyas-Meyer-Oseas (MMO) and Miyuguchi-Preneel (MP), initial vector IV (which acts as the key to the underlying block cipher) is a fixed public constant assumed to be known apriori to the attacker. Such scenarios are called "known key settings" in the attack model. Under such conditions, the aim of the attacker is to find a property which distinguishes known key instantiations of target block cipher from random permutations [17,20]. These settings are considered much stronger from the attacker's point of view since he unwillingly loses some degree of freedom reducing chances of carrying out actual generic attacks such as finding full collisions. Until recently, most of the collision attacks on hash functions under MMO and MP modes were restricted to variants of generic attack such as pseudo-collisions [18] and near collisions [29]. In [15], Khovratovich used biclique technique to mount actual collision and preimage attacks on Grøstl and Skein under known key settings. He proposed a variant of classical biclique technique used in [5] to carry out his attack. He termed this variant as *sliced biclique* technique (details of which are discussed in Section 3.3). Though the results of this work are quite interesting, yet they have not been studied further. Although the security of GFN have been studied earlier under known key settings [25,9,14,24,8], all these previous studies have utilized rebound attack technique [21] for their cryptanalysis. These factors motivated us to investigate the use of sliced biclique framework to study Type-2, GFN based constructions under known key settings.

It is generally desired that round function F inside a generalized Feistel network should provide good diffusion and confusion properties. This is often realized by implementing F as a *substitution-permutation network* (nonlinear S-box transformation followed by linear permutation) as part of the round function design. There is a general belief that increasing the number of active S-boxes provides more security against certain attacks. In [6], Bogdanov and Shibutani stressed on the importance of double SP (substitution-permutation) layers in the round function of Feistel networks as opposed to the single SP layer in the traditional design. They analyzed several designs such as single SP, double SP, SPS (substitution-permutation-substitution) and multiple SP layers and showed that double SP (shown in Fig. 2) layer achieves maximum security with respect to proportion of active S-boxes in all S-boxes involved against differential and linear cryptanalysis. They especially compared double SP structure with single SP and showed that for Type-1 and Type-2 GFNs, proportion of linearly and differentially active S-boxes in double SP instantiations is 50% and 33% higher respectively as compared to the single SP instantiation. Their research advocated a possibility of designing more efficient and secure block cipher based constructions using double SP layer. In [24], Sasaki presented a 7-round distinguisher attack on 4-branch, type-2 GFN with double SP layer and a 6-round near collision attack on the compression function based on the same structure. Kumar et al. [8] further improved the distinguishing attack on 4-branch, type-2 GFN with double SP layer by showing a 8-round distinguisher for the same. However, the form of truncated differential trails followed in [24,8] cannot be used to launch collision attack when the above GFN structure is instantiated in compression function modes under known key settings.

Our Contributions. The main contributions of this work are as follows:

1. We apply sliced biclique technique to construct a 8-round distinguisher on 4-branch, Type-2 Generalized Feistel Network.
2. We use the distinguisher so constructed to demonstrate a 8-round collision attack on 4-branch, Type-2 GFN based compression functions (in MMO and MP mode) under known key settings with a complexity of 2^{56} (on 128-bit hash output). The attack can be directly translated to collision attacks on MMO and MP mode based hash functions and pseudo-collision attacks on Davies-Meyer (DM) mode based hash functions.
3. When the round function F is instantiated with double SP layer, we demonstrate the first 8-round collision attack on 4-branch, Type-2 GFN with double SP layer.

4. We investigate CLEFIA which is a real world-implementation of 4-branch, Type-2 GFN and demonstrate an 8-round collision attack on CLEFIA based hash function with a complexity of 2^{56}.

The paper is organized as follows. In Section 2 we give the notations used in our paper followed by Section 3 which explains the important preliminaries. In Section 4, we present our distinguishing attack on 8 rounds of 4 branch, Type-2 GFN under fixed key settings. We use this distinguishing attack to show collision attack on 4-branch, Type-2 GFN based compression function in Section 5 followed by extension of this attack to hash functions in Section 6. Finally in Section 7, we summarize and conclude our work. The collision attack on CLEFIA based hash function is discussed in Appendix A.

2 Notation

We consider 4-branch, type-2, generalized Feistel network for our attack. Following notation is followed in the rest of the paper.

$\mathbf{N}$: Input message size (in bits)
$\mathbf{n}$: Message word size (in bits) which is input to each branch, i.e., $n = N/4$
$\$\mathbf{R}$: Round R
$\$\mathbf{R_p}$: p^{th} word in round R. Each round has 4 words corresponding to 4 partitions of 4-branch GFN, i.e., $1 \leq p \leq 4$
$\$\mathbf{R_p^l}$: l^{th} block of word p in round R

3 Preliminaries

In this section, we give a brief overview of the key concepts used in our cryptanalysis technique to facilitate better understanding.

3.1 Type-2 Generalized Feistel Network (GFN) instantiated with double SP layer

One round of Type-2 GFN is shown in Fig. 3. A GFN with 4 branches divides the input B into four equal parts $[B_1, B_2, B_3, B_4]$. A round of Type-2 GFN with left cyclic shift outputs $[F(B_1) \oplus B_2, B_3, F(B_3) \oplus B_4, B_1]$ for some keyed nonlinear function F [6]. On the other hand, a round of Type-2 GFN with right cyclic shift outputs $[F(B_3) \oplus B_4, B_1, F(B_1) \oplus B_2, B_3]$ (shown in Fig. 1) for round function F.

The round transformation function F when defined by non-linear S-box layer followed by a permutation layer P exhibits substitution permutation structure. The permutation P is generally implemented using standard MDS matrix [22,31]. If this SP structure is applied twice one after another then it is called double SP, as shown in Fig. 2. Few reasons favoring double SP over single SP function are as follows [6]:

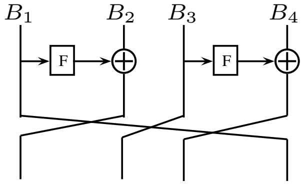

Fig. 3. 4-branch, Type-2, Generalized Feistel Network with left cyclic shift.

- The second S-box in double SP provides larger number of active S boxes when differential and linear attacks are applied.
- The second permutation layer in double SP structure limits the differential effect, i.e., number of differential trails resulting in same differential is smaller as compared to round function having single permutation layer.

3.2 *t*-bit Partial Target Preimage Attack

Let the output of a hash function H with initial chaining value IV and message M be denoted by h, i.e., $h = \mathrm{H}(IV,\, M)$. In this attack, when the attacker is given t-bits of h, his aim is to find a message M' such that the hash output $h' = \mathrm{H}(IV,\, M')$ matches these t-bits of h and at the same positions. The other bits of hash output $\mathrm{H}(IV,\, M')$ are generated randomly.

3.3 Sliced Biclique Cryptanalysis

In this section, we describe sliced biclique cryptanalysis technique to show preimage attack on hash function. Later, we use this preimage attack to launch collision attack. We consider MMO mode for our explanation. In the MMO mode $H = E_{IV}(M) \oplus M$, where IV is the initial chaining value acting as the key for the block cipher E, M is the message and H is the hash value produced. Since we assume IV to be public and hence known to the attacker, the cipher E becomes a simple permutation, i.e., $H = E(M) \oplus M$. Sliced biclique technique can then be applied for preimage search as follows.

The attacker first selects an internal intermediate state Q and partitions the full state space into sets of size 2^{2d} represented as $Q_{i,j}$ for some suitable range of i and j. Each set is defined by its base state $Q_{0,0}$ which is randomly selected by the attacker. Let f be a sub-permutation within E which maps $Q_{i,j}$ to another set of intermediate states $P_{i,j}$, i.e., $Q_{i,j} \underset{f}{\rightarrow} P_{i,j}$. These $Q_{i,j}$ and $P_{i,j}$ are obtained using 2^d Δ_i and ∇_j differentials as follows:

1. $Q_{0,0} \underset{f}{\rightarrow} P_{0,0}$ .... (base computation),
2. $Q_{i,0} = Q_{0,0} \oplus \Delta_i$,
3. $Q_{i,0} \underset{f}{\rightarrow} P_{i,0}$,
4. $Q_{0,j} = Q_{0,0} \oplus \nabla_j$,

5. $Q_{0,j} \xrightarrow[f]{} P_{0,j}$,
6. $Q_{i,j} = Q_{i,0} \oplus \nabla_j$,
7. $P_{i,j} = P_{0,j} \oplus \Delta_i$, where $0 \leq i, j \leq 2^d - 1$.

It has been shown in [15] that $Q_{i,j} \xrightarrow[f]{} P_{i,j}$ forms a biclique, if Δ_i and ∇_j trails are non-interleaving, i.e., they do not share any active non-linear component between them. [1] The parameter d is called the dimension of the biclique. Each $Q_{0,0}$ defines one biclique structure consisting of 2^{2d} intermediate states.

To find a valid preimage M, the attacker then applies meet-in-the-middle (MITM) technique in the rest of the rounds. In the MITM stage, the attacker chooses an internal state $v \in \{E \setminus f\}$ and computes its value both in the forward direction as a function of P (denoted as $\overrightarrow{v_{i,j}}$) and in the backward direction as a function of Q (denoted as $\overleftarrow{v_{i,j}}$) respectively for every (i, j) pair. This process is shown in Fig. 4.

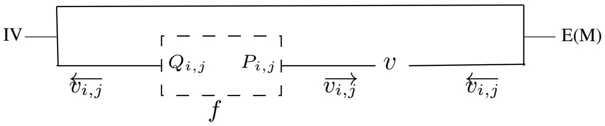
IV
E(M)
$Q_{i,j}$
$P_{i,j}$
v
$\overleftarrow{v_{i,j}}$
$\overrightarrow{v_{i,j}}$
$\overleftarrow{v_{i,j}}$
f

Fig. 4. Biclique Attack.

To compute $\overleftarrow{v}$ in the backward direction, the value of $E(M)$ is required (as shown in Fig. 4) which can be easily calculated by $E(M) = H \oplus M$. To reduce the complexity of the attack, the attacker tries to choose the state v such that in the forward direction it only depends on j and in the backward direction it only depends on i, i.e., states $Q_{i,j}$ and $P_{i,j}$ form a *sliced biclique* if the following conditions hold [15] [2]:

$$\forall i, j : \qquad \overrightarrow{v_{i,j}} = \overrightarrow{v_{0,j}},$$
$$\forall i, j : \qquad \overleftarrow{v_{i,j}} = \overleftarrow{v_{i,0}}.$$

Let $\overrightarrow{v_{0,j}} = \overrightarrow{v_j}$ and $\overleftarrow{v_{i,0}} = \overleftarrow{v_i}$. Finally, the attacker checks if:

$$\exists i, j : \qquad \overrightarrow{v_j} = \overleftarrow{v_i}.$$

If such an (i, j) pair exists, the corresponding $Q_{i,j}$ becomes the preimage candidate. If not, then the attacker picks up another set of states with different base value $Q_{0,0}$ and repeats the whole procedure.

[1] It is not necessary for independent biclique/sliced biclique attack to have Δ and ∇ differentials start from distinct ends of the subcipher. The only requirement that is essential is that both trails should be non-interleaving.

[2] In the traditional biclique key recovery attack in [5], this special restriction on v is not required.

Complexity of the attack. The sliced biclique attack comprises of 2 phases - biclique construction phase and MITM phase. Let the block cipher E consist of y rounds and the number of rounds covered in the biclique phase be x. This implies the number of rounds covered in the MITM phase is $y - x = z$. For each set of messages, in the biclique phase, since all $\Delta_i \neq \nabla_j$ and Δ_i trails are independent of ∇_j trails, the construction of biclique is simply reduced to computation of Δ_i and ∇_j trails independently which requires no more than 2.2^d computations of f, i.e.,

$$\text{Complexity of biclique phase} = 2^d \times \frac{x}{y} + 2^d \times \frac{x}{y} = 2^{d+1} \times \frac{x}{y}.$$

Similarly, in the MITM phase, the attacker needs to call each of $\overrightarrow{v_j}$ and $\overleftarrow{v_i}$ for 2^d times, i.e., a total of 2^{d+1} times. Let the number of rounds covered in the forward and backward direction be a and b respectively. Hence,

$$\text{Complexity of MITM phase} = 2^d \times \frac{a}{y} + 2^d \times \frac{b}{y} = 2^d \times \frac{a + b(= z)}{y} = 2^d \times \frac{y - x}{y}.$$

It is now easy to check that the overall complexity of sliced biclique preimage attack for one set of messages does not require more than 2^d full computations of E, i.e.,

$$\text{Total Complexity} = 2^{d+1} \times \frac{x}{y} + 2^d \times \frac{y - x}{y} = 2^d \times (1 + \frac{x}{y}) \approx 2^d \text{ since, } x \ll y.$$

If m bicliques are constructed, then the total cost is $m \times 2^d$. For further reading on sliced biclique and classical bicliques one can refer to [15] and [16,5] respectively.

4 Distinguishing Attack on 4-branch, Type-2 GFN based Permutation using Sliced Biclique Cryptanalysis Technique

In this section, we present a 8-round distinguisher on permutation E_k (where k is the key) which is a 8-round, 4-branch, Type-2 Generalized Feistel Network. We assume that the S-box layer has good differential property and the P-layer implements standard MDS matrix. [3] We also assume that the key k (that is IV in the overlying hash function construction) is a fixed constant. The distinguishing property used by the distinguisher is as follows:

Distinguishing Property. Let E_k be a block cipher with message size $N =$ 128-bits. The aim of the adversary is to collect 2^{16} (plaintext, ciphertext) pairs such that the XOR of the lower 16 bits of the third word in the plaintext and

[3] In this line of work, implementation of P-layer as a standard MDS matrix having optimal branch number is believed to be a good design choice [6,25,14,24]

the lower 16 bits of the third word in the ciphertext (where each word is of size 32-bits) is always a 16-bit constant value chosen by the attacker, i.e.,

$$(\text{plaintext})_3^2 \oplus (\text{ciphertext})_3^2 = \text{constant} \tag{1}$$

where, $|\text{constant}| = 16$-bits. [4]

In case of random permutation. When E_k is a random permutation, the probability that any (plaintext, ciphertext) pair satisfies the desired property (as mentioned in Equation 1) is approximately 2^{-16}. This means that the expected time complexity to generate one such (plaintext, ciphertext) pair is 2^{16}. Hence, expected time complexiy to generate 2^{16} such (plaintext, ciphertext) pairs is 2^{32}.

In case of E instantiated with 4-branch, Type-2, GFN. For the illustration of our attack, we consider N =128-bit and n = 32-bit each. The attacker first chooses a random base value $Q_{0,0}$ (as discussed in Section 3.3). Let $\Delta_i = (\bar{0}\bar{0} \mid i\bar{0} \mid \bar{0}\bar{0} \mid \bar{0}\bar{0})$ and $\nabla_j = (\bar{0}\bar{0} \mid \bar{0}j \mid \bar{0}\bar{0} \mid \bar{0}\bar{0})$ where $(0 \leq i, j \leq 2^{16} - 1)$ be the Δ and ∇ differences injected in Round 4. Here each $\bar{0}$ represents 0^{16}. The propagation of Δ_i trail (marked as '|' in green) and ∇_j trail (marked as '-' in red) is shown in Fig. 5 and Fig. 6 respectively. In these figures, the four words shown in each round are the corresponding inputs to four branches at each round. In ∇_j trail, the attacker first injects the given j difference in $\$4_2^2$ word only. As the ∇_j trail propagates as shown in Fig. 6, $\$4_1$ and $\$4_4$ words are subsequently affected. The dimension of this biclique is d=16.

It is easy to check that Δ_i and ∇_j trails are independent and do not share any non-linear components (shown in Fig. 7) between them in rounds 4 and 5. Thus, a 2-round biclique (consisting of $2^{2d} = 2^{32}$ messages) is formed where the biclique covers rounds \$4 and \$5. Now the aim of the attacker is to find a matching variable v which only depends on Δ_i trail in one direction and ∇_j trail in the other direction (as discussed in Section 3.3). Hence, from round 6 only ∇_j trail is propagated in the forward direction and from round 3 only Δ_i trail is propagated in the backward direction (as shown in Fig. 8). At the end of 8^{th} round it can be seen that $\$1_3^2$ (marked in yellow in Fig. 8) in the backward direction is not affected by Δ_i trail (i.e., will be affected by ∇_j trail only) and $\$8_3^2$ (marked in yellow in Fig. 8) in the forward direction remains unaffected by ∇_j trail (i.e., will be affected by Δ_i trail only). Through feed forward operation, 16 bits of $\$1_3^2$ can then be matched with 16 bits of $\$8_3^2$. Hence, in this attack we choose $\$8_3^2$ to be our matching variable v and $|v| = 16$ which is denoted by t.

Once the matching variable v is obtained, as mentioned above, through our biclique attack, $2^{2d} = 2^{32}$ (plaintext, ciphertext) pairs are generated in a set. Out of these 2^{2d} (plaintext, ciphertext) pairs, there exists $2^{2d-t} = 2^{16}$ (plaintext, ciphertext) pairs which match on matching variable v. In other words, if we XOR the lower 16 bits of the third word in the plaintext and the lower 16 bits of the third word in the ciphertext (i.e., at positions $\$1_3^2$ and $\$8_3^2$ respectively),

[4] Here $(plaintext)_3^2$ denotes second block of third word of plaintext as described in Section 2. The term $(ciphertext)_3^2$ can be understood similarly.

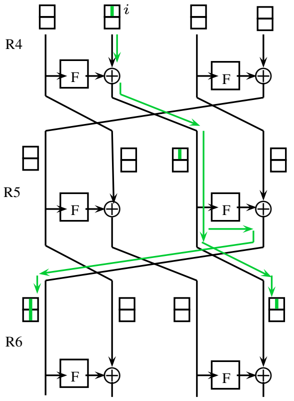

Fig. 5. Δ_i difference injection in Round 4 and its propagation.

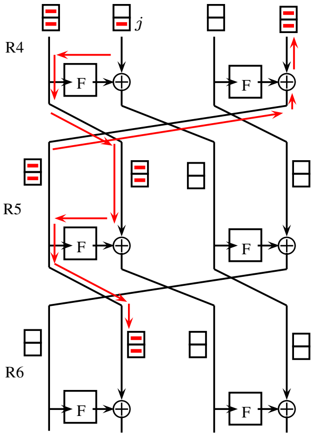

Fig. 6. ∇_j difference injection in Round 4 and its propagation.

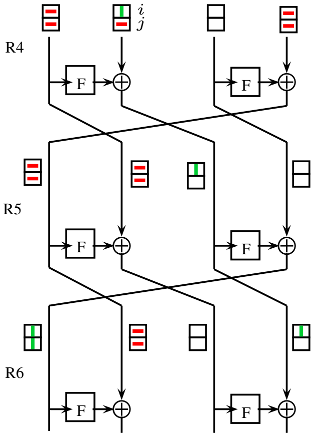

Fig. 7. 2-round biclique placed in Round 4 - 5.

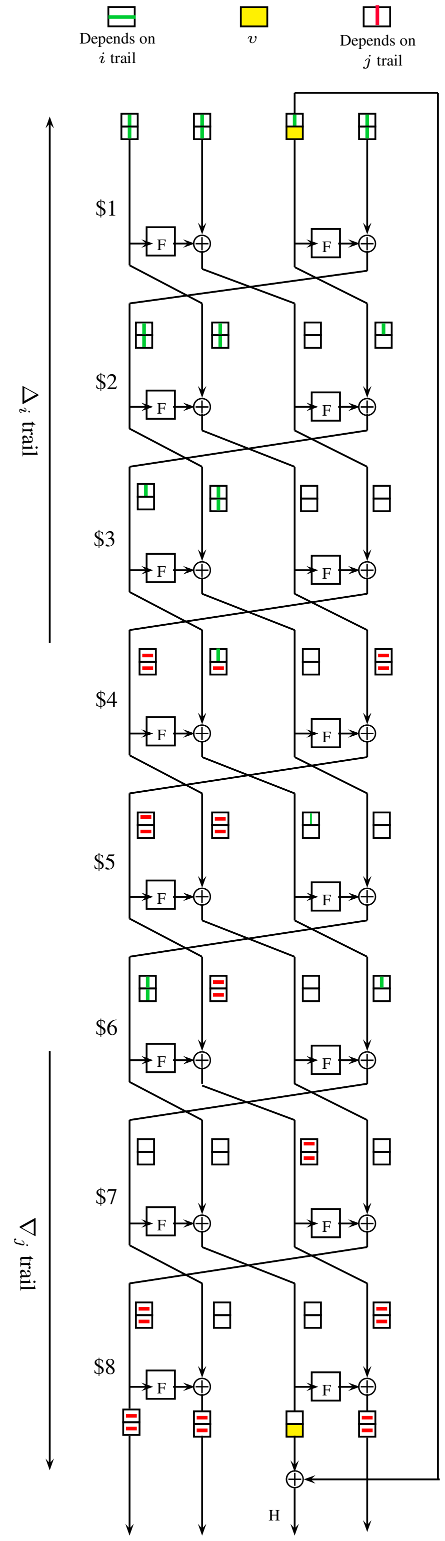

Fig. 8. Matching in 8 rounds of 4-branch Type-2 GFN with right cyclic shift.

Equation 1 will always be satisfied. These 2^{16} (plaintext, ciphertext) pairs will be generated with a computational complexity of $2^d = 2^{16}$ (as discussed in § 3.3) which is lower than the computational complexity of 2^{32} in case of random permutation. Hence, a valid distinguisher for E when instantiated with 4-branch, Type-2, GFN is constructed.

Similarly, our attack can be applied to messages of other sizes as well. In table 1, we report the complexity values for our distinguisher attack on message inputs of different size.

Table 1. Complexity values for our distinguishing attack on message inputs of different size. Here N represents the input message size in bits and #(P-C) pairs represent the number of plaintext-ciphertext pairs needed for our attack. The number of plaintext-ciphertext pairs depends on the size of matching variable v.

N	n	#(P-C) pairs	Complexity of our attack	Complexity of random permutation
64	16	2^8	2^8	2^{16}
256	64	2^{32}	2^{32}	2^{64}
512	128	2^{64}	2^{64}	2^{128}

5 Collision Attack on 4-branch, Type-2, GFN based compression function

The distinguisher constructed in the previous section can be used to launch collision attack on 4-branch, Type-2, GFN based compression function as described below. Here the compression function is assumed to be in MMO mode and the output is assumed to be of $N = 128$-bits.

- The attacker first chooses a t-bit constant of his choice.
- In the above attack, the attacker then finds a matching variable v, where $|v| \leq t$. In our attack, $|v| = t = 16$ bits.
- There are $2^{2d} = 2^{32}$ messages in a biclique set. Out of these 2^{2d} messages, only 2^{2d-t} messages will match on v. This means that out of 2^{32} messages only 2^{16} messages will survive the MITM phase.
- In other words, it can be said that the attacker has generated 2^{16} t-bit partial target preimages with these t-bits equal to an arbitrarily chosen constant selected in first step.
- These 2^{16} t-bit partial target preimages collide on $t = 16$ bits. Hence, if the attacker generates $2^{(N-t)/2}$ such preimages which collide on t-bits, there exists a colliding pair with high probability which collide on the remaining $N - t$ bits as well. Thus, the attacker will generate $2^{(128-16)/2} = 2^{56}$ such t-bit partial target preimages to obtain a collision on complete hash output H with high probability.

– Now, one sliced biclique generates 2^{16} t-bit partial target preimages. Hence, to generate 2^{56} such preimages, the attacker needs to construct $2^{56-16} = 2^{40}$ sliced bicliques (or, $2^{(N-t)/2-(2d-t)}$ bicliques where, $2^{(N-t)/2} = 2^{56}$ and $2^{(2d-t)} = 2^{16}$).

Complexity of the collision attack . Since the computational complexity of performing sliced biclique attack once is $2^d = 2^{16}$ (as discussed in Section 3.3), hence computational complexity of running sliced biclique attack 2^{40} times is $2^{40} \times 2^{16} = 2^{56}$. Therefore, given IV, the complexity to find a pair of messages $(M,\ M')$ such that $\mathrm{CF}(IV,\ M) = \mathrm{CF}(\mathrm{IV},\ M')$, when CF (i.e., compression function) is instantiated with 8-rounds of 4-branch type-2 GFN is 2^{56} ($< 2^{64}$ brute-force attack). Here, compression function output is of 128-bits size. In general, the complexity of the attack is given by the following formula:

$$\text{Complexity} = 2^{\frac{(N-t)}{2}-(2d-t)} \times 2^d.$$

For the purpose of illustration, we show the cost of our attack for various message sizes in Table 2.

Table 2. Complexity values for our collision attack on message inputs of different size. Here N represents the input message size in bits, n represents the branch word size in bits and t represents the size of matching variable v in bits. In our attack $d = t$ always.

N	n	t	Rounds	Complexity of our attack	Brute force complexity
64	16	8	8	2^{28}	2^{32}
128	32	16	8	2^{56}	2^{64}
256	64	32	8	2^{112}	2^{128}
512	128	64	8	2^{224}	2^{256}

Since we need to store all partial preimages to find the colliding pair, memory required is of the order of 2^{56} (for 128-bit output). However, it is mentioned in [15] that memoryless equivalents of these attacks do exist. In Appendix A, we show the collision attack on CLEFIA which is a real world implementation of 4-branch, Type-2, GFN.

Collision Attack on 4-branch Type-2 GFN with Double SP layer . The above attack technique is generic and independent of the internal F-function structure. Hence, if we instantiate the round function F with double SP-layer, the above attack can be directly translated to 8-round collision attack on 4-branch, Type-2 GFN with double SP layer based compression function with a complexity of 2^{56}. This betters the 6-round near collision attack on the same

Table 3. Comparison of our results with previous cryptanalytic results on 4-branch, Type-2, GFN with double SP layer.

Rounds	Attack Type	Reference
6	Near Collisions	[24]
7	Distinguishing	[24]
8	Distinguishing	[8]
8	Distinguishing	This work, § 4
8	Full Collisions	This work, § 5

structure shown by Sasaki in [24]. In Table 3 we compare our result with the previous cryptanalysis results on 4-branch, Type-2 GFN with double SP layer.

As discussed above, since the attack technique is generic, presence of multiple SP layers in the round function F does not provide any extra resistance against sliced biclique attack as compared to double SP layer. In fact, in our collision attack neither the attack complexity nor the the number of rounds attacked change if double SP layer is replaced by multiple SP layers. This is in contrast to attacks such as rebound attacks [21], where the number of SP layers inside the round function F influence the number of rounds attacked [25,9,14,24,8] in Generalized Feistel Networks.

6 Collision Attack on Hash Functions

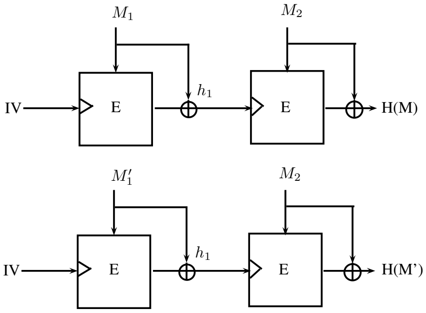

Fig. 9. Collision Attack.

In this attack, given the IV, the aim of the attacker is to find a pair of messages (M, M') such that $H(M)$ =$H(M')$. To do so, the attacker first finds two messages M_1 and M_1' which collide to same hash value h_1 using collision attack technique described in Section 5 with a complexity of 2^{56}. Now he concatenates any message M_2 with M_1 and M_1' (as shown in Fig. 9) such that $H(M_1\|M_2)$ =$H(M_1'\|M_2)$. Message M_2 can also be chosen such that it satisfies padding restrictions (where length of input message is appended at the end) if required. In this way, collision attack can be carried out on 4-branch, Type-2, GFN with double SP layer based hash function with a complexity of 2^{56}. Since we assume known key settings (i.e., key part to the underlying block cipher is known to the attacker), hence this attack can be used to generate collisions in MP and MMO based hash functions but pseudo collisions in DM based hash functions.

7 Conclusions

In this work, we apply the sliced biclique technique to show collision attack on 8-rounds of 4-branch, type-2 GFN. When it is instantiated with double SP layer, we present the first 8-round collision on 4-branch, type-2 GFN with double SP layer. It would be interesting to apply sliced biclique technique to attack other potential targets. One possible extension can be to apply this attack technique on 2-branch, Type-2 GFN such as Shavite-3 etc.

Acknowledgements. The authors would like to thank the anonymous reviewers for their valuable comments as it helped in improving the quality of the paper.

References

1. Farzaneh Abed, Christian Forler, Eik List, Stefan Lucks, and Jakob Wenzel. Biclique Cryptanalysis Of PRESENT, LED, And KLEIN. Cryptology ePrint Archive, Report 2012/591, 2012. `http://eprint.iacr.org/2012/591`.
2. Ross J. Anderson and Eli Biham. Two Practical and Provably Secure Block Ciphers: BEARS and LION. In Gollmann [10], pages 113–120.
3. Eli Biham and Orr Dunkeman. The SHAvite-3 Hash Function. Submission to NIST SHA-3 competition. `www.cs.technion.ac.il/~orrd/SHAvite-3/`.
4. Andrey Bogdanov. On the differential and linear efficiency of balanced Feistel networks. *Inf. Process. Lett.*, 110(20):861–866, 2010.
5. Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT'11*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2011.
6. Andrey Bogdanov and Kyoji Shibutani. Generalized Feistel networks revisited. *Des. Codes Cryptography*, 66(1-3):75–97, 2013.
7. Mustafa Çoban, Ferhat Karakoç, and Özkan Boztas. Biclique Cryptanalysis of TWINE. In Josef Pieprzyk, Ahmad-Reza Sadeghi, and Mark Manulis, editors, *CANS'12*, volume 7712, pages 43–55. Springer, 2012.
8. Donghoon Chang, Abhishek Kumar, and Somitra Kumar Sanadhya. Security Analysis of GFN: 8-Round Distinguisher for 4-Branch Type-2 GFN. In Goutam Paul and Serge Vaudenay, editors, *INDOCRYPT'13*, volume 8250 of *Lecture Notes in Computer Science*, pages 136–148. Springer, 2013.
9. Le Dong, Wenling Wu, Shuang Wu, and Jian Zou. Known-key distinguishers on type-1 Feistel scheme and near-collision attacks on its hashing modes. *Frontiers of Computer Science*, 8(3):513–525, 2014.

10. Dieter Gollmann, editor. *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 21-23, 1996, Proceedings*, volume 1039 of *Lecture Notes in Computer Science*. Springer, 1996.
11. Viet Tung Hoang and Phillip Rogaway. On Generalized Feistel Networks. In Tal Rabin, editor, *CRYPTO'10*, volume 6223 of *Lecture Notes in Computer Science*, pages 613–630. Springer, 2010.
12. Deukjo Hong, Bonwook Koo, and Daesung Kwon. Biclique Attack on the Full HIGHT. In Howon Kim, editor, *ICISC'11*, volume 7259 of *Lecture Notes in Computer Science*, pages 365–374. Springer, 2011.
13. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jaesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In Louis Goubin and Mitsuru Matsui, editors, *CHES'06*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer, 2006.
14. HyungChul Kang, Deukjo Hong, Dukjae Moon, Daesung Kwon, Jaechul Sung, and Seokhie Hong. Known-Key Attacks on Generalized Feistel Schemes with SP Round Function. *IEICE Transactions*, 95-A(9):1550–1560, 2012.
15. Dmitry Khovratovich. Bicliques for Permutations: Collision and Preimage Attacks in Stronger Settings. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT'12*, volume 7658 of *Lecture Notes in Computer Science*, pages 544–561. Springer, 2012.
16. Dmitry Khovratovich, Christian Rechberger, and Alexandra Savelieva. Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family. In Anne Canteaut, editor, *FSE'12*, volume 7549 of *Lecture Notes in Computer Science*, pages 244–263. Springer, 2012.
17. Lars R. Knudsen and Vincent Rijmen. Known-key distinguishers for some block ciphers. In Kaoru Kurosawa, editor, *ASIACRYPT'07*, volume 4833 of *Lecture Notes in Computer Science*, pages 315–324. Springer, 2007.
18. Ji Li, Takanori Isobe, and Kyoji Shibutani. Converting Meet-In-The-Middle Preimage Attack into Pseudo Collision Attack: Application to SHA-2. In Anne Canteaut, editor, *FSE'12*, volume 7549 of *Lecture Notes in Computer Science*, pages 264–286. Springer, 2012.
19. Hamid Mala. Biclique Cryptanalysis of the Block Cipher SQUARE. Cryptology ePrint Archive, Report 2011/500, 2011. `http://eprint.iacr.org/2011/500`.
20. Florian Mendel, Thomas Peyrin, Christian Rechberger, and Martin Schläffer. Improved Cryptanalysis of the Reduced Grøstl Compression Function, ECHO Permutation and AES Block Cipher. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *SAC'09*, volume 5867 of *Lecture Notes in Computer Science*, pages 16–35. Springer, 2009.
21. Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl. In Orr Dunkelman, editor, *FSE'09*, volume 5665 of *Lecture Notes in Computer Science*, pages 260–276. Springer, 2009.
22. Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bosselaers, and Erik De Win. The cipher SHARK. In Gollmann [10], pages 99–111.
23. Ronald L. Rivest, Matthew J. B. Robshaw, and Yiqun Lisa Yin. RC6 as the AES. In *AES Candidate Conference*, pages 337–342, 2000.
24. Yu Sasaki. Double-SP Is Weaker Than Single-SP: Rebound Attacks on Feistel Ciphers with Several Rounds. In Steven D. Galbraith and Mridul Nandi, editors, *INDOCRYPT'12*, volume 7668 of *Lecture Notes in Computer Science*, pages 265–282. Springer, 2012.

25. Yu Sasaki and Kan Yasuda. Known-Key Distinguishers on 11-Round Feistel and Collision Attacks on Its Hashing Modes. In Antoine Joux, editor, *FSE'11*, volume 6733 of *Lecture Notes in Computer Science*, pages 397–415. Springer, 2011.
26. Bruce Schneier and John Kelsey. Unbalanced Feistel Networks and Block Cipher Design. In Dieter Gollmann, editor, *FSE'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 121–144. Springer, 1996.
27. Taizo Shirai and Kyoji Shibutani. Improving Immunity of Feistel Ciphers against Differential Cryptanalysis by Using Multiple MDS Matrices. In Bimal K. Roy and Willi Meier, editors, *FSE'04*, volume 3017 of *Lecture Notes in Computer Science*, pages 260–278. Springer, 2004.
28. Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-Bit Blockcipher CLEFIA (Extended Abstract). In Alex Biryukov, editor, *FSE'07*, volume 4593 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2007.
29. Bozhan Su, Wenling Wu, Shuang Wu, and Le Dong. Near-Collisions on the Reduced-Round Compression Functions of Skein and BLAKE. In Swee-Huay Heng, Rebecca N. Wright, and Bok-Min Goi, editors, *CANS'10*, volume 6467 of *Lecture Notes in Computer Science*, pages 124–139. Springer, 2010.
30. Tomoyasu Suzaki and Kazuhiko Minematsu. Improving the Generalized Feistel. In Seokhie Hong and Tetsu Iwata, editors, *FSE'10*, volume 6147 of *Lecture Notes in Computer Science*, pages 19–39. Springer, 2010.
31. Serge Vaudenay. On the need for multipermutations: Cryptanalysis of MD4 and SAFER. In Bart Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 286–297. Springer, 1994.
32. Wenling Wu, Wentao Zhang, and Dongdai Lin. Security on Generalized Feistel Scheme with SP Round Function. *I. J. Network Security*, 3(3):215–224, 2006.
33. Shao-zhen Chen Tian-min Xu. Biclique Attack of the Full ARIA-256. Cryptology ePrint Archive, Report 2012/011, 2012. `http://eprint.iacr.org/2012/011`.
34. Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai. On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 461–480. Springer, 1989.

A 8-Round Collision Attack on CLEFIA based Compression Function

In this section, we investigate CLEFIA which is a real world-implementation of 4-branch, Type-2 GFN. In the attacks discussed in Section 4 and Section 5, we considered 4-branch, Type-2 GFN with double SP layer where right cyclic shift is applied on the message sub-blocks at the end of each round. This was done to facilitate direct comparison with previous results [24,8] on the same structure. However in [34], Type-2 GFN's have been defined with left cyclic shift and is followed in all the practical implementations of Type-2 GFN structure - e.g., RC6 [23], CLEFIA [28], HIGHT [13] etc. Yet, similar attack procedure (as discussed in Section 5) can be applied on CLEFIA but with different Δ_i and ∇_j trails. CLEFIA is a 128-bit block cipher and supports three key lengths - 128-bit, 192-bit and 256-bit. The number of rounds correspondingly are 18, 22 and 26. Here, in this section, we examine CLEFIA with 128-bit keysize. [5] WK_0 and WK_1 represent the whitening keys at the start of the cipher. Each round has two 32-bit round keys RK_{2i-2} and RK_{2i-1} (where, $1 \leq \text{i} \leq 18$).

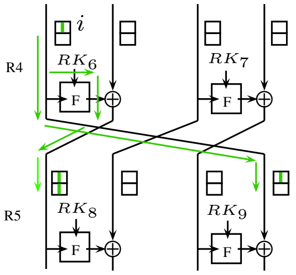

Fig. 10. Δ_i difference injection in Round 4 and its propagation

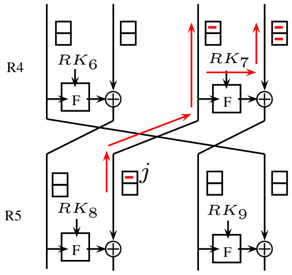

Fig. 11. ∇_j difference injection in Round 5 and its propagation

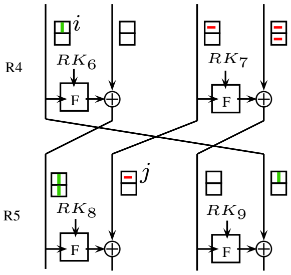

Fig. 12. 1-round biclique placed in Round 4

In this attack, let $\Delta_i = (i\bar{0} \mid \bar{0}\bar{0} \mid \bar{0}\bar{0} \mid \bar{0}\bar{0})$ be the Δ difference injected in Round 4 and $\nabla_j = (\bar{0}\bar{0} \mid j\bar{0} \mid \bar{0}\bar{0} \mid \bar{0}\bar{0})$ be the ∇ difference injected in Round 5 where $(0 \leq i, j \leq 2^{16}-1)$. Here each $\bar{0}$ represents 0^{16}. The attacker first chooses a random base value $Q_{0,0}$ and then injects the Δ_i and ∇_j differences accordingly. The propagation of Δ_i trail (marked as '|' in green) and ∇_j trail (marked as '-' in red) is shown in Fig. 10 and Fig. 11 respectively. The dimension of this biclique is d=16. It is easy to check that Δ_i and ∇_j trails are independent and do not share any non-linear components (shown in Fig. 12) between them in round 4. Thus a 1-round biclique (consisting of $2^{2d} = 2^{32}$ messages) is formed in $\$4$ round.

From round 5 only ∇_j trail is propagated in the forward direction and from round 3 only Δ_i trail is propagated in the backward direction (as shown in Fig. 13). At the end of 8^{th} round it can be seen that $\$1^2_3$ (marked in yellow in Fig. 13) in the backward direction is not affected by Δ_i trail and $\$8^2_3$ (marked in yellow in Fig. 13) in the forward direction remains unaffected by ∇_j trail. Through feed forward operation, 16 bits of $\$1^2_3$ can then be matched with 16 bits of $\$8^2_3$. Hence, in this attack we choose $\$8^2_3$ to be our matching variable v. The steps of collision attack for CLEFIA are exactly the same as discussed in Section 5 and Section 6. Therefore, we can generate collisions in 8-rounds of CLEFIA based hash function with a complexity of 2^{56}.

[5] The attack works on other key sizes as well since key is constant under known key settings.

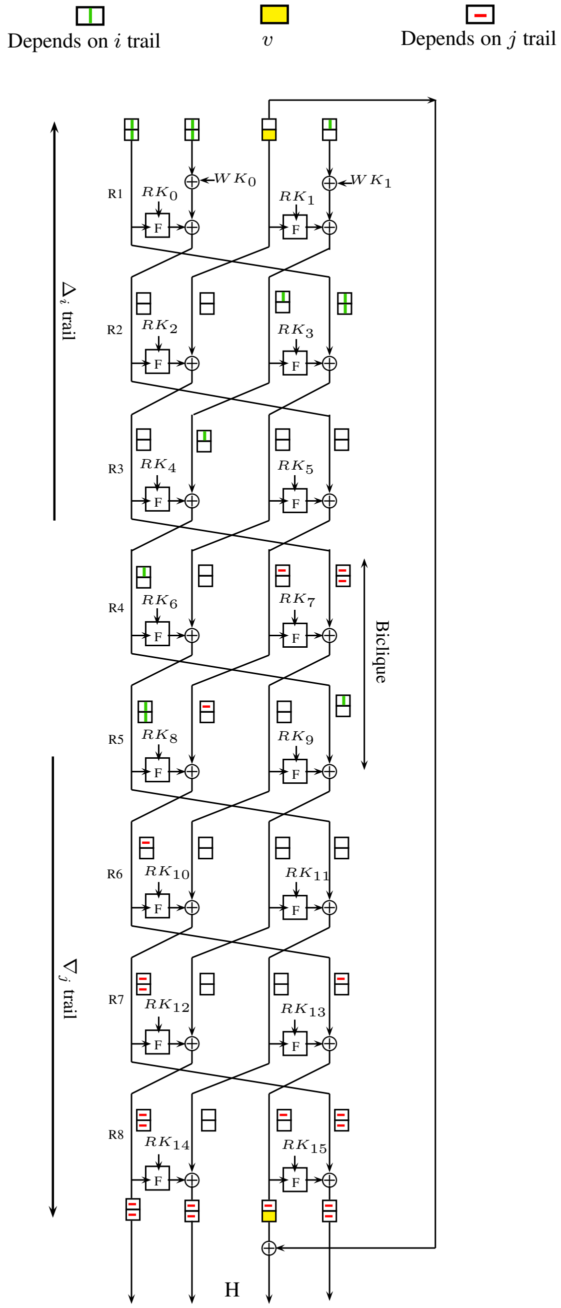

Fig. 13. Matching in 8 rounds of CLEFIA